CO-EVOLVING TUTOR AND STUDENT SIMULATOR FOR DIALOGUE TUTORING

Anonymous authors
Paper under double-blind review

ABSTRACT

Optimizing large language models for dialogue tutoring requires feedback about how guidance changes students' reasoning and problem-solving progress. An objective inspired by the zone of proximal development (ZPD) favors guidance associated with intermediate predicted response correctness. Our analysis identifies the largest simulation errors at intermediate reference correctness. As teaching policies evolve, student models must continue to represent the mistakes and incomplete reasoning that instructional guidance is intended to address. We propose COMET, an online reinforcement-learning framework that couples instructional utility optimization with student-response modeling through policy-induced dialogue histories. Simulator validation performance, in turn, controls the granularity of the Tutor's ZPD reward. The simulator objective combines cognitive rewards with reference-based behavioral constraints, jointly targeting reasoning quality and agreement with recorded student responses. A shared judge-anchored evaluator refines its rubric and distills rubric-based judgments into a prediction head for efficient cognitive scoring. COMET achieves 85.3% Clean Success@15. With an independent frozen student and a shared answer-disclosure guard, it improves Success@15 over a same-base prompted tutor by 20.1 percentage points. These findings demonstrate the effectiveness of tutor–student co-evolution for learning tutoring policies through self-play, with instructional gains that transfer to an independent student simulator.[1]

1 INTRODUCTION

Personalized tutoring supports learning through instructional decisions tailored to a student's evolving understanding (Bloom, 1984). Large language models extend this approach to open-ended dialogue, where effective guidance must respond to students' reasoning while preserving opportunities for independent problem solving (Macina et al., 2023). Existing approaches train tutoring policies from instructional demonstrations and response preferences (Scarlatos et al., 2025b), optimize dialogue outcomes through simulated student interaction (Dinucu-Jianu et al., 2025), or learn policies over high-level tutoring actions from static dialogue data (Nam et al., 2025). Complementary research develops student simulators through role prompting and training on recorded dialogues, targeting linguistic, behavioral, and cognitive aspects of student responses (Markel et al., 2023; Scarlatos et al., 2026).

The zone of proximal development (ZPD) concerns tasks learners can accomplish with guidance but not independently (Vygotsky, 1978). Socratic tutoring aims to elicit students' reasoning without disclosing the answer, and a ZPD-inspired objective favors guidance associated with intermediate predicted response correctness. Estimating correctness during Tutor optimization depends on simulated student responses, yet the simulator is commonly held fixed as the Tutor policy changes. Tutoring datasets record student responses to the instructional choices made during data collection. An updated Tutor can induce histories absent from these records, for which matched student-response supervision is unavailable. The initial simulator's answer-matching estimates have their largest MSE at intermediate smoothed reference correctness (Figure 1a). Student-response modeling must therefore adapt to Tutor-induced histories and preserve the mistakes and incomplete reasoning that instructional guidance is intended to address.

[1]Anonymous code repository: `https://anonymous.4open.science/r/comet_mode-FBE0`.

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.

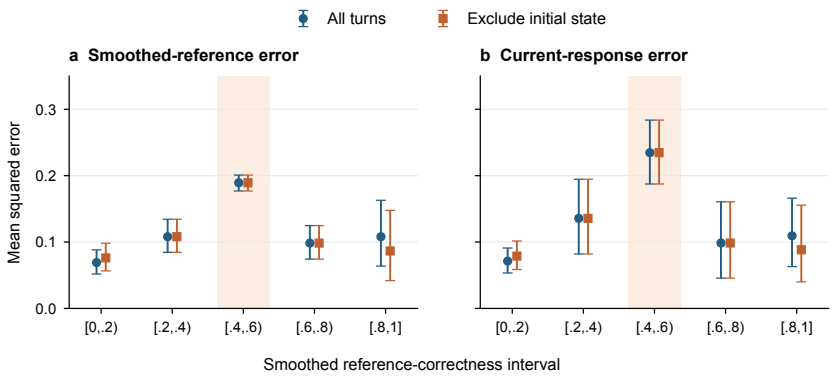


Figure 1: Initial simulator error on MathDial validation (217 dialogues, 1,316 turns). (a) MSE against smoothed reference correctness. (b) Brier score against the current response's binary correctness label. Points show means with pointwise 95% dialogue-bootstrap intervals; squares exclude each dialogue's initial state. Shading marks the central interval.

COMET treats Tutor optimization and Student Simulator adaptation as distinct online reinforcement-learning objectives coupled through policy-induced dialogue histories. The simulator supplies both feedback for comparing instructional actions and responses that determine subsequent states; Tutor updates, in turn, change the histories on which the simulator must respond. COMET fixes the simulator and evaluator during each Tutor update, then adapts the simulator to histories induced by the updated Tutor. A local GRPO analysis bounds how candidate-relative changes in simulated feedback perturb the Tutor's local GRPO surrogate gradient.

Our contributions are threefold:

- **A coupled reinforcement-learning formulation of tutoring and student simulation.** We show that student responses can provide identical ZPD-guided feedback yet induce different successor dialogue histories. We formulate Tutor optimization and simulator adaptation as distinct conditional objectives linked by Tutor-induced histories, with the simulator shaping both response transitions and Tutor reward estimates. For a fixed candidate group, we derive a conditional bound on how relative reward perturbations affect the local GRPO surrogate gradient.
- **A framework for tutor–student co-evolution.** We introduce COMET, which connects policy improvement to simulator adaptation through tutor-induced histories and reference-based behavioral constraints. Validation-gated rewards and judge-anchored evaluation coordinate the feedback used to optimize the two models as their interaction distribution evolves.
- **Empirical advantages in tutoring and student modeling.** COMET improves tutoring success under Socratic constraints, reduces overestimation at intermediate reference correctness, and enables zero-shot knowledge tracing without knowledge-component annotations. In matched evaluation with an independent frozen student, it improves Success@15 by 20.1 percentage points over a same-base prompted tutor, demonstrating benefits beyond the co-trained interaction partner.

2 Related Work

Learning tutoring policies. Existing work trains tutors from demonstrations, pedagogical preferences, or simulated dialogue outcomes, with policies acting either on utterances or on higher-level instructional moves. SocraticLM fine-tunes on synthetic Socratic dialogues with simulated students assigned different cognitive states (Liu et al., 2024). PACE synthesizes mathematics tutoring dialogues tailored to simulated student personas and fine-tunes a tutor on teacher responses (Liu et al., 2025). Expert decision models condition response generation on identified student errors, remediation strategies, and instructional intent (Wang et al., 2024). LearnLM combines supervised

pedagogical demonstrations with reinforcement learning from human feedback (RLHF), using human judgments of pedagogical instruction following to train its reward model (LearnLM Team et al., 2024). Following distillation of GPT-4o tutoring responses, Scarlatos et al. (2025b) construct direct preference optimization (DPO) pairs from a weighted combination of predicted next-turn correctness and GPT-4o pedagogical scores. For conversation-level objectives, PPDPP trains a dialogue policy planner through self-play and reinforcement learning, keeping the tutor and student language models fixed (Deng et al., 2024). Online reinforcement learning samples complete dialogues with a fixed student model and rewards post-dialogue success together with pedagogical quality (Dinucu-Jianu et al., 2025). An offline route learns a policy over high-level tutoring actions from static dialogue data, using compact student-state representations to select actions that an LLM realizes as utterances (Nam et al., 2025).

Student simulation. Realistic simulation requires reproducing students' errors and maintaining their behavioral characteristics across dialogue turns. Prompted personas specify student backgrounds and behavior for interactive teacher practice (Markel et al., 2023). Training on recorded student turns supplies direct behavioral supervision, and reference-based DPO uses fidelity scores to rank candidate responses. Comparisons across linguistic, behavioral, and cognitive dimensions show improvements from fine-tuning, with reproducing specific student errors remaining difficult (Scarlatos et al., 2026). Multi-turn reinforcement learning addresses consistency by rewarding adherence to the assigned persona and agreement with earlier utterances, keeping the task agent fixed (Abdulhai et al., 2025). Self-play methods use model-generated data for iterative language-model training. SPIN contrasts supervised reference responses with responses generated by earlier model iterations (Chen et al., 2024). SPAG uses outcomes from self-play in the adversarial language game Adversarial Taboo as feedback for offline reinforcement learning (Cheng et al., 2024).

