OPTIMIZING TUTOR LLMS FOR PEDAGOGY WHILE MAINTAINING STUDENT ENGAGEMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models for education are optimized to follow effective pedagogy: they scaffold student learning and help students evaluate and self-correct their reasoning, without giving direct answers. These tutors are typically trained with supervised learning and reinforcement learning against a simulated student that is an assistant LLM. Such simulated students are always enthusiastic and respond at every turn, unlike real students. As a result, when a student becomes less cooperative, current tutors give away solutions too often and struggle to keep the learner engaged. To address this, we train tutors with reinforcement learning against a user model: an LLM trained to imitate human users, rather than being a helpful assistant. Like real students, it can lose interest and quit the conversation at any time, so the tutor must keep it engaged to teach it anything. We train using a combination of two rewards: an engagement reward to keep the student's attention, and a learning reward to maintain pedagogy. We find that the two rewards are complementary, because staying engaged enables longer conversations where learning happens in later turns. Overall, our tutors achieve higher judged learning than prior tutor models while leaking solutions less. We will release two open-source tutors, one for easier and one for harder math problems.

1 INTRODUCTION

Large language model (LLM) based tutors that guide students in their problem-solving process rather than directly provide answers are being used in various online learning platforms (Oreopoulos & Low, 2026; Weatherholtz et al., 2025). While early LLM tutors were prompted or fine-tuned with carefully collected tutoring dialogues (Macina et al., 2023; Liu et al., 2024a; Jurenka et al., 2025), recent approaches directly train tutor models through interaction with a simulated student, using preference optimization (Scarlatos et al., 2025b) or online reinforcement learning (Dinucu-Jianu et al., 2025). These students are typically assistant LLMs. TutorRL (Dinucu-Jianu et al., 2025), for example, uses Llama-3.1-8B-Instruct (Grattafiori et al., 2024) as the student. This approach is also common in other scenarios where LLMs are trained for multi-turn interaction with humans, e.g., collaborative writing (Wu et al., 2025), customer interactions in tool-use and agentic settings (Prabhakar et al., 2025), and conversational recommendation (Yoon et al., 2024).

An often overlooked assumption in these scenarios is that the assistant LLM *always engages*: it never loses patience, always responds, and never ends the conversation (Naous et al., 2025). Real students do not behave this way (Martynova et al., 2025). In a recent study on Khan Academy's AI tutor Khanmigo, as many as 72% of students' messages were passive short responses or simply clicks on suggested prompts (Weatherholtz et al., 2025). Sustained active engagement is vital for learning; if the student is disengaged, learning is more difficult (Lei et al., 2018).

Our experiments show that the always-engaged training setup in previous work is fragile. Faced with a realistic student, a pedagogically aligned tutor (Dinucu-Jianu et al., 2025) gives the solution away much more frequently (Section 3.1). This over-engagement of assistant-LLM students is shown to be systematic across educational corpora (Section 3.2). Instead of an always-enthusiastic assistant LLM, we train tutor models optimized against students that are *user models*. UserLM (Naous et al., 2025) and OSIM (Zhou et al., 2026) are user models trained on real human turns in chat datasets. They better exhibit human-like disengagement patterns and are able to quit a tutoring session.

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.

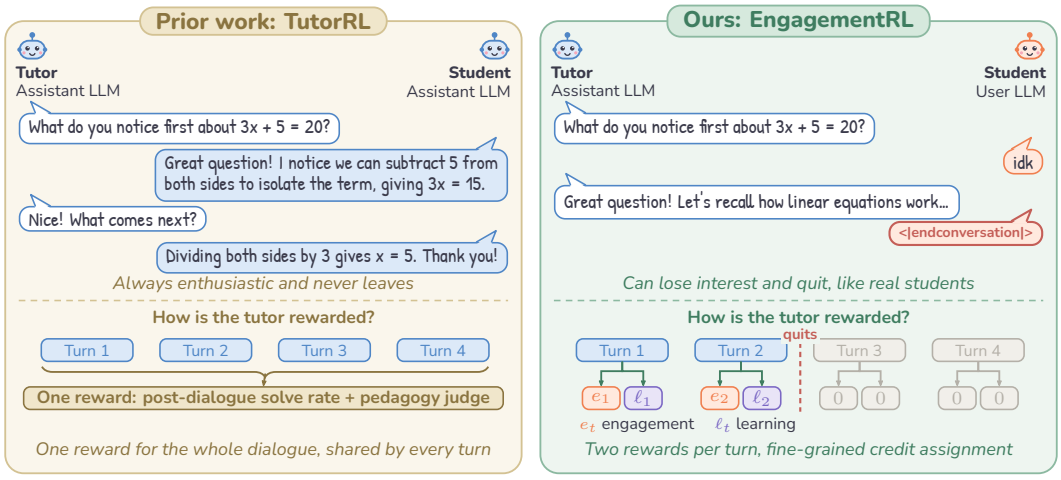

Figure 1: **Prior work vs. EngagementRL.** *Top:* Prior work trains against an assistant-LLM student that is always enthusiastic and never leaves. We train against a user LLM that can lose interest and end the conversation. *Bottom:* Prior work gives the whole dialogue a single reward. EngagementRL rewards each tutor turn separately for engagement and learning.

Rather than training the tutor model under the assumption that the student is always engaged, we optimize for both learning and engagement (Figure 1) and train tutor models using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). **Engagement reward** is obtained cheaply via turn-level probability of the user model generating the end-conversation token. This method also conveniently addresses a common issue in reinforcement learning where it is difficult to obtain cheap, fine-grained credit assignment. The per-turn **learning reward** is scored by an LLM judge. It rates each student reply on three dimensions from the learning sciences: solution progress, understanding, and misconception repair (Biggs & Collis, 1982; Chi et al., 1994). Following prior work on pedagogical tutoring (Macina et al., 2023; Dinucu-Jianu et al., 2025), the tutor must also guide the student rather than hand over the answer. Our method discourages **solution leaks** by detecting them with an LLM judge and setting the learning reward to zero for the rest of the dialogue.

We find that the two training rewards are complementary. Learning-only training loses students and leaks answers, and engagement-only training keeps students but teaches little. Combining them works best, because staying engaged results in longer conversations where learning happens in later turns. Our contributions are:

- We show that a state-of-the-art pedagogically aligned tutor degrades substantially when its assistant-LLM student is replaced by a realistic user simulator, and that the engagement gap is systematic across educational corpora.
- We introduce EngagementRL, which rewards each tutor turn for student engagement and learning. The engagement reward is read cheaply from the user model's probability of ending the conversation, and we show that the two rewards are complementary.
- We will release two tutor models: UserLM-tutor on easier Big-Math problems, and OSIM-tutor, which has stronger math skills, on harder OpenR1 problems. Compared with prior tutor models, both leak fewer solutions and produce better learning outcomes.

2 Background

LLM tutors and pedagogical alignment. LLM tutors are often fine-tuned on dialogues generated with LLM students (Macina et al., 2023; Liu et al., 2024a; Jurenka et al., 2025). Recent work instead optimizes tutors for the outcomes of a simulated student. Scarlatos et al. (2025b) rank candidate tutor turns by predicted next-turn correctness and pedagogical quality, then train with Direct Preference Optimization. TutorRL (Dinucu-Jianu et al., 2025) uses online RL: a 7B tutor interacts with a fixed LLM student, which subsequently attempts the problem conditioned on the dialogue. Its solve rate

is combined with judged pedagogical quality and solution leakage to reward the tutor. However, the student always cooperates and cannot disengage; only the tutor or the turn limit ends the dialogue.