Dialogue knowledge tracing. Knowledge tracing estimates student mastery from the history of learning interactions. Probabilistic skill models and recurrent neural models infer knowledge states from exercise–response sequences (Corbett & Anderson, 1995; Piech et al., 2015). Later methods use concept-indexed memory or attention to retrieve information from past interactions (Zhang et al., 2017; Ghosh et al., 2020; Choi et al., 2020), and question-specific representations distinguish exercises associated with the same skill (Liu et al., 2023). Dialogue observations additionally require identifying the skills elicited by a tutor and interpreting evidence of mastery in free-form student responses. LLMKT estimates mastery and turn-level correctness from dialogue histories and LLM-annotated knowledge components. DKT-Sem incorporates semantic representations of dialogue turns and knowledge components into recurrent knowledge tracing (Scarlatos et al., 2025a).

3 Problem Formulation

We study dialogue tutoring for mathematical problem solving. Each instance consists of a problem q, a reference solution $g = (g_{\text{steps}}, g_{\text{ans}})$, and an initial misconception profile e_0. A Tutor with policy π_ψ interacts with a target student whose response distribution π_S^* is unknown, producing a dialogue $d = (t_1, s_1, \ldots, t_M, s_M)$ with $M \leq M_{\max}$. After the dialogue, the student produces a final answer $\hat{a}$ without further guidance. The tutoring objective is the expected exact-answer success

$$J_{\text{task}}(\psi) = \mathbb{E}_{(d,\hat{a})\sim(\pi_\psi,\pi_S^*)}\big[\mathbb{1}[\hat{a} = g_{\text{ans}}]\big], \tag{1}$$

pursued without directly disclosing the answer. We adapt MathDial's tutoring-move taxonomy by separating strategy hinting from direct answer disclosure (Macina et al., 2023). The resulting categories are attention focusing, probing, strategy hinting, direct answer disclosure, and generic response. The direct-answer-disclosure move denotes revelation of the final answer or a complete key derivation. These move categories enter the pedagogical reward in Section 4.2.

Let $x_m = (\mathcal{P}, t_1, s_1, \ldots, t_{m-1}, s_{m-1})$ denote the dialogue state before Tutor utterance t_m, where $\mathcal{P}$ is the initial dialogue context. The history preceding the student response is $h_m = (x_m, t_m)$. During training, the Tutor $\pi_\psi(t_m \mid x_m)$ interacts with a learned Student Simulator $\pi_\phi(s_m \mid h_m)$, inducing the history distribution $\mu_{\psi,\phi}$.

4 METHOD

Simulated student responses play two roles in dialogue tutoring: they become part of subsequent dialogue states and provide correctness feedback for comparing Tutor actions. A scalar correctness score is a many-to-one summary of a response, so identical immediate feedback can accompany different successor histories. The learning problem therefore requires feedback that remains informative as instructional contexts change and a response model that preserves errors and incomplete reasoning beyond correctness alone. Accordingly, we propose COMET, which co-evolves the Tutor and Student Simulator while keeping instructional utility and student-response modeling as distinct objectives (Figure 2).

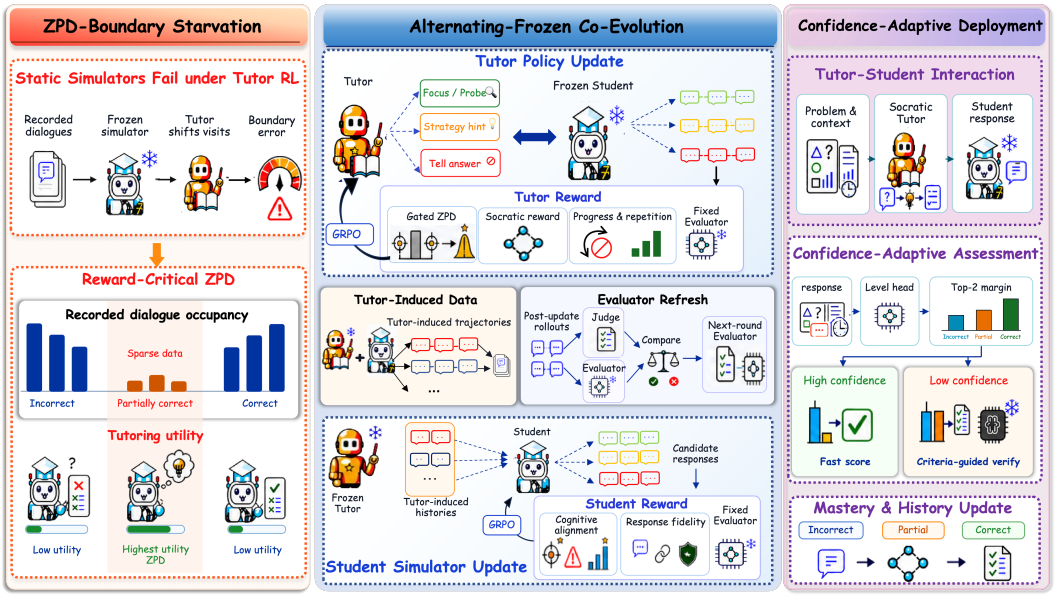


Figure 2: Tutor–simulator co-evolution in COMET. Simulated responses provide Tutor feedback and form subsequent dialogue histories. The updated Tutor induces the histories used for simulator adaptation, and the evaluator is refreshed between rounds with confidence-adaptive scoring.

4.1 TUTOR–STUDENT CO-EVOLUTION

At a nonterminal dialogue state, the joint turn distribution factorizes as $p_{\psi,\phi}(t_m, s_m \mid x_m) = \pi_\psi(t_m \mid x_m)\pi_\phi(s_m \mid h_m)$. With ϕ fixed, the Tutor policy ψ determines the induced history distribution $\mu_{\psi,\phi}$ on which simulator responses are required. At a fixed dialogue state and evaluator, ϕ specifies the distribution of simulated responses used to score each candidate Tutor action.

In round r, Phase A holds the preceding simulator $\pi_\phi^{(r-1)}$ and evaluator $\mathcal{E}^{(r-1)}$ fixed during the Tutor update. Candidate actions at a given state are thus compared under a common response model and cognitive scoring function. Let $\nu_A^{(r)}$ denote the distribution of Tutor training states and $\mathcal{B}_T$ the batch of Tutor utterances containing candidate t. The Tutor objective is

$$J_T^{(r)}(\psi) = \mathbb{E}_{x\sim\nu_A^{(r)},\, t\sim\pi_\psi(\cdot|x)}\left[R_T\left(x, t, \mathcal{B}_T; \pi_\phi^{(r-1)}, \mathcal{E}^{(r-1)}\right)\right]. \tag{2}$$

After the Tutor update, COMET collects histories by pairing $\pi_\psi^{(r)}$ with $\pi_\phi^{(r-1)}$. Their distribution is $\mu_{\psi^{(r)},\phi^{(r-1)}}$, and Phase B adapts the simulator on the collected histories while holding the updated Tutor and $\mathcal{E}^{(r-1)}$ fixed. For each history h, s^{ref} denotes the assigned recorded student turn and $\mathcal{G}_S$ the group of sampled responses containing $\hat{s}$. The simulator objective is

$$J_S^{(r)}(\phi) = \mathbb{E}_{h\sim\mu_{\psi^{(r)},\phi^{(r-1)}},\, \hat{s}\sim\pi_\phi(\cdot|h)}\left[R_S\left(h, \hat{s}, s^{\text{ref}}, \mathcal{G}_S; \mathcal{E}^{(r-1)}\right)\right]. \tag{3}$$

The task objective in Eq. (1) evaluates exact-answer success after a complete dialogue. Equations (2) and (3) define conditional turn-level training objectives whose expectations include the sampled responses and candidate groups used to compute each reward. Both policies start from their preceding-round parameters and are updated using KL-regularized Group Relative Policy Optimization (GRPO) (Shao et al., 2024). After Phase B, the evaluator is refreshed, so the next Tutor phase uses the updated simulator and evaluator.

4.2 Validation-Gated Tutor Optimization

The Tutor objective distinguishes challenges estimated to be within the student's reach from answer disclosure by evaluating predicted response correctness and the instructional move separately. Given a candidate Tutor utterance t_m and history $h_m = (x_m, t_m)$, COMET samples N responses from the frozen simulator. The evaluator assigns each response a correctness level $\ell^{(j)} = \ell_{\mathcal{E}^{(r-1)}}(h_m, \hat{s}_m^{(j)}) \in \{0, 1, 2\}$, corresponding to incorrect, partially correct, or correct. The sample mean $\hat{y}_m$ estimates the expected normalized correctness level under this simulator and evaluator,

$$\hat{s}_m^{(j)} \sim \pi_\phi^{(r-1)}(\cdot \mid h_m), \quad j = 1, \ldots, N, \qquad \hat{y}_m = \frac{1}{N} \sum_{j=1}^{N} \frac{\ell^{(j)}}{2}. \tag{4}$$

The ZPD-inspired utility $u(x_m, t_m) = 1 - |2\hat{y}_m - 1|$ assigns its highest value to guidance with intermediate estimated response correctness. Let a_r denote the validation ROC-AUC of the frozen simulator used for Tutor optimization in round r. COMET uses it to select the resolution of the ZPD reward. Below θ_{warm}, candidates with estimates in $[0.2, 0.8]$ receive the same reward. Once the threshold is reached, the continuous utility distinguishes candidates by their distance from 0.5:

$$r_{\text{ZPD}} = \begin{cases} \mathbb{1}[\hat{y}_m \in [0.2, 0.8]], & a_r < \theta_{\text{warm}}, \\ u(x_m, t_m), & a_r \geq \theta_{\text{warm}}. \end{cases} \tag{5}$$

The interval reward is a thresholded form of the same utility, since $\mathbb{1}[\hat{y}_m \in [0.2, 0.8]] = \mathbb{1}[u(x_m, t_m) \geq 0.4]$. For $\hat{y}_m > 0.5$, the continuous utility decreases as estimated correctness rises. The progress term rewards sufficiently large increases in $\hat{y}_m$ between successive turns and becomes active when the validation gate switches r_{ZPD} to the continuous utility. The turn-level Tutor reward is

$$R_T = \lambda_{\text{ZPD}} r_{\text{ZPD}} + \lambda_{\text{ped}} r_{\text{ped}} + \lambda_{\text{prog}} r_{\text{prog}} - \lambda_{\text{rep}} r_{\text{rep}}(t_m; \mathcal{B}_T). \tag{6}$$

A correct student response can arise from reasoning or from repeating an answer supplied by the Tutor. The pedagogical reward r_{ped} therefore scores the instructional move separately from response correctness. It assigns higher scores to attention focusing and probing than to strategy hinting, and penalizes direct answer disclosure. The repetition penalty r_{rep} discourages near-duplicate Tutor utterances within a training batch (Appendix B.3). Training dialogues terminate when $\hat{y}_m$ reaches the stopping threshold or the turn budget is exhausted.

Fix a dialogue state and K Tutor candidates. Compare their reward vectors $\mathbf{R}, \mathbf{R}' \in \mathbb{R}^K$ under two sets of sampled student responses, with the evaluator, validation gate, and preceding correctness estimate fixed. Let $C = I_K - \mathbf{1}\mathbf{1}^\top / K$ remove the group mean. Assume that both reward vectors have group standard deviations at least $\sigma_{\min} > 0$ and that each candidate score gradient has norm at most L_π. At the sampling policy, the reward-dependent local GRPO surrogate gradient g satisfies

$$\|g(\mathbf{R}) - g(\mathbf{R}')\|_2 \leq \frac{L_\pi}{\sigma_{\min}} \frac{\|C(\mathbf{R} - \mathbf{R}')\|_2}{\sqrt{K}}, \tag{7}$$

The pedagogical and repetition terms are fixed for these candidates, so differences in sampled student responses reach the update through candidate-relative ZPD and, when active, progress rewards (Appendix B.5).

4.3 Simulator Adaptation on Tutor-Induced Histories

Immediate correctness-based Tutor feedback does not uniquely specify student-response behavior: distinct responses can receive the same feedback while inducing different successor histories (Appendix B.4). COMET therefore adapts the Student Simulator on Tutor-induced histories with an objective that combines cognitive assessment, agreement with recorded responses, and candidate diversity:

$$R_S(h, \hat{s}, s^{\mathrm{ref}}, \mathcal{G}_S; \mathcal{E}) = \lambda_{\mathrm{eval}} r_{\mathrm{eval}}(h, \hat{s}; \mathcal{E}) + R_{\mathrm{ref}}(h, \hat{s}, s^{\mathrm{ref}}) + R_{\mathrm{div}}(h, \hat{s}, \mathcal{G}_S; \mathcal{E}). \tag{8}$$

The cognitive term $r_{\mathrm{eval}} = \frac{1}{3}(\hat{\ell}/2 + \hat{e}_{\mathrm{err}} + \hat{e}_{\mathrm{know}})$ combines current-response correctness with evidence about reasoning errors and knowledge acquisition. The level $\hat{\ell}$ scores current-response correctness, $\hat{e}_{\mathrm{err}}$ estimates the score remaining after rubric-defined deductions for reasoning errors, and $\hat{e}_{\mathrm{know}}$ credits correct reuse of earlier steps and self-correction across turns. The Tutor uses $\hat{\ell}/2$ in its ZPD and progress terms. The reasoning-error and knowledge-acquisition scores remain specific to the simulator objective. Cognitive scoring therefore favors correct reasoning but does not by itself reproduce the mistakes and incomplete reasoning found in recorded dialogues. The reference term R_{ref} anchors the simulator to the assigned recorded response s^{ref} through linguistic, dialogue-act, and semantic agreement. The diversity term R_{div} penalizes repetitive responses within $\mathcal{G}_S$ and, in groups containing multiple correctness levels, rewards candidates whose level differs from the group mode (Appendix B.3).

4.4 Judge-Anchored Evaluation and Distillation

Within a co-evolution round, both policy updates require a common cognitive scoring map. Later rounds must incorporate reasoning patterns revealed by newly induced histories. COMET holds $\mathcal{E}^{(r-1)}$ fixed throughout round r. A fixed external judge J supplies reference judgments for refining the textual rubric $\mathcal{S}_{\mathrm{eval}}^{(r)}$, and the multi-task prediction head $h_\omega^{(r)}$ approximates rubric-based judgments from a frozen encoder representation. Let $\mathcal{E}^{(r)}(h, s)$ denote the scoring map obtained after round r, which returns a correctness level in $\{0, 1, 2\}$ together with scores for reasoning errors and knowledge acquisition. This approximation supports repeated candidate scoring. The scoring map itself remains fixed within each round.

After Phase B, examples on which the judge and prediction head agree support existing rubric rules. Recurring disagreements, together with their supporting trajectories, guide rubric refinement. A frozen local language model applies the revised rubric to label examples, and the prediction head is retrained on labels accumulated across rounds. Let $(\ell^*, e^*_{\mathrm{err}}, e^*_{\mathrm{know}})$ denote the rubric-based targets for an evaluator input. The head outputs $(p_\omega, \hat{e}_{\mathrm{err}}, \hat{e}_{\mathrm{know}})$, where p_ω is a distribution over correctness levels. Its training loss is

$$\mathcal{L}_h = \lambda_\ell \mathcal{L}_{\mathrm{CE}}(p_\omega, \ell^*) + \lambda_e \mathcal{L}_{\mathrm{MSE}}(\hat{e}_{\mathrm{err}}, e^*_{\mathrm{err}}) + \lambda_k \mathcal{L}_{\mathrm{MSE}}(\hat{e}_{\mathrm{know}}, e^*_{\mathrm{know}}). \tag{9}$$

For each candidate response, the margin between the two largest probabilities in p_ω determines whether the evaluator uses the head's prediction or obtains a score from the local rubric-conditioned model. Appendices A and D give the architecture, training settings, and routing thresholds.