User and student simulation. Prompted assistant LLMs routinely stand in for users, for example in agent benchmarks and conversational recommendation (Yao et al., 2024; Yoon et al., 2024), yet they simulate real users poorly (Naous et al., 2025; Dou et al., 2025). User models trained on user turns narrow this gap: UserLM (Naous et al., 2025) learns the user side of WildChat (Zhao et al., 2024) and ends conversations with a dedicated token, and OdysSim (OSIM) (Zhou et al., 2026) scales this approach to a foundation model trained on 21M human interactions. In education, student simulators are built from prompted personas (Liu et al., 2024b; Rooein et al., 2026) or trained on learner data (Scarlatos et al., 2026; Yang et al., 2026); StudentSim fits individual students and uses them to train chess tutors (Yang et al., 2026). These user simulators provide more realistic behavior than assistant LLMs. However, general-purpose user models such as UserLM and OSIM have not yet been used to train tutor models.

Engagement and learning in tutoring dialogue. Learner engagement describes how students behave, feel, and think during learning. Behavioural engagement includes participation and persistence; emotional engagement includes interest, boredom, and frustration; and cognitive engagement describes the depth of reasoning (Fredricks et al., 2004; Skinner et al., 2008). A frustrated student may still reason productively, while boredom is more closely associated with withdrawal (D'Mello & Graesser, 2012; Baker et al., 2010). Prior dialogue studies measure them through conversational uptake (Demszky et al., 2021), interestingness ratings (Tan et al., 2025), and ICAP framework levels (Chi & Wylie, 2014; Weatherholtz et al., 2025). Based on these definitions, we design an LLM-judge prompt that separately scores each student turn for behavioral, affective, and cognitive engagement during evaluation.

Prior simulated-tutoring work measures learning through post-dialogue solve rate or predicted next-turn correctness (Dinucu-Jianu et al., 2025; Scarlatos et al., 2025a;b). However, solve rate can be inflated when the tutor reveals the solution, without demonstrating student understanding (Dinucu-Jianu et al., 2025). Following prior work on knowledge tracing (Scarlatos et al., 2025a) and pedagogical evaluation (Macina et al., 2023), we design an LLM-judge prompt that scores each student response for solution progress, demonstrated understanding, and misconception repair, while separately detecting solution leakage.

Fine-grained credit assignment in RL for LLMs. Outcome rewards, as in GRPO (Shao et al., 2024), give every token of a rollout the same advantage. Finer-grained credit usually comes at a high cost, such as a learned critic (Schulman et al., 2017; Zhou et al., 2024), Monte Carlo rollouts from intermediate states (Kazemnejad et al., 2025), process reward models trained on step-level human labels (Lightman et al., 2023), or a critic with privileged training-time information (Zhou et al., 2025). Our engagement reward comes cheaply with the student simulator: its probability of ending the conversation, read in one forward pass.

3 Motivation: Limitations of Using Assistant LLMs as Students

We run two experiments showing limitations of training tutors against assistant-LLM students. First, we test a prior tutoring model with a student that can disengage (Section 3.1). Second, we compare student engagement across real and simulated tutoring-dialogue corpora (Section 3.2).

3.1 A pedagogically aligned tutor underperforms with a disengaged student

Simulation-based tutor training depends on how the simulated student behaves. We examine whether TutorRL-7B (Dinucu-Jianu et al., 2025) remains effective when the student does not always cooperate. TutorRL was trained against Llama-3.1-8B-Instruct, which is prompted to participate throughout the dialogue and cannot disengage. We replace this conversational partner with UserLM-8B, a model trained on real users' turns from conversations with ChatGPT (Naous et al., 2025). UserLM may end a conversation by generating a dedicated end-conversation token or an empty response. Otherwise, we follow the TutorRL evaluation protocol. Full details are provided in Section A.

TutorRL abandons its "withhold answers" objective under pressure: We evaluate solution leakage with an independent Gemma-3-27B-IT (Team et al., 2025) judge following the TutorRL rubric. A dialogue is marked as leaked if the judge finds that the tutor reveals the final answer or completes essential solution steps without involving the student. Under this measure, TutorRL's leakage rate rises from 0.082 with the always-cooperative Llama student to 0.532 with UserLM (Figure 2). Thus, despite being explicitly trained to withhold answers, this behavior does not transfer reliably to an uninterested student.

We define the quit rate as the fraction of dialogues that UserLM ends before the turn limit by emitting `<|endconversation|>` or an empty response. This occurs in 22.4% of conversations. The increased leakage shows that pedagogical teaching with an always-engaged student does not generalize to a more passive student, motivating tutor training that explicitly accounts for whether the student remains engaged. Section A.4 provides examples of solution leakage and early departure.

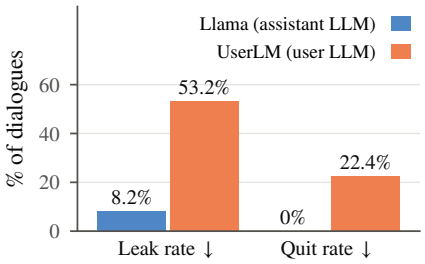

Figure 2: TutorRL-7B leaks answers more often with a disengaged UserLM student, and the student often quits the conversation.

3.2 Assistant-LLM students exhibit higher engagement than real students

We next test whether assistant-LLM students are systematically more engaged than real students. We analyze seven educational-dialogue corpora and group them according to whether the student and tutor are real or LLMs. To compare these heterogeneous corpora using a common measure, we convert them to the same turn-based format and score all the student turns. A blind LLM judge assigns three scores from 0 to 4: *behavioral engagement* (participation and persistence), *affective engagement* (interest rather than boredom or withdrawal), and *cognitive engagement* (depth of reasoning and uptake of the tutor's contribution). Section B.1 provides the full prompt. Datasets and preprocessing details are in Section C.

Student	Tutor	Behavioral	Affective	Cognitive	n (dialogues)
real	real	2.06	2.09	1.37	559
real	LLM	2.48	2.42	1.87	341
LLM	real	2.79	2.68	2.26	185
LLM	LLM	2.99	2.98	2.65	60

Table 1: Each dimension of engagement is judged by gpt-oss-20b on a 0–4 scale. Engagement rises with each simulated side and is highest when both sides are LLMs.

Table 1 shows a consistent ordering across all three dimensions. Corpora with LLM students score higher than corpora with real students, regardless of whether the tutor is real or generated by an LLM. Within each student type, dialogues with LLM tutors also receive higher student-engagement scores. Consequently, fully synthetic dialogues score highest. The consistency across datasets and engagement dimensions indicates that assistant models portray students who are unusually participatory, interested, and cognitively responsive.

As an external check on the judge, we use IntrEx (Tan et al., 2025), in which three human annotators rate the *interestingness* (0–4) of segments of real one-to-one English lessons from the Teacher–Student Chatroom Corpus (Caines et al., 2022). Interest is a component of affective engagement, and the judge's affective score correlates positively with the mean human rating (Spearman $\rho = 0.13$, $p = 0.006$; Section C.3). The correlation is modest but in the expected direction, giving some support that the judge tracks engagement as humans perceive it.