5 Experiments

COMET couples Tutor optimization with student-response modeling. We evaluate tutoring effectiveness and transfer to an independent student model (Section 5.2), the roles of the two updates and their objectives (Section 5.3), correctness readouts without knowledge-component annotations (Section 5.4), and simulator-bias dynamics across co-evolution rounds (Section 5.5).

5.1 Experimental Setup

MathDial (Macina et al., 2023) contains tutoring dialogues in which human teachers guide LLM-simulated students through multi-step mathematics problems. We use its training split for COMET training and validation and its official test split for tutoring evaluation. CoMTA (Miller & DiCerbo,

2024) contains anonymized Khanmigo tutoring dialogues and is used only for student-modeling evaluation.

The Tutor and Student Simulator are initialized separately by fine-tuning Qwen3-8B on recorded Tutor and student turns. Baselines are prompted LLM Tutors from several families and an SFT-only Qwen3-8B Tutor; the independent-student evaluation also uses a prompted Socratic Tutor with the same backbone. Appendices B–C describe partitions, training, and evaluation protocols.

5.2 Tutoring Outcomes

Socratic tutoring aims to guide students toward a solution without supplying the answer. We find that high task success can coexist with detected disclosure: for several prompted tutors with high raw Success@15, successful dialogues containing a Tutor utterance that numerically matches the reference answer account for 6.8–17.5% of all evaluated dialogues (Table 1). To align evaluation with the Socratic objective, we define Clean Success@15 as the proportion of all dialogues in which at least three of five student responses sampled after a single Tutor turn numerically match the reference answer within 15 turns and no Tutor utterance contains such a numerical match. COMET achieves the highest Clean Success@15 (85.3%) with a Telling Rate of 1.56% (Figure 3). The SFT Tutor has the lowest Telling Rate but also the lowest raw and clean success. This contrast shows why disclosure control and successful problem solving should be evaluated jointly.

Table 1: Tutoring outcomes on the MathDial test set (599 dialogues).

Method	Base	S@15	Tell.	Clean	Succ.+Tell	Turns	ZoneCov
GPT-4o	prop.	0.907	5.25%	0.838	6.8%	4.30	12.2%
Claude Sonnet 4.5	prop.	0.923	11.45%	0.773	15.0%	3.60	15.9%
Claude Opus 4.5	prop.	0.938	15.03%	0.763	17.5%	2.82	18.5%
DeepSeek V3	671B MoE	**0.948**	8.06%	0.828	12.0%	3.95	15.4%
Qwen 2.5-72B	72B	0.836	6.63%	0.788	4.8%	4.04	10.9%
Qwen3-8B-Inst.	8B	0.649	2.82%	0.601	4.8%	8.23	9.8%
SFT-Qwen3-8B	8B	0.406	**0.17%**	0.397	0.8%	5.87	4.7%
COMET	8B	0.883	1.56%	**0.853**	3.0%	4.21	**27.3%**

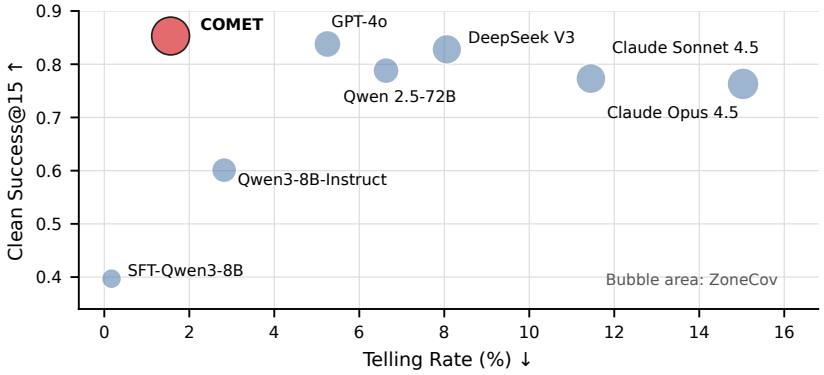


Figure 3: Clean Success@15 versus Telling Rate for the tutors in Table 1. Marker size increases with ZoneCov.

To assess transfer beyond the co-trained interaction partner, we conduct a separate final-answer evaluation with an independent frozen Llama-3.1-8B student under a shared answer-disclosure guard. COMET achieves 67.7% Success@15, exceeding the same-backbone Prompted-Socratic and SFT Tutors by 20.1 and 27.4 percentage points, respectively (Table 2). Both paired 95% confidence intervals exclude zero (Appendix C.4), supporting transfer of the learned guidance to a student that did not participate in co-evolution.

Table 2: Independent frozen Llama-3.1-8B student; tutors share Qwen3-8B, information access, 15 turns, no reference solutions, and the same disclosure guard (169 problems, 3 seeds).

Tutor	Success@15 (%)
COMET	**67.7**
Prompted-Socratic	47.5
SFT Tutor	40.2

5.3 Ablation Study

To identify which parts of the coupled objective produce the tutoring outcomes in Section 5.2, we evaluate ablations under the same MathDial test protocol and metrics as Table 1. Each variant modifies only round five from a shared four-round co-evolved Tutor–simulator pair and is evaluated with its resulting Tutor–simulator pair (Table 3).

Table 3: Ablations on the MathDial test set (599 dialogues). Intervals are Wilson 95% confidence intervals for Clean Success@15. Other columns follow Table 1.

Variant	S@15	Clean	95% CI	Tell.	ZoneCov	Turns
Full COMET	**0.883**	**0.853**	[0.822, 0.879]	**1.56%**	**27.3%**	4.21
Co-evolution phases						
w/o Phase A	0.830	0.780	[0.745, 0.811]	3.33%	19.5%	4.29
w/o Phase B	0.820	0.740	[0.703, 0.773]	6.57%	25.4%	4.43
Tutor rewards						
w/o r_{ZPD}	0.780	0.720	[0.682, 0.754]	2.89%	17.7%	6.96
w/o r_{ped}	0.810	0.740	[0.703, 0.773]	3.45%	18.2%	4.58
Binary r_{ped}	0.790	0.770	[0.734, 0.802]	2.07%	22.3%	4.02
w/o r_{prog}	0.826	0.800	[0.766, 0.830]	2.19%	15.0%	4.44
Simulator rewards						
w/o r_{eval}	0.793	0.745	[0.708, 0.778]	4.20%	23.0%	4.42
w/o $R_{\mathrm{ref}}, R_{\mathrm{div}}$	0.811	0.795	[0.760, 0.825]	2.05%	26.2%	4.25

The student-side results highlight the importance of the adaptation objective: omitting Phase B reduces Clean Success@15 to 74.0% and raises the Telling Rate to 6.57%, whereas retaining Phase B without cognitive evaluation yields 74.5% Clean Success@15, close to omitting adaptation altogether. Jointly removing R_{ref} and R_{div} also lowers clean success. On the Tutor side, removing r_{ZPD} or r_{ped} yields 72.0% and 74.0% Clean Success@15, below 78.0% without Phase A. In these round-five comparisons, an incomplete instructional reward makes continued Tutor optimization worse than skipping the update. The ZPD ablation produces the longest dialogues, whereas removing r_{ped} increases Telling Rate and replacing it with a binary reward only partly recovers clean success. Removing r_{prog} instead lowers ZoneCov from 27.3% to 15.0%. These contrasts are consistent with complementary roles for challenge shaping, pedagogical constraints, and progress incentives.

5.4 Student-Response Prediction and Assessment

COMET's correctness feedback serves two purposes: predicting the correctness of a response elicited by a Tutor action and assessing an observed response. We evaluate two readouts. The generative readout aggregates evaluator-assigned scores over candidate responses, whereas the discriminative readout uses the Level head's expected normalized level for an available response. Figure 4 reports these readouts under their evaluation protocols; Appendix C.3 specifies the inputs.

Without knowledge-component supervision, the generative readout achieves a MathDial ROC-AUC of 0.895, compared with 0.767 for LLMKT, a difference of 12.8 percentage points. The discriminative readout achieves 0.781 on MathDial by scoring the observed response with its dialogue context and reference solution. Figure 4 also includes the CoMTA readouts. These results provide separate evidence for prediction from simulated candidates and assessment of observed responses.

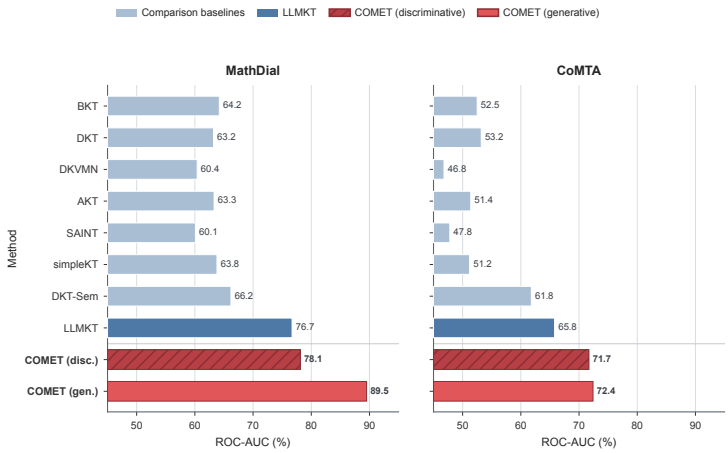


Figure 4: Turn-level correctness ROC-AUC under the protocols in Appendix C.3.

5.5 Simulator Error and Co-evolution Dynamics

Because the ZPD utility is maximized near intermediate correctness, error in this region is relevant to the feedback used for Tutor optimization. Across co-evolution rounds, validation Success@15 rises from 59.0% at $R = 0$ to 88.9% at $R = 5$, while mean dialogue length falls from 9.00 to 4.06 turns (Table 4). In the previously identified central bin $[0.4, 0.6)$, where the initial simulator has its largest MSE, the positive mean bias falls from $+0.133$ to $+0.082$, an approximately 38% reduction in overestimation (Figure 1; Appendix C.1). ZoneCov also rises from 8.2% to 29.4%, providing a corresponding diagnostic of target-range coverage. At $R = 7$, central-bin bias rises to 0.088 and Success@15 falls to 87.1%, despite a further reduction in Telling Rate to 3.17%. Relative to initialization, tutoring outcomes and central-bin estimation therefore improve concurrently, whereas the later reduction in disclosure does not coincide with higher tutoring success.

Table 4: Simulator bias and tutoring outcomes on MathDial validation across co-evolution rounds. Central-bin $|\text{bias}|$ is computed on validation turns with smoothed reference correctness in $[0.4, 0.6)$.

Metric	R=0	R=3	**R=5**	R=7
Central-bin $\lvert\text{bias}\rvert$	0.133	0.097	**0.082**	0.088
Success@15	0.590	0.673	**0.889**	0.871
Mean turns	9.00	6.02	**4.06**	4.87
ZoneCov	8.2%	15.1%	**29.4%**	26.3%
Telling Rate	0.08%	4.71%	3.58%	3.17%

6 Conclusion

COMET addresses the coupled learning of Tutor policies and student-response models in ZPD-inspired dialogue tutoring. Tutor updates alter the histories on which simulated responses are needed, and those responses shape both instructional reward estimates and subsequent dialogue states. Correctness-based optimization alone does not specify how a simulator should reproduce students' mistakes and incomplete reasoning, making behavioral fidelity a separate modeling requirement. COMET therefore optimizes instructional utility and student-response modeling under distinct objectives. Tutor-first co-evolution ties simulator adaptation to the histories induced by policy improvement, with judge-anchored cognitive evaluation and reference-based behavioral constraints jointly defining what the simulator learns. For a fixed group of Tutor candidates, our analysis bounds how changes in their relative simulator-dependent rewards perturb the local GRPO surrogate gradient. The results show reduced overestimation at intermediate reference correctness and improved tutoring with an independent frozen student. These findings support treating student-response modeling as an optimization objective alongside instructional progress in ZPD-guided tutoring.

AI USE STATEMENT

Generative AI tools were used to assist with language editing and analysis of experimental results, and to check the manuscript for errors and inconsistencies.

REFERENCES

Marwa Abdulhai, Ryan Cheng, Donovan Clay, Tim Althoff, Sergey Levine, and Natasha Jaques. Consistently simulating human personas with multi-turn reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 52920–52957. Curran Associates, Inc., 2025. doi: 10.52202/085713-1766. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/4c91443877f8388d8190c938ac5a4d4d-Paper-Conference.pdf.

Benjamin S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13(6):4–16, 1984. doi: 10.3102/0013189X013006004.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6621–6642. PMLR, 2024. URL https://proceedings.mlr.press/v235/chen24j.html.

Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. Self-playing adversarial language game enhances LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 37, pp. 126515–126543, 2024. doi: 10.52202/079017-4019.

Youngduck Choi, Youngnam Lee, Junghyun Cho, Jineon Baek, Byungsoo Kim, Yeongmin Cha, Dongmin Shin, Chan Bae, and Jaewe Heo. Towards an appropriate query, key, and value computation for knowledge tracing. In *Proceedings of the Seventh ACM Conference on Learning @ Scale*, pp. 341–344. ACM, 2020. doi: 10.1145/3386527.3405945.

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. doi: 10.1007/BF01099821.

Yang Deng, Wenxuan Zhang, Wai Lam, See-Kiong Ng, and Tat-Seng Chua. Plug-and-play policy planner for large language model powered dialogue agents. In *International Conference on Learning Representations*, volume 2024, pp. 10058–10090, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/29e8437db7b549160ce03d336ff66f65-Abstract-Conference.html.

David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 272–292. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.15. URL https://aclanthology.org/2025.emnlp-main.15/.

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

LearnLM Team, Abhinit Modi, et al. LearnLM: Improving Gemini for learning, 2024. URL https://arxiv.org/abs/2412.16429.

Ben Liu, Jihai Zhang, Fangquan Lin, Xu Jia, and Min Peng. One size doesn't fit all: A personalized conversational tutoring agent for mathematics instruction. In *Companion Proceedings of the ACM on Web Conference 2025*, pp. 2401–2410. ACM, 2025. doi: 10.1145/3701716.3717527. URL https://doi.org/10.1145/3701716.3717527.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. SocraticLM: Exploring Socratic personalized teaching with large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 85693–85721, 2024. doi: 10.52202/079017-2721.

Zitao Liu, Qiongqiong Liu, Jiahao Chen, Shuyan Huang, and Weiqi Luo. simpleKT: A simple but tough-to-beat baseline for knowledge tracing. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2302.06881.

Jakub Macina, Nico Daheim, Sankalan Pal Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5602–5621, 2023. doi: 10.18653/v1/2023.findings-emnlp.372. URL https://arxiv.org/abs/2305.14536.

Julia M. Markel, Steven G. Opferman, James A. Landay, and Chris Piech. GPTeach: Interactive TA training with GPT-based students. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, pp. 226–236. ACM, 2023. doi: 10.1145/3573051.3593393.

Pepper Miller and Kristen DiCerbo. LLM based math tutoring: Challenges and dataset, 2024. URL https://doi.org/10.35542/osf.io/5zwv3. EdArXiv preprint; accompanying CoMTA dataset: https://github.com/Khan/tutoring-accuracy-dataset.

Hyunji Nam, Omer Gottesman, Amy Zhang, Dean Foster, Emma Brunskill, and Lyle Ungar. Efficient RL for optimizing conversation level outcomes with an LLM-based tutor, 2025. URL https://arxiv.org/abs/2507.16252.