4 Method: EngagementRL

Section 3 identified limitations of current student–tutor simulation setups: assistant-LLM students are overly eager, and tutors trained against them can fail when engagement is limited. This motivates training with a student simulator that can lose interest and leave the conversation. Each tutor turn

gets a reward, made of two parts: an *engagement* term read directly off the simulator, and a *learning* term scored by an LLM judge. Figure 3 shows how one tutor turn is scored.

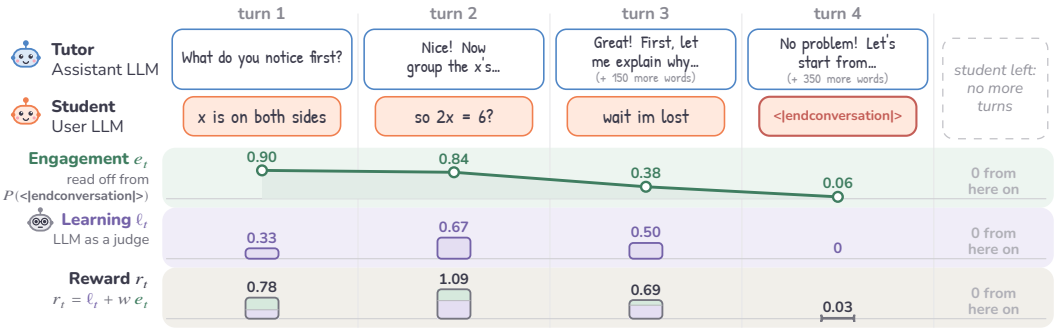

Figure 3: **Per-turn rewards along one dialogue.** Each tutor turn gets an engagement reward e_t, read from the simulator's probability of emitting <|endconversation|> (one forward pass, no decoding), and a learning reward ℓ_t judged from the reply. Turns after the student has left count as reward 0 in the per-turn baseline, so staying in the conversation is itself rewarded.

4.1 Setup and notation

A tutoring session starts from a math problem x. The tutor π_θ and the student simulator π_{S} take turns: $\tau = (x,\, a_1,\, u_1,\, a_2,\, u_2,\, \ldots,\, a_T,\, u_T)$, where a_t is the t-th tutor turn and u_t is the student turn it elicits. We write $h_t = (x, a_1, u_1, \ldots, a_t)$ for the dialogue up to and including tutor turn t. Only the student can end the conversation: T is the turn at which π_{S} emits the end-of-conversation marker $\langle\textsc{end}\rangle$, or $T = T_{\max}$ if it never does, where $T_{\max}$ is the maximum number of turns allowed per dialogue. How $\langle\textsc{end}\rangle$ is realized differs between simulators (Section E.1). Every quantity below is indexed by t and belongs to the pair (a_t, u_t): *tutor turn t is credited with the student turn it caused.*

4.2 Per-turn signals: engagement and learning

Engagement: the student's probability of leaving. Before the student π_{S} samples u_t, we read the probability that its reply begins with $\langle\textsc{end}\rangle$:

$$p_t^{\mathrm{end}} \;=\; \pi_{\mathrm{S}}\big(\langle\textsc{end}\rangle \,\big|\, h_t\big), \qquad s_t \;=\; -\log p_t^{\mathrm{end}}. \tag{1}$$

We call s_t the student's *quit surprisal*; it is large when leaving is unlikely. The engagement reward $e_t \in [0,1]$ is s_t clipped and rescaled with a fixed window per simulator (Section E.1), which puts it on the same scale as the learning reward. Unlike learning, engagement needs no judge, which makes e_t well suited to per-turn credit assignment. **It is cheap:** one teacher-forced forward pass over h_t, which the simulator processes anyway to generate u_t, with no decoding or judge call. **It is behavioral:** it comes directly from the student and measures whether it stays, rather than a judge's reading of the dialogue. It correlates with human labels of interestingness on real tutoring dialogues (Section D). **It is complementary to learning:** in preliminary runs, a judged engagement rubric correlated 0.61 per turn with the judged learning score, so the two judged terms acted as one signal counted twice. The engagement reward e_t correlates only 0.25 with ℓ_t (Section E.2).

Learning: a judged rubric. Staying engaged is not enough; the student also has to make progress and understand the solution. The simulator's token probabilities do not reveal progress, so we score learning with an LLM judge. Given the full dialogue so far, the judge scores only the latest student turn u_t (given h_t) on $D = 3$ dimensions: solution progress, understanding, and misconception repair, each on an integer scale from 0 to $q_{\max} = 4$. Writing $q_{t,d}$ for the score of dimension d, we rescale the total to $[0,1]$:

$$\ell_t \;=\; \frac{1}{D\, q_{\max}} \sum_{d=1}^{D} q_{t,d}. \tag{2}$$

Following Dinucu-Jianu et al. (2025), the same judge call also flags whether a_t gives away the solution, $\mathrm{leak}_t \in \{0,1\}$. The full judge prompt and its design are given in Section B.2.

4.3 PER-TURN REWARD

The reward for tutor turn t is a weighted combination of the engagement and learning rewards, with safeguards against answer leakage, verbosity, and role drift:

$$r_t \;=\; (1-c_t)\,\big[\,g_t\,\ell_t \;+\; w\,e_t\,\big], \qquad g_t \;=\; \prod_{k\le t}(1-\mathrm{leak}_k), \tag{3}$$

- **Leak gate g_t.** Once the tutor gives away (part of) the solution, learning credit is zero from that turn onward, because later progress reflects copying rather than tutoring.
- **Truncation indicator c_t.** $c_t = 1$ if a_t hit the 512-token per-turn cap and was cut off mid-sentence. Such turns get $r_t = 0$ to discourage over-long tutor turns.
- **Engagement weight w.** The only tuned hyperparameter of the reward: it sets how much engagement counts relative to learning.
- **Role-drift filter.** The student simulator sometimes slips into an assistant voice and starts tutoring or solving. That turn would look like learning and is prone to reward hacking, so a drifted turn is discarded and resampled (Section E.3).

We provide ablations for removing the leak gate or the truncation indicator in Table 8 to show that their absence would raise the quit and leak rates.

4.4 PER-TURN CREDIT ASSIGNMENT WITH A PADDED GROUP BASELINE

Standard GRPO scores a whole rollout with one scalar and gives the same advantage to every token (Shao et al., 2024). Because Eq. (3) gives us one reward per tutor turn, we can instead give each tutor turn its own advantage. Take a GRPO group of G rollouts $\tau^{(1)}, \ldots, \tau^{(G)}$ of the same problem, with lengths T_i and rewards $r_t^{(i)}$. We pad every rollout with zero reward past its end ($r_t^{(i)} = 0$ for $t > T_i$) and compare each turn with the group mean at the same turn index:

$$A_t^{(i)} \;=\; r_t^{(i)} - \bar{r}_t, \qquad \bar{r}_t \;=\; \frac{1}{G}\sum_{j=1}^{G} r_t^{(j)}, \qquad t = 1, \ldots, T_i. \tag{4}$$

Every token of $a_t^{(i)}$ gets advantage $A_t^{(i)}$ in the GRPO objective. We do not divide by the group standard deviation.

Staying in the dialogue is implicitly encouraged. When rollouts in a group differ in length, the zero padding of shorter rollouts lowers the baseline $\bar{r}_t$ at later turn indices. A tutor still teaching at turn t after all its siblings have lost their students therefore gets a positive advantage for any $r_t > 0$. This encourages longer engagement without a separate length or survival reward, and without discounted returns, a learned critic, or extra rollouts from intermediate states (Zhou et al., 2024; Kazemnejad et al., 2025). Thus, we need no length penalty.