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

Alexander Scarlatos, Ryan S. Baker, and Andrew Lan. Exploring knowledge tracing in tutor-student dialogues using LLMs. In *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, pp. 249–259. ACM, 2025a. doi: 10.1145/3706468.3706501. URL https://arxiv.org/abs/2409.16490.

Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. Training LLM-based tutors to improve student learning outcomes in dialogues. In *Artificial Intelligence in Education*, volume 15877 of *Lecture Notes in Computer Science*, pp. 251–266. Springer Nature Switzerland, 2025b. doi: 10.1007/978-3-031-98414-3_18. URL https://arxiv.org/abs/2503.06424.

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 (Volume 1: Long Papers)*, pp. 42349–42385, 2026. doi: 10.18653/v1/2026.acl-long.1960. URL https://arxiv.org/abs/2601.04025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Lev S. Vygotsky. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA, 1978.

Rose Wang, Qingyang Zhang, Carly Robinson, Susanna Loeb, and Dorottya Demszky. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 2174–2199, 2024. doi: 10.18653/v1/2024.naacl-long.120.

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.

APPENDIX CONTENTS

A Evaluator Implementation 14
A.1 Multi-task Prediction Head 14
A.2 Evaluation Rubric and Updates 14
B Training and Implementation Details 15
B.1 COMET Co-evolution Algorithm 15
B.2 Training Setup 15
B.3 Reward Components 15
B.4 Correctness Feedback and Student Behavior 16
B.5 Candidate-Relative Feedback and Local Tutor Updates 16
C Additional Analyses and Evaluation Protocols 17
C.1 Error Decomposition by Correctness Bin 17
C.2 Number of Student Samples 17
C.3 Knowledge-Tracing Evaluation 17
C.4 Independent-Student Evaluation 18
C.5 Comparison with Prior Protocols 18
D Deployment and Evaluation Efficiency 18
D.1 Confidence-Based Routing 18
D.2 Routing Efficiency 18
E Case Study: Resolving a Quantity Misconception 19

A Evaluator Implementation

A.1 Multi-task Prediction Head

Each evaluator input is encoded by the frozen Qwen3-8B backbone of the Student Simulator, using the last-layer hidden state of the final non-padding token as a 4096-dimensional representation. Three independent two-layer MLPs, each with a 256-dimensional GELU hidden layer and dropout of 0.1, predict the correctness level, the error score, and the knowledge-acquisition score. The heads contain approximately 3.15M trainable parameters.

The heads are trained with the objective $\mathcal{L}_h$ in Section 4.4. The classification loss uses inverse-square-root class weights computed on the training split. During the first three epochs, only the correctness and error losses are active, with $\lambda_\ell = 1.0$ and $\lambda_e = 0.5$. The knowledge-acquisition loss is then added with $\lambda_k = 0.1$. Training uses AdamW with a learning rate of 5×10^{-5}, a batch size of 64, and at most 10 epochs, and stops after three epochs without improvement in the held-out composite score.

The external judge J is GPT-4o. It provides the initial supervision and the reference judgments used to revise the rubric. A frozen local Qwen3-8B model receives the current rubric $\mathcal{S}_{\text{eval}}^{(r)}$ as its system prompt, generates training labels for h_ω, and scores low-confidence responses. After each round, the heads are retrained on the accumulated labels with the encoder kept frozen.

A.2 Evaluation Rubric and Updates

The rubric $\mathcal{S}_{\text{eval}}$ defines the criteria for assigning correctness levels $\ell^* \in \{0, 1, 2\}$, deducting penalties for reasoning errors, and scoring knowledge acquisition. Its rules also distinguish direct answer disclosure from scaffolding, and each rule has a stable identifier.

The error-score target e^*_{err} starts at 1 and is reduced by penalties for arithmetic, unit, operation, and constraint-handling errors. The knowledge-acquisition target e^*_{know} starts at 0.5 and increases for correct reuse of prior knowledge, conceptual improvement, and self-correction. The score decreases when a response contradicts the student's earlier correct reasoning. Both scalar targets are clipped to the interval $[0, 1]$.

After Phase B, rollout examples are split into agreement and disagreement sets by comparing h_ω with the external judge. Agreement cases support existing rules, and recurring disagreements yield candidate refinements. A refinement is accepted only if it is supported by trajectories and remains compatible with the previous rubric.

B Training and Implementation Details

B.1 COMET Co-evolution Algorithm

Algorithm 1 Tutor-first co-evolution of the Tutor, Student Simulator, and evaluator.

Input: base model θ_0; training corpus $\mathcal{D}_{\text{train}}$; rounds R; Tutor candidates K; student samples N; gate threshold θ_{warm}; fixed external judge J.

Output: Tutor $\pi_\psi^{(R)}$, Student Simulator $\pi_\phi^{(R)}$, and evaluator $\mathcal{E}^{(R)} = (\mathcal{S}_{\text{eval}}^{(R)}, h_\omega^{(R)})$.

1. Initialize $\pi_\psi^{(0)}$ and $\pi_\phi^{(0)}$ by supervised fine-tuning on the Tutor and student turns of $\mathcal{D}_{\text{train}}$.
2. Initialize $\mathcal{S}_{\text{eval}}^{(0)}$ with J, and train $h_\omega^{(0)}$ on its labels.
3. **For** $r = 1, \ldots, R$:
   (a) **Phase A.** Initialize the Tutor from $\pi_\psi^{(r-1)}$, and hold $\pi_\phi^{(r-1)}$ and $\mathcal{E}^{(r-1)}$ fixed.
   (b) Repeat until the Telling Rate stopping criterion is met: sample K Tutor candidates for each training dialogue state; sample N responses from $\pi_\phi^{(r-1)}$ for each candidate; compute $\hat{y}_m$, r_{ZPD}, and R_T with $\mathcal{E}^{(r-1)}$; update the Tutor with KL-regularized GRPO.
   (c) Save the Tutor as $\pi_\psi^{(r)}$.
   (d) **Phase B.** Generate trajectories $\mathcal{T}^{(r)}$ with $\pi_\psi^{(r)}$ and $\pi_\phi^{(r-1)}$. Initialize the simulator from $\pi_\phi^{(r-1)}$, and hold $\pi_\psi^{(r)}$ and $\mathcal{E}^{(r-1)}$ fixed.
   (e) For each batch from $\mathcal{T}^{(r)}$: sample candidate responses for each student-turn history; compute R_S with $\mathcal{E}^{(r-1)}$; update the simulator with KL-regularized GRPO.
   (f) Save the simulator as $\pi_\phi^{(r)}$.
   (g) **Evaluator update.** Compare h_ω with J on rollout examples, and split them into agreement and disagreement sets $\mathcal{D}^+$ and $\mathcal{D}^-$.
   (h) Refine the rubric from recurring disagreements, retaining rules supported by agreement cases, to obtain $\mathcal{S}_{\text{eval}}^{(r)}$.
   (i) Label examples under $\mathcal{S}_{\text{eval}}^{(r)}$ and retrain $h_\omega^{(r)}$.
4. **Return** $\pi_\psi^{(R)}, \pi_\phi^{(R)}, \mathcal{E}^{(R)}$.

B.2 Training Setup

MathDial's 2,262 training dialogues are divided into 2,045 for optimization and 217 for validation. The official test set contains 599 dialogues. Both policies are initialized from Qwen3-8B and fine-tuned on their role-specific MathDial turns for three epochs with bf16 precision and a learning rate of 2×10^{-5}. In Phase A, each dialogue state has $K = 4$ Tutor candidates, and $N = 5$ student responses are sampled for each candidate. Training dialogues contain at most $M_{\max} = 15$ turns and end once $\hat{y}_m \geq 0.8$. The ZPD reward switches from the interval form to the continuous utility once the simulator's validation ROC-AUC reaches $\theta_{\text{warm}} = 0.65$.

Both phases use a learning rate of 1×10^{-6} and a KL coefficient of 0.01. Phase A measures the Telling Rate over consecutive windows of 100 steps and stops when its relative change stays below 10% for three consecutive checks.

The validation split of 217 dialogues contains 1,316 student turns. It is used for the validation gate and evaluator assessment and contributes no gradient updates. All experiments run on one node with four NVIDIA A800 80GB GPUs.

B.3 Reward Components

The Tutor reward in Eq. (6) uses the weights

$$R_T = 0.3\, r_{\text{ZPD}} + 0.3\, r_{\text{ped}} + 0.3\, r_{\text{prog}} - 0.1\, r_{\text{rep}}.$$

A frozen tutoring-move classifier labels each Tutor utterance. The tiered Socratic reward r_{ped} is 1.0 for attention focusing or probing, 0.3 for strategy hinting, 0 for a generic response, and -1.0 for

direct answer disclosure. After the validation gate is passed, the progress reward is zero on the first turn and, when a preceding estimate is available, is

$$r_{\text{prog}} = \mathbb{1}[\hat{y}_m > \hat{y}_{m-1} + 0.1].$$

The repetition penalty is the excess ROUGE-L overlap with the most similar Tutor utterance in the same GRPO batch:

$$r_{\text{rep}}(t_i) = \max\Big(0, \max_{j \neq i} \text{ROUGE-L}(t_i, t_j) - 0.7\Big).$$

The Student Simulator reward in Eq. (8) uses

$$R_S = 0.5\, r_{\text{eval}} + R_{\text{ref}} + R_{\text{div}}, \qquad r_{\text{eval}} = \tfrac{1}{3}\big(\hat{\ell}/2 + \hat{e}_{\text{err}} + \hat{e}_{\text{know}}\big),$$
$$R_{\text{ref}} = 1.0\, r_{\text{lang}} + 0.5\, r_{\text{acts}} + 0.2\, r_{\text{anchor}}, \qquad R_{\text{div}} = 0.1\, r_{\text{disc}} - 0.6\, r_{\text{col}}.$$

When a recorded behavioral reference is available, the reference-based terms compare each candidate with that response. The language reward $r_{\text{lang}} = 0.6\,\text{BERTScore-F1} + 0.4\,\text{ROUGE-L}$ is computed against the assigned reference, and r_{acts} and r_{anchor} reward dialogue-act and semantic agreement with it. In groups with mixed correctness, r_{disc} rewards candidates whose correctness level differs from the group mode. The penalty r_{col} applies when a candidate's mean Sentence-BERT similarity to the other candidates exceeds 0.85. All coefficients are fixed across rounds.

B.4 Correctness Feedback and Student Behavior

Fix a response history $h = (x, t)$ with at least one further turn available, the Tutor candidate batch, the preceding correctness estimate, the validation gate, and the evaluator $\mathcal{E}$. The simulator induces a distribution over the three correctness levels,

$$P^{\ell}_{\phi,\mathcal{E}}(k \mid h) = \sum_{s:\, \ell_{\mathcal{E}}(h,s)=k} \pi_\phi(s \mid h), \qquad k \in \{0, 1, 2\}.$$

Suppose two distinct responses $s \neq s'$ both receive level 1. Simulators that deterministically produce s or s' give the same $\hat{y}_m = 0.5 < 0.8$, current-turn Tutor reward, and decision to continue. Their successor histories $(h, s) \neq (h, s')$ nevertheless differ. Thus immediate correctness-based tutoring feedback does not determine the simulator's response behavior.

B.5 Candidate-Relative Feedback and Local Tutor Updates

Fix a dialogue state x, a group of K Tutor candidates $\mathcal{B}_T = \{t_i\}_{i=1}^K$, the preceding correctness estimate, the validation gate, and evaluator $\mathcal{E}$. Let $\mathbf{R}$ and $\mathbf{R}'$ be two realized Tutor-reward vectors obtained by sampling student responses for these same candidates. With $C = I_K - \mathbf{1}\mathbf{1}^\top/K$, the pedagogical and repetition terms in Eq. (6) are identical across the two evaluations. Hence

$$C(\mathbf{R} - \mathbf{R}') = \lambda_{\text{ZPD}} C(\mathbf{r}_{\text{ZPD}} - \mathbf{r}'_{\text{ZPD}}) + \lambda_{\text{prog}} C(\mathbf{r}_{\text{prog}} - \mathbf{r}'_{\text{prog}}),$$

where the progress term applies when active. The reward vectors include the realized samples used to compute each candidate's $\hat{y}_m$.

For any reward vector with nonzero group standard deviation, define $\sigma(\mathbf{R}) = \|C\mathbf{R}\|_2/\sqrt{K}$ and $A(\mathbf{R}) = C\mathbf{R}/\sigma(\mathbf{R})$. At the Tutor policy ψ_0 used to sample the fixed candidates, let $v_i = |t_i|^{-1} \sum_j \nabla_\psi \log \pi_\psi(t_{i,j} \mid x, t_{i,<j})|_{\psi=\psi_0}$ be candidate i's mean token score gradient. The importance ratios equal one at ψ_0, so the reward-dependent part of the local GRPO surrogate gradient is $g(\mathbf{R}) = K^{-1} \sum_i A_i(\mathbf{R}) v_i$. The KL term is identical for $\mathbf{R}$ and $\mathbf{R}'$ at this fixed policy and cancels from their gradient difference (Shao et al., 2024).

Suppose $\sigma(\mathbf{R}), \sigma(\mathbf{R}') \geq \sigma_{\min} > 0$ and $\|v_i\|_2 \leq L_\pi$ for every candidate. Normalization of the centered vectors gives

$$\frac{\|A(\mathbf{R}) - A(\mathbf{R}')\|_2}{\sqrt{K}} \leq \frac{\|C(\mathbf{R} - \mathbf{R}')\|_2}{\sqrt{K}\, \sigma_{\min}}.$$

Indeed, for nonzero vectors u, v, $\|u/\|u\|_2 - v/\|v\|_2\|_2 \leq \|u - v\|_2 / \min(\|u\|_2, \|v\|_2)$. Applying Cauchy–Schwarz to $K^{-1} \sum_i [A_i(\mathbf{R}) - A_i(\mathbf{R}')] v_i$ yields Eq. (7). A common reward shift lies in the null space of C and therefore leaves the local reward-dependent gradient unchanged. Equal immediate feedback can still correspond to different successor histories, as shown in Appendix B.4.

C Additional Analyses and Evaluation Protocols

The coarse ZPD reward uses $\hat{y}_m \in [0.2, 0.8]$. In Table 1, ZoneCov averages the within-dialogue fraction of Tutor turns whose five-sample numerical answer-match rate lies in $[0.3, 0.7]$. The error analyses define the central bin by the reference correctness $y_m^{\mathrm{gt}} \in [0.4, 0.6)$.

C.1 Error Decomposition by Correctness Bin

We group the 1,316 MathDial validation turns into five bins of smoothed reference correctness y_m^{gt}. For the initial simulator, bias is the within-bin mean of the answer-matching estimate minus this reference score. Table C.1 decomposes the corresponding mean squared error into squared bias and residual variance.

For Figure 1, let $c_m \in \{0, 1\}$ indicate whether the recorded response passes the reference-answer matching rule, and let $\tilde{p}_m$ be the fraction of five simulated responses passing the same rule. The reference score follows $y_m^{\mathrm{gt}} = 0.5c_m + 0.5y_{m-1}^{\mathrm{gt}}$, initialized with c_1. Panels (a) and (b) average $(\tilde{p}_m - y_m^{\mathrm{gt}})^2$ and $(\tilde{p}_m - c_m)^2$, respectively. Intervals resample whole dialogues; excluding the initial state leaves subsequent reference scores unchanged.

Table C.1: Initial simulator error by bin of smoothed reference correctness on MathDial validation.