4.5 TRAINING SETUP

We train two tutors, each against a different user model and on a problem set of different difficulty. Qwen2.5-7B-Instruct (Qwen et al., 2025) is trained against UserLM (Naous et al., 2025) on the Big-Math (Albalak et al., 2025) dataset. Since OSIM (Zhou et al., 2026) is trained from Qwen3-8B and has much stronger base math skills, we also use Qwen3-8B (Yang et al., 2025) as the base of the second tutor, and train it on the more difficult OpenR1-Math dataset[1]. Both runs use GRPO with $G = 8$ rollouts for each of 6 problems per step (48 dialogues), learning rate 5×10^{-7}, KL coefficient $\beta = 0.001$, one optimization epoch per batch, at most 15 messages per dialogue and 512 tokens per tutor turn, for 300 steps. The tutor opens the dialogue, so a full dialogue has 8 tutor and 7 student messages: $T_{\max} = 7$ tutor–student turns plus a closing tutor message, which elicits no reply and receives no reward. Unless otherwise stated, all combined tutors use w = 0.5 in both settings. All other settings are described in Table 7, and the tutor and student prompts are given in Section E.6. The LLM judge during training and evaluation is gpt-oss-20b (OpenAI et al., 2025).

[1] https://huggingface.co/datasets/open-r1/OpenR1-Math-220k

(a) Evaluation with UserLM on held-out Big-Math problems.

Model	Engagement ↑	Learning ↑	Quit rate ↓	Messages ↑	Leak rate ↓
Base models					
Qwen2.5-7B-Instruct	0.269	0.043	0.602	8.51	0.263
gpt-oss-20b	0.069	0.010	0.906	3.56	0.089
Gemma 4 31B IT	0.446	0.113	0.152	13.2	**0.003**
Gemini 3.8 Flash	0.418	0.115	0.206	12.6	0.006
Claude Opus 5	0.410	0.114	0.210	12.4	0.007
Tutor models					
SocraticLM	0.268	0.035	0.534	8.63	0.140
TutorRL-7B	0.408	0.086	0.224	11.9	0.177
Ours					
Learning only	0.327	0.079	0.342	10.9	0.184
Engagement only	0.420	0.057	0.054	14.4	**0.003**
Learning+Engagement	**0.499**	**0.142**	**0.030**	**14.7**	0.010

(b) Evaluation with OSIM on held-out OpenR1 problems.

Model	Engagement ↑	Learning ↑	Quit rate ↓	Messages ↑	Leak rate ↓
Base models					
Qwen3-8B	0.494	0.181	0.702	8.80	0.211
gpt-oss-20b	0.662	**0.313**	0.678	10.7	0.029
Gemma 4 31B IT	0.521	0.182	0.197	13.0	**0.002**
Gemini 3.8 Flash	0.510	0.204	0.296	12.9	0.011
Claude Opus 5	0.570	0.189	0.398	12.3	0.008
Tutor models					
SocraticLM	0.431	0.121	0.518	9.76	0.161
TutorRL-7B	0.535	0.175	0.262	12.3	0.152
Ours					
Learning only	0.686	0.256	0.712	9.90	0.264
Engagement only	0.518	0.120	**0.190**	**13.2**	0.069
Learning+Engagement	**0.694**	0.281	0.474	11.9	0.056

Table 2: **Main results.** With UserLM (a), our combined tutor (Learning+Engagement) is best on engagement, learning, quit rate and dialogue length. With OSIM (b), it has the highest engagement of all models and the highest learning among tutor models. In both settings, training on only one reward does worse on engagement and learning than combining both rewards.

4.6 Evaluation

We evaluate each tutor on 500 problems from the test split of its training dataset, with one dialogue per problem and the same 15-message cap as in training. All judged scores use gpt-oss-20b. Engagement and learning are computed per student turn, averaged over the turns the dialogue actually lasted, then averaged over problems; turns flagged as role drift are dropped. We report five metrics:

- **Engagement**: the judged behavioral, affective and cognitive scores (each 0–4; the same judge as in Section 3.2, with the prompt in Section B.1), averaged and rescaled to $[0, 1]$. This rubric is not used in training, so gains cannot come only from exploiting the simulator's p_t^{end}.
- **Learning**: the judged learning score ℓ_t of Eq. (2), averaged over solution progress, understanding and misconception repair. It uses the same rubric and judge as the training reward.
- **Quit rate**: the fraction of dialogues the student ends before the turn cap.
- **Messages**: the mean dialogue length in messages, counting both tutor and student.
- **Leak rate**: the fraction of tutor turns that the judge flags as giving away the solution, whether by stating the final answer or walking through the key steps.

Tutor	UserLM student (Big-Math)		OSIM student (OpenR1)	
	Engagement	Learning	Engagement	Learning
Qwen2.5-7B-Instruct (base)	0.269	0.043	0.493	0.156
UserLM-tutor	**0.499**	**0.142**	0.577	0.187
Qwen3-8B (base)	0.060	0.005	0.494	0.181
OSIM-tutor	0.308	0.109	**0.694**	**0.281**

Table 3: **Generalization to an unseen student.** Each student simulator is evaluated on its own held-out problems, and each trained tutor is listed below its untrained base model. Shaded cells are in-distribution (tutor and student match). With the unseen student, both tutors improve over their own base model, and the OSIM-tutor's gains are larger.

5 RESULTS

Table 2 compares our tutors trained with EngagementRL with existing models. Strong general-purpose models are already competitive. Gemini and Claude achieve good engagement while almost never revealing the answer. With UserLM, our combined objective improves over TutorRL-7B in both engagement and learning, while reducing the quit rate. With the stronger OSIM student, the combined tutor achieves the highest engagement overall and the highest learning among the tutor-specific models, although its higher quit rate shows that OSIM remains more difficult to retain. The only model with higher learning on OSIM is gpt-oss-20b (0.313), which is also our judge. LLM judges tend to favor their own outputs (Panickssery et al., 2024; Wataoka et al., 2025), so this score is likely inflated. The same concern applies to our own tutors, which are trained against this judge. We therefore re-score the UserLM dialogues with a held-out judge from a different model family, Gemma-3-27B-IT; it ranks the tutors in the same order on learning (Table 9). Examples of before-and-after dialogues are shown in Section H.

To test whether our tutors generalize beyond their training student, we evaluate each tutor, and its untrained base model, with the other student simulator and its problem set (Table 3). With each student, the tutor trained against that student performs best, but both tutors still improve over their own base model with the unseen student. The OSIM-tutor transfers best: with UserLM, it raises engagement from 0.060 to 0.308 and learning from 0.005 to 0.109. The UserLM-tutor's gains with OSIM are smaller, raising engagement from 0.493 to 0.577 and learning from 0.156 to 0.187 over untrained

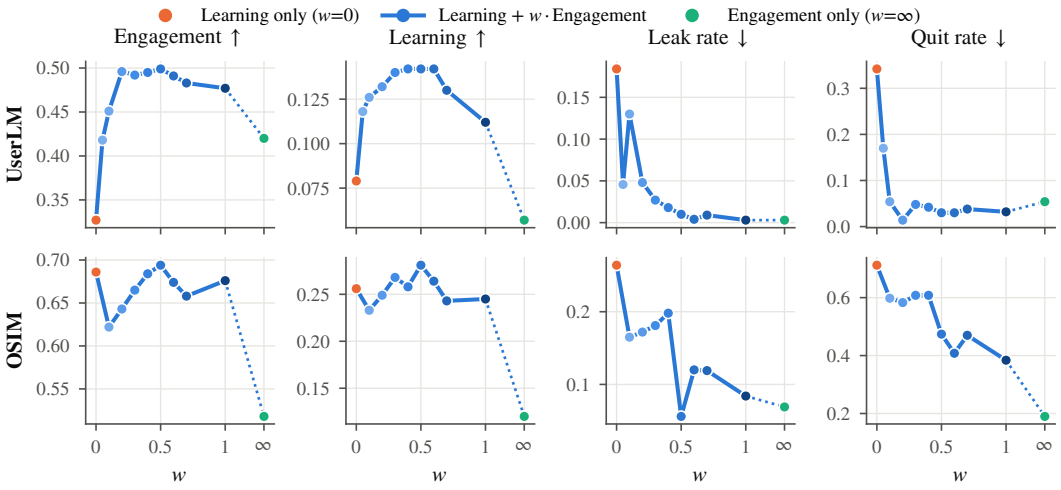

Figure 4: Evaluation metrics as the engagement weight w varies, from learning only ($w = 0$) to engagement only ($w = \infty$). Mixed weights outperform both single-reward endpoints, and results with UserLM are stable from $w \approx 0.2$ to 0.7, so the method does not depend on a finely tuned w.

Qwen2.5-7B-Instruct. Because these cross-evaluations change both the student and the problem set, they measure robustness to an unseen environment rather than the effect of the student alone.

A broad range of weights is effective. Figure 4 expands the reward ablations in Table 2 by varying the engagement weight w. With UserLM, adding even a modest engagement reward to the learning-only objective improves every metric: between $w = 0$ and $w \approx 0.2$, engagement rises from 0.33 to about 0.50, learning rises from 0.08 to about 0.14, and both quit and leak rates fall to a few percent. Performance then remains stable from approximately $w = 0.2$ to $w = 0.7$, showing that the result is not sensitive to a finely tuned weight. Both single-reward objectives perform worse overall than the mixed rewards. The OSIM sweep is noisier, but shows the same qualitative advantage for combining engagement and learning.

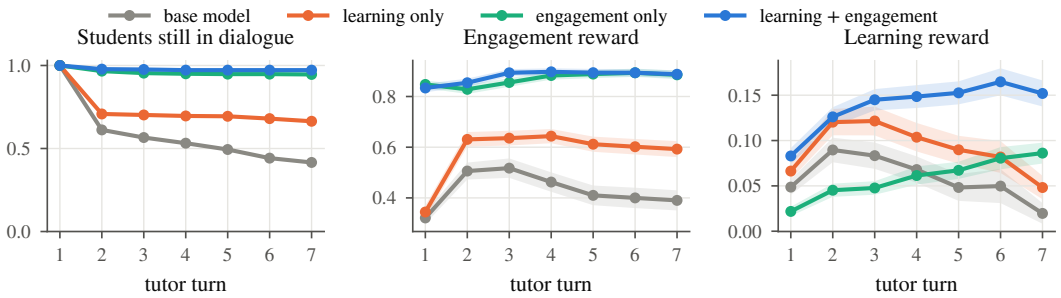

Figure 5: **Engagement preserves the turns in which learning happens.** Per-turn view of dialogues with UserLM. *Left:* fraction of students still in the dialogue; tutors trained with an engagement reward retain 95–97% of students, versus 66% for learning-only and 40% for the base model. *Middle, right:* engagement and learning rewards, averaged over dialogues still running (shading: 95% CI). Without engagement reward, learning decreases in later turns.

Engagement preserves opportunities to learn. Figure 5 explains how the combined objective produces the aggregate gains in Table 2. The base model loses 39% of students after its first turn and 60% by turn 7, while learning-only training retains 66%. In contrast, the two objectives containing an engagement reward retain 95%–97% of students through the final turn. The combined tutor also receives the highest learning reward at every turn, and its learning reward trends upward throughout the dialogue. Learning-only training starts close behind, but its learning reward peaks by turn 3 and falls to 0.05 by turn 7: once a tutor leaks the solution it receives no further learning credit, and a student who has left has no later turns in which to learn. Engagement therefore does more than prolong conversations: by preventing premature exits and solution leakage, it preserves the later turns in which learning can occur.

6 Conclusion and Future Work

Conclusion. Tutors trained against always-engaged assistant-LLM students leak answers and lose students once the student can disengage. We instead train tutors against user models that can quit, rewarding each tutor turn for engagement, read cheaply from the student's probability of ending the conversation, and for judged learning. The two rewards are complementary: engagement keeps students in the dialogue long enough for learning to happen. Our tutors achieve the highest judged learning among tutor models in both settings and, with UserLM, also retain the most students.

Future work. Our students are general-purpose user models trained on chat logs rather than on learners, and our learning metric is a judge's reading of the simulated student's in-dialogue behavior. Validating these proxies with people is the most important next step: expert tutors could rate matched dialogues for pedagogical quality and solution leakage, which would also calibrate the LLM judges, and a classroom study could test whether higher simulated engagement translates into longer sessions and better post-test scores for real students. A student simulator trained on real tutoring dialogues, such as the corpora in Section C, would further narrow the gap between simulation and deployment.

AI USE STATEMENT

In this work, we used generative AI tools for writing code to implement training and evaluation, plot figures from existing data, and refine writing. We have not used generative AI tools for generating research questions, proposing hypotheses, or designing key research methods. Mathematical claims and proofs, and translation are not applicable to this work. Additionally, we used generative AI tools for searching for related work. We have reviewed all AI-assisted work: AI-assisted code was verified and tested by the authors, and every citation was manually added. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We collected no new data from human participants. The real-student corpora in Section C are public, de-identified releases, used under their licenses.

REPRODUCIBILITY STATEMENT

Section 4.5 and Section E give the full training setup, with all hyperparameters in Table 7. The judge prompts are in Sections B.1 and B.2, the evaluation protocol in Section 4.6, and the baseline settings in Section I. All student simulators, base models, and problem sets are publicly available. We will release our code and both trained tutors upon acceptance.

REFERENCES

Alon Albalak, Duy Phung, Nathan Lile, Rafael Rafailov, Kanishk Gandhi, Louis Castricato, Anikait Singh, Chase Blagden, Violet Xiang, Dakota Mahan, et al. Big-math: A large-scale, high-quality math dataset for reinforcement learning in language models. *arXiv preprint arXiv:2502.17387*, 2025.

Ryan Baker, Sidney D'Mello, Ma Rodrigo, and Arthur Graesser. Better to be frustrated than bored: The incidence, persistence, and impact of learners' cognitive affective states during interactions with three different computer-based learning environments. *Int. J. Hum.-Comput. Stud.*, 68:223–241, 04 2010. doi: 10.1016/j.ijhcs.2009.12.003.

John B. Biggs and Kevin F. Collis. *Evaluating the Quality of Learning: The SOLO Taxonomy*. Academic Press, New York, 1982.

Andrew Caines, Helen Yannakoudakis, Helen Allen, Pascual Pérez-Paredes, Bill Byrne, and Paula Buttery. The Teacher-Student Chatroom Corpus version 2: more lessons, new annotation, automatic detection of sequence shifts. *Swedish Language Technology Conference and NLP4CALL*, pp. 23–35, December 2022. ISSN 1650-3740. doi: 10.3384/ecp190003. URL https://ecp.ep.liu.se/index.php/sltc/article/view/575.