Bin	n	MSE	bias^2	variance	bias (R=0)	bias (R=5)	Δ bias
$[0, 0.2)$	825	0.069	0.009	0.060	$+0.093$	$+0.071$	-0.023
$[0.2, 0.4)$	84	0.108	0.008	0.100	-0.090	-0.114	-0.024
$[0.4, 0.6)$	**225**	**0.189**	**0.018**	**0.172**	$\mathbf{+0.133}$	$\mathbf{+0.082}$	$\mathbf{-0.051}$
$[0.6, 0.8)$	69	0.098	0.020	0.079	$+0.141$	$+0.031$	-0.110
$[0.8, 1.0]$	113	0.108	0.010	0.098	-0.102	-0.110	-0.009

The central bin had the highest initial MSE, with residual variance accounting for most of the error. After co-evolution, overestimation decreased in the central and adjacent upper bins. The two bins with negative initial bias moved further below zero.

C.2 Number of Student Samples

Table C.2 reports MathDial validation results for $N \in \{1, 3, 5, 8\}$ student samples per Tutor candidate.

Table C.2: Effect of the number of student samples per Tutor candidate on MathDial validation.

N	Success@15	Telling Rate	Mean turns	ZoneCov
1	0.843	4.78%	4.38	28.9%
3	0.829	4.42%	4.49	23.3%
5 (default)	**0.889**	3.58%	**4.06**	**29.4%**
8	0.862	**3.20%**	5.15	24.1%

The default $N = 5$ gave the highest validation success and ZoneCov, and the shortest dialogues. Increasing N to eight lowered the Telling Rate but reduced success and lengthened dialogues.

C.3 Knowledge-Tracing Evaluation

The CoMTA assessment uses the 153 non-calculus dialogues retained by LLMKT (Miller & DiCerbo, 2024; Scarlatos et al., 2025a); COMET is not trained on CoMTA.

The evaluation combines response prediction with response assessment. LLMKT predicts the correctness of s_m from the problem, preceding exchanges, and current Tutor utterance without access to s_m itself. The two COMET readouts are evaluated under their designated dataset-specific protocols. For the MathDial discriminative evaluation, the Level head receives the observed student response together with the dialogue context and reference solution; the resulting 0.781 score therefore assesses an observed response rather than a pre-response prediction.

For the LLMKT baseline, the turn-level correctness labels and their human-evaluation provenance follow Scarlatos et al. (Scarlatos et al., 2025a): GPT-4o generated the labels, and three former mathematics teachers assigned a mean accuracy score of 0.93. ROC-AUC is computed over student turns. The 95% confidence interval of the MathDial generative readout, [0.875, 0.916], is obtained from dialogue-level cluster-bootstrap resampling.

C.4 Independent-Student Evaluation

The matched comparison in Table 2 uses a frozen Llama-3.1-8B student and seeds 271828, 1729, and 314159, with the same 169 problems under each seed (507 dialogues per tutor). The tutors share the Qwen3-8B base model, information access, and a 15-turn budget, and none receives reference answers or solutions. Success@15 is pooled across seeds. Paired cluster-bootstrap intervals preserve problem-level pairing, and McNemar tests within each seed are Holm-corrected across the three seeds.

Table C.3: Paired Success@15 differences with the independent frozen student, in percentage points computed from unrounded results. The last column bounds the Holm-adjusted p-values across seeds.

Comparison	Difference	95% CI	Holm-adjusted p
COMET – Prompted-Socratic	$+20.1$	[13.6, 26.4]	≤ 0.0027
COMET – SFT	$+27.4$	[22.3, 32.5]	$\leq 3 \times 10^{-6}$

All three tutors pass through the same structural output guard, which blocks final-answer equivalents and complete derivations before a message reaches the student. Post-guard checks found no remaining violations of either type. A separate audit counts Tutor turns containing a reference-solution quantity that appears in neither the problem nor the student's initial attempt. This rate was 2–6% for COMET and the SFT Tutor across seeds, compared with 58% and 81% for reference-informed oracle-curriculum and direct-answer controls that bypass the guard.

C.5 Comparison with Prior Protocols

Prior tutoring studies differ in student model, interaction budget, and outcome measure, so their results are not directly comparable with Table 2. Table C.4 summarizes these protocols.

Table C.4: Evaluation protocols of prior tutoring studies.

Study	Evaluation protocol	Outcome measure
MathDial (Macina et al., 2023)	Tutoring an InstructGPT student for up to 10 turns	Success@10 (ChatGPT 53%, fine-tuned Flan-T5-780M 39%)
SocraticLM (Liu et al., 2024)	Human pairwise ranking of single-round responses against GPT-4	Normalized overall quality (SocraticLM 0.62; GPT-4 0.50)
Nam et al. (2025)	Claude 3 Sonnet student, 300 conversations per tutor	Conversation-level problem-solving success

D Deployment and Evaluation Efficiency

D.1 Confidence-Based Routing

For each candidate response, the Level head outputs probabilities over the three correctness levels. With $p_{(1)}$ and $p_{(2)}$ the largest and second-largest probabilities, the confidence is $c = p_{(1)} - p_{(2)}$, and the evaluator is

$$\mathcal{E}^{(r)}(\hat{s}, t_m, h_m) = \begin{cases} h_\omega^{(r)}(\mathrm{Enc}(\hat{s}, t_m, h_m, q, g_{\mathrm{steps}})), & c \geq \tau_{\mathrm{conf}}, \\ \mathrm{LLM}\Big(\hat{s}, t_m, h_m \mid \mathcal{S}_{\mathrm{eval}}^{(r)}\Big), & c < \tau_{\mathrm{conf}}. \end{cases}$$

The default threshold is $\tau_{\mathrm{conf}} = 0.50$. This configuration applies no separate filter for answer disclosure. The matched independent-student evaluation additionally applies the output guard described in Appendix C.4.

D.2 Routing Efficiency

Table D.1 gives the fast-path rate, expected scoring latency per response, and speedup at two thresholds. The fast-path rate is the fraction of responses for which the evaluator retains the prediction head's outputs.

Table D.1: Routing and expected evaluation latency on an NVIDIA A800. Speedups are relative to GPT-4o at 1.4 s per request.

τ_{conf}	Fast-path rate	Expected latency	Speedup vs. GPT-4o
0.50 (default)	**74.1%**	**117 ms**	$\approx \mathbf{12}\times$
0.65	33.9%	190 ms	$\approx 7\times$

Raising τ_{conf} from 0.50 to 0.65 sends more responses to the local rubric-conditioned model and increases expected scoring latency.

E Case Study: Resolving a Quantity Misconception

Table E.1 presents two exchanges from a COMET R5 dialogue on MathDial test (case 5000300_2_1). The student initially confuses the number of pets per friend with the total for a group, giving an answer of 80 instead of 32. The Tutor questions the repeated multiplication and then asks the student to count the two groups separately before adding Taylor's pets. The first displayed response still miscounts the group and gives 52. In the second response, the student computes $24 + 4 + 4 = 32$. Neither of the two Tutor utterances directly supplies the final answer. Of the five simulated answers shown for each exchange, two are correct after the first and all five after the second.

Table E.1: A tutoring dialogue and the corresponding sampled student answers. Dialogue excerpts are abridged.

Problem. Taylor and his friends decide to bring their pets together for a walk. Three of his friends come first, each having twice as many pets as Taylor has. Another two of his friends have two pets each. If Taylor has four pets, what is the total number of pets they have together? The correct total is $4 + 3(2 \times 4) + 2 \times 2 = 32$.

First exchange	
Tutor	Why does 3×24 work here? Can we fix just that mistake? What would the numbers look like? Remember to count every pet.
Student	Each friend has 8 pets, but I counted 6 people: $6 \times 8 = 48$, then $+4 = 52$. Does that sound right?
Samples	52, 32, 26, 76, 32 (2/5 correct)
Second exchange	
Tutor	How many friends had 8 pets? Then how many more friends are there, and what did they each have? Add Taylor's pets last.
Student	3 friends $\times 8 = 24$; two friends $\times 2 = 4$, plus Taylor's 4. $24 + 4 + 4 = 32$.
Samples	32, 32, 32, 32, 32 (5/5 correct)