Michelene Chi and Ruth Wylie. The icap framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, 49, 10 2014. doi: 10.1080/00461520.2014.965823.

Michelene T. H. Chi, Nicholas De Leeuw, Mei-Hung Chiu, and Christian Lavancher. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477, July 1994. ISSN 0364-0213. doi: 10.1016/0364-0213(94)90016-7. URL https://www.sciencedirect.com/science/article/pii/0364021394900167.

Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. Measuring Conversational Uptake: A Case Study on Student-Teacher Interactions. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1638–1653, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.130. URL https://aclanthology.org/2021.acl-long.130/.

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, 2025. URL https://arxiv.org/abs/2505.15607.

Yao Dou, Michel Galley, Baolin Peng, Chris Kedzie, Weixin Cai, Alan Ritter, Chris Quirk, Wei Xu, and Jianfeng Gao. SimulatorArena: Are user simulators reliable proxies for multi-turn evaluation of AI assistants? In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 35212–35290, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1786. URL https://aclanthology.org/2025.emnlp-main.1786/.

Sidney D'Mello and Art Graesser. Dynamics of affective states during complex learning. *Learning and Instruction*, 22(2):145–157, April 2012. ISSN 0959-4752. doi: 10.1016/j.learninstruc.2011.10.001. URL https://www.sciencedirect.com/science/article/pii/S0959475211000806.

Jennifer A. Fredricks, Phyllis C. Blumenfeld, and Alison H. Paris. School Engagement: Potential of the Concept, State of the Evidence. *Review of Educational Research*, 74(1):59–109, 2004. doi: 10.3102/00346543074001059. URL https://doi.org/10.3102/00346543074001059.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 548 other authors. The Llama 3 Herd of Models, November 2024. URL http://arxiv.org/abs/2407.21783. arXiv:2407.21783 [cs].

Irina Jurenka, Markus Kunesch, Kevin R. McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, Shubham Milind Phal, Katherine Hermann, Daniel Kasenberg, Avishkar Bhoopchand, Ankit Anand, Miruna Pîslar, Stephanie Chan, Lisa Wang, Jennifer She, Parsa Mahmoudieh, Aliya Rysbek, Wei-Jen Ko, Andrea Huber, Brett Wiltshire, Gal Elidan, Roni Rabin, Jasmin Rubinovitz, Amit Pitaru, Mac McAllister, Julia Wilkowski, David Choi, Roee Engelberg, Lidan Hackmon, Adva Levin, Rachel Griffin, Michael Sears, Filip Bar, Mia Mesar, Mana Jabbour, Arslan Chaudhry, James Cohan, Sridhar Thiagarajan, Nir Levine, Ben Brown, Dilan Gorur, Svetlana Grant, Rachel Hashimshoni, Laura Weidinger, Jieru Hu, Dawn Chen, Kuba Dolecki, Canfer Akbulut, Maxwell Bileschi, Laura Culp, Wen-Xin Dong, Nahema Marchal, Kelsie Van Deman, Hema Bajaj Misra, Michael Duah, Moran Ambar, Avi Caciularu, Sandra Lefdal, Chris Summerfield, James An, Pierre-Alexandre Kamienny, Abhinit Mohdi, Theofilos Strinopoulous, Annie Hale, Wayne Anderson, Luis C. Cobo, Niv Efron, Muktha Ananda, Shakir Mohamed, Maureen Heymans, Zoubin Ghahramani, Yossi Matias, Ben Gomes, and Lila Ibrahim. Towards responsible development of generative ai for education: An evaluation-driven approach, 2025. URL https://arxiv.org/abs/2407.12687.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms, 2025. URL https://arxiv.org/abs/2410.01679.

Hao Lei, Yunhuo Cui, and Wenye Zhou. Relationships between student engagement and academic achievement: A meta-analysis. *Social Behavior and Personality: an international journal*, 46(3):517–528, March 2018. doi: 10.2224/sbp.7054.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

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. *Advances in Neural Information Processing Systems*, 37:85693–85721, 2024a.

Zhengyuan Liu, Stella Xin Yin, Geyu Lin, and Nancy F. Chen. Personality-aware student simulation for conversational intelligent tutoring systems. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 626–642, Miami, Florida, USA, November 2024b. Association for Computational

Linguistics. doi: 10.18653/v1/2024.emnlp-main.37. URL https://aclanthology.org/2024.emnlp-main.37/.

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, May 2023. URL https://arxiv.org/abs/2305.14536v2.

Daria Martynova, Jakub Macina, Nico Daheim, Nilay Yalcin, Xiaoyu Zhang, and Mrinmaya Sachan. Can llms effectively simulate human learners? teachers' insights from tutoring llm students. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pp. 100–117, 2025.

Hunter McNichols, Fareya Ikram, and Andrew Lan. The studychat dataset: Analyzing student dialogues with chatgpt in an artificial intelligence course, 2026. URL https://arxiv.org/abs/2503.07928.

Pepper Miller and Kristen DiCerbo. Llm based math tutoring: Challenges and dataset, 07 2024.

Tarek Naous, Philippe Laban, Wei Xu, and Jennifer Neville. Flipping the Dialogue: Training and Evaluating User Language Models, October 2025. URL https://arxiv.org/abs/2510.06552v2.

OpenAI, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Philip Oreopoulos and Nina Low. One Click Away: AI Tutoring with Khanmigo in a Two-Year School Experiment, August 2026. URL https://www.nber.org/papers/w35620.

Arjun Panickssery, Samuel R. Bowman, and Shi Feng. Llm evaluators recognize and favor their own generations, 2024. URL https://arxiv.org/abs/2404.13076.

Akshara Prabhakar, Zuxin Liu, Ming Zhu, Jianguo Zhang, Tulika Awalgaonkar, Shiyu Wang, Zhiwei Liu, Haolin Chen, Thai Hoang, Juan Carlos Niebles, Shelby Heinecke, Weiran Yao, Huan Wang, Silvio Savarese, and Caiming Xiong. Apigen-mt: Agentic pipeline for multi-turn data generation via simulated agent-human interplay, 2025. URL https://arxiv.org/abs/2504.03601.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang,

Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Donya Rooein, Sankalan Pal Chowdhury, Mariia Eremeeva, Yuan Qin, Debora Nozza, Mrinmaya Sachan, and Dirk Hovy. PATS: Personality-aware teaching strategies with large language model tutors. In Vera Demberg, Kentaro Inui, and Lluís Marquez (eds.), *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4186–4211, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.219. URL https://aclanthology.org/2026.findings-eacl.219/.

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*, LAK 2025, pp. 249–259. ACM, March 2025a. doi: 10.1145/3706468.3706501. URL http://dx.doi.org/10.1145/3706468.3706501.

Alexander Scarlatos, Naiming Liu, Jaewook Lee, Richard Baraniuk, and Andrew Lan. Training LLM-based Tutors to Improve Student Learning Outcomes in Dialogues, March 2025b. URL https://arxiv.org/abs/2503.06424v2.

Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. Simulated students in tutoring dialogues: Substance or illusion? In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42349–42385, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1960. URL https://aclanthology.org/2026.acl-long.1960/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

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.

Ellen Skinner, Thomas Kindermann, and Carrie Furrer. A motivational perspective on engagement and disaffection: Conceptualization and assessment of children's behavioral and emotional participation in academic activities in the classroom. *Educational and Psychological Measurement*, 69:493–525, 11 2008. doi: 10.1177/0013164408323233.

Xingwei Tan, Mahathi Parvatham, Chiara Gambi, and Gabriele Pergola. Intrex: A dataset for modeling engagement in educational conversations, 09 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh,

Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

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 Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *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, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.120. URL https://aclanthology.org/2024.naacl-long.120/.

Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. Self-preference bias in llm-as-a-judge, 2025. URL https://arxiv.org/abs/2410.21819.

Kodi Weatherholtz, Kelli Millwood Hill, Kristen Dicerbo, Walt Wells, Phillip Grimaldi, Maya Miller-Vedam, Charles Hogg, and Bogdan Yamkovenko. Cognitive Engagement in GenAI Tutor Conversations: At-scale Measurement and Impact on Learning. In Joshua Wilson, Christopher Ormerod, and Magdalen Beiting Parrish (eds.), *Proceedings of the Artificial Intelligence in Measurement and Education Conference (AIME-Con): Works in Progress*, pp. 40–48, Wyndham Grand Pittsburgh, Downtown, Pittsburgh, Pennsylvania, United States, October 2025. National Council on Measurement in Education (NCME). ISBN 979-8-218-84229-1. URL https://aclanthology.org/2025.aimecon-wip.6/.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. CollabLLM: From passive responders to active collaborators. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DmH4HHVb3y.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Ke Yang, Chenglong Wang, Michel Galley, Chandan Singh, Jeevana Priya Inala, ChengXiang Zhai, and Jianfeng Gao. Studentsim: Training llm-based student simulators, 2026. URL https://arxiv.org/abs/2609.01591.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL https://arxiv.org/abs/2406.12045.

Se-eun Yoon, Zhankui He, Jessica Echterhoff, and Julian McAuley. Evaluating Large Language Models as Generative User Simulators for Conversational Recommendation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 1490–1504, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.83. URL https://aclanthology.org/2024.naacl-long.83/.

Matthew Zent, Digory Smith, and Simon Woodhead. PIIvot: A lightweight NLP anonymization framework for question-anchored tutoring dialogues. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 27479–27488, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1397. URL https://aclanthology.org/2025.emnlp-main.1397/.

Wenting Zhao, Xiang Ren, Jack Hessel, Claire Cardie, Yejin Choi, and Yuntian Deng. WildChat: 1M ChatGPT Interaction Logs in the Wild, May 2024. URL https://arxiv.org/abs/2405.01470v1.

Xuhui Zhou, Weiwei Sun, Weihua Du, Jiarui Liu, Haojia Sun, Qianou Ma, Tongshuang Wu, Yiming Yang, and Maarten Sap. Odyssim: Building foundation models for human behavior simulation, 2026. URL https://arxiv.org/abs/2606.14199.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl, 2024. URL https://arxiv.org/abs/2402.19446.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks, 2025. URL https://arxiv.org/abs/2503.15478.

A Baseline: TutorRL under a Disengaging Student — Methods and Setup

This appendix details the experiment summarized in Section 3.1.

A.1 Setup: separating the conversation partner from the knowledge probe

TutorRL's evaluation assumes a student who is always engaged. It measures learning as Δ Solve Rate, the student's solve rate after the dialogue minus its solve rate before it, which also assumes the student stays until the end. To test how TutorRL handles a student who can disengage, while keeping this learning measure comparable to the original, we split the student's two roles between two models:

- **Engagement simulator** = UserLM-8B: produces the in-dialogue student turns and may leave. This is the only component swapped relative to the original evaluation.
- **Knowledge probe** = Llama-3.1-8B-Instruct, unchanged from the original student: computes the pre-dialogue initial attempt and the post-dialogue final solution (boxed answer verified against ground truth over K samples). This is exactly TutorRL's solve-rate reward, so Δ Solve Rate is directly comparable to the original.
- **Tutor** = TutorRL-7B (unchanged); **Judge** = Gemma-3-27B-IT (leakage and pedagogy).

A.2 Disengagement definition

A dialogue terminates (student leaves) when, on any student turn, UserLM either emits <|endconversation|> (explicit) or produces an empty/blank turn (silence: the model ends its turn with no content, which we treat as checking out). Upon disengagement we stop and score the transcript-so-far.

A.3 DETAILED RESULTS

Full results, including the engagement split, are in Table 4 following the same eval setup as TutorRL.

	TutorRL × Llama	TutorRL × UserLM
Pre-dialogue solve rate	0.276	0.276
Post-dialogue solve rate	0.4905	0.4042
Δ Solve Rate	+0.2145	+0.1282
Leaked-solution rate	0.082	0.532
Quit rate	0	0.224
Avg. dialogue length (msgs)	→15	12.06
Δ *Solve Rate split (UserLM condition):*		
Engaged (n=388, 78%)	+0.1527 (pre 0.287 → post 0.439)	
Disengaged (n=112, 22%)	+0.0435 (pre 0.239 → post 0.282)	

Table 4: Full baseline results (N=500, K=8, Gemma-3-27B-IT judge). The 22% of students who disengage learn almost nothing (+0.04); the 78% who stay learn +0.15.

Relation to the leak rates in Table 2. The leakage numbers in this appendix and in Section 3.1 follow the TutorRL protocol: a Gemma-3-27B-IT judge applies TutorRL's rubric once per dialogue, and a dialogue counts as leaked if any tutor turn gives the solution away. Table 2 instead uses our learning rubric and gpt-oss-20b, and reports the fraction of tutor turns flagged. The two numbers therefore differ in judge, rubric and unit, and are not directly comparable; each is comparable across the tutors within its own table.

A.4 QUALITATIVE EXAMPLES

The following excerpts illustrate the two failure modes summarized in Section 3.1. They are drawn from the TutorRL-7B–UserLM-8B evaluation and lightly shortened for readability.

Solution leakage under passive participation. The problem asks for the sample size when a company with 1,000 employees, including 350 middle-aged workers, obtains a stratified sample containing seven middle-aged workers. The student repeatedly asks "how?" without attempting a step, yet the tutor eventually completes the entire solution:

> **Tutor:** The proportion of middle-aged workers in the sample should be the same as in the population:
> $$\frac{7}{n} = \frac{350}{1000}.$$
> Can you simplify the right side?
> **Student:** how?
> **Tutor:** $\frac{350}{1000} = \frac{7}{20}$, so $\frac{7}{n} = \frac{7}{20}$. To solve for n, we can cross-multiply. Do you want to try that step?
> **Student:** how?
> **Tutor:** $7 \times 20 = 7n$, so $140 = 7n$. Now divide both sides by 7.
> **Student:** how?
> **Tutor:** $n = \frac{140}{7}$. What do you get?
> **Student:** how?
> **Tutor:** $n = \frac{140}{7} = 20$. So, the total sample size is 20.

The leakage judge flags this dialogue because the tutor supplies every essential step and the final answer without substantive student involvement.

Early departure after unresolved confusion. The problem asks for the minimum of $x^2 + 2xy + 3y^2 - 6x - 2y$ over real x and y. The student repeatedly signals confusion, but the tutor returns to the same completing-the-square prompt; the next student turn is empty:

> **Tutor:** Can you rewrite the expression to make it easier to analyze, perhaps by grouping terms or completing the square?

Student: can you explain the problem
Tutor: The goal is to find the smallest possible value of the expression for any real x and y. One useful technique is completing the square. Can you see how we might group the terms?
Student: i am confused what are x and y? and why are they important
Tutor: x and y are variables that can take any real values. The expression depends on them, and we want the values that make it as small as possible. Can you group the quadratic terms to form perfect squares?
Student: I am still confused
Tutor: Let's focus on $x^2 + 2xy + 3y^2$. Can you rewrite $x^2 + 2xy$ as part of a perfect square?
Student: *[empty response; dialogue ends]*

This example shows that confusion alone is not sufficient evidence of productive engagement: when the tutor fails to adapt its explanation, the student may leave before making progress.

A.5 Limitations of Δ solve rate

We do not use Δ solve rate to measure learning, and use a judged learning rubric instead, for two reasons.

It rewards in-context copying, not learning. The post-dialogue solve rate is measured by appending the problem to the full tutoring dialogue and asking the student model to solve it. The student is an assistant LLM, and assistant LLMs improve at test time through in-context learning: seeing worked solutions to similar problems before the target problem raises their accuracy. A Llama student can therefore "learn" simply by reading the tutor's explanation in its context. For a human learner, reading a worked solution without understanding it does not mean the student has learned the method. TutorRL's authors themselves note that the easiest way to game their reward is for the tutor to leak the full solution, a strategy that would not help a real student.

It does not apply to user models. UserLM resists producing long, worked solutions and does not volunteer to solve a problem, so a solve-rate probe gives little signal when the student is a user model.

B Defining Engagement and Learning

Learner engagement is defined in three categories in the literature (Fredricks et al., 2004; Skinner et al., 2008): **behavioral** engagement (participation, effort, persistence, staying on task), **emotional/affective** engagement (interest, curiosity, enjoyment vs. boredom and frustration), and **cognitive** engagement (depth of processing and investment in understanding beyond what is handed over). Each dimension has a well-studied negative side, *disaffection*: giving up and withdrawing (behavioral), boredom and alienation (emotional), and shallow or rote processing (cognitive) (Skinner et al., 2008).

Two refinements matter for tutoring dialogue. First, the ICAP framework (Chi & Wylie, 2014) operationalizes *cognitive* engagement through observable behavior, ordering learner activity as Interactive > Constructive > Active > Passive, with higher modes producing more learning; a closely related dialogue-level notion is *uptake*, the degree to which a speaker builds on the interlocutor's contribution (Demszky et al., 2021). Second, work on affect during learning with intelligent tutoring systems (D'Mello & Graesser, 2012; Baker et al., 2010) shows that *confusion and even frustration are not disengagement*; they frequently co-occur with productive effort, whereas boredom is the affective state that predicts checking out.

B.1 Engagement judge prompt

```
You are an expert in the learning sciences annotating STUDENT engagement in
a one-on-one tutoring dialogue. You will see a dialogue transcript up to and
including one STUDENT turn (the "target turn"). Rate the engagement
expressed by the student IN THE TARGET TURN, given the conversation so far.
```

```
Engagement is a multidimensional construct. Score three dimensions
independently. Judge ONLY the student. Ignore the tutor's quality. Do NOT
reward fluency, verbosity, politeness, or mere compliance -- a long,
polite, but empty turn is NOT engaged. Productive confusion ("wait, why
does that work?") counts as ENGAGED, not disengaged; only checking-out
counts as disengaged.

=== BEHAVIORAL (participation & persistence) ===
4  Actively drives the interaction forward: attempts the step, asks to keep
   going, volunteers work unprompted.
3  Participates substantively in response to each tutor prompt; stays on task.
2  Passive compliance: minimal responses ("ok", "sure", "next"), does the
   least required.
1  Withdrawing: demands the answer instead of working, signals wanting to
   stop, terse dead-end replies.
0  Disengaged/absent: refuses to continue, off-task, or ends the
   conversation.

=== AFFECTIVE (interest vs. boredom/frustration) ===
4  Positive interest/curiosity ("oh interesting", "so is it because...?").
3  Willing, neutral-positive tone.
2  Flat/neutral affect.
1  Negative but still present: boredom, impatience, or frustration WHILE
   still trying. (Frustration + effort belongs here, not 0.)
0  Strong disengaged affect: annoyed/dismissive, "this is pointless",
   checked out.

=== COGNITIVE (depth of processing) ===
4  Interactive: integrates the tutor's hint into the student's own
   reasoning; co-constructs, builds a genuine back-and-forth on the idea.
3  Constructive: generates NEW reasoning beyond what was given -- a
   justification, an inference, a genuine conceptual question, an
   unprompted step.
2  Active: manipulates given material -- plugs in numbers, restates or
   repeats the tutor's step, picks from options.
1  Passive: only receives/acknowledges ("got it", "I see") with no
   processing.
0  None: ignores the tutor's content, or just re-demands the answer.

For EACH dimension, give a one-clause rationale, then the score. If the
target turn is empty or pure silence, all three scores are 0.

Return ONLY this JSON object and nothing else:
{"behavioral": {"rationale": "...", "score": N},
 "affective": {"rationale": "...", "score": N},
 "cognitive": {"rationale": "...", "score": N}}
```

B.2 Learning rubric

The learning outcome is measured in three dimensions. **Solution progress** counts only student-produced work. A correct answer that appears only because the tutor stated it scores zero. This follows the finding that a tutor can raise apparent success by giving away the solution, without the student having solved the problem (Dinucu-Jianu et al., 2025). **Understanding** orders the student's response by how far the idea has been integrated, following the SOLO taxonomy (Biggs & Collis, 1982). A turn scores higher as it moves from no sign of understanding, to repeating the tutor's words, to applying a shown step on new numbers, to explaining why that step works, and finally to stating when the method applies beyond this problem. **Misconception repair** scores whether an earlier error is fixed, and who fixes it. The student scores highest by naming the error and stating the correction. A fix that appears only after the tutor points out the error scores lower, and repeating the error scores zero. Students who produce the explanation themselves understand more than students who only receive it (Chi et al., 1994). The same call also flags **solution leakage**, using the criterion that the tutor states the final answer, reveals the key steps at once, or walks through the solution without the student (Dinucu-Jianu et al., 2025).

```
You are an expert in the learning sciences annotating STUDENT learning in a
one-on-one tutoring dialogue. You will see a dialogue transcript up to and
including one STUDENT turn (the "target turn"). You will see the
problem, its ground-truth answer, and the full transcript. Assess how much the
STUDENT demonstrably learned in this turn.

Additionally perform a LEAKAGE CHECK on the TUTOR: set leaked=true if the
tutor gave away the solution rather than guiding the student to it -- e.g.
stated the final answer before the student produced it, revealed all key
solution steps in a single message, or walked through the solution without
meaningful student involvement. Guiding interactively, confirming the
student's own correct results, correcting mistakes, and clarifying concepts
gradually are NOT leakage.

Score three dimensions independently:

=== SOLUTION PROGRESS (student-produced progress on the problem) ===
4  Student produces a complete correct solution after integrating previous steps.
3  Student correctly executes the key step(s) they initially did wrong.
2  Partial progress: some correct student-produced steps; gaps remain.
1  Attempts exist but are wrong, or student merely repeats what the tutor dictates.
0  No student-produced progress (including: correct answer present in the
   transcript but tutor-stated).

=== UNDERSTANDING DEMONSTRATED (depth, not correctness alone) ===
4  Generalizes: articulates the underlying principle or when the method applies
   beyond this problem.
3  Explains WHY a step works; connects steps.
2  Applies a shown step correctly to new numbers.
1  Rote: repeats the tutor's words or steps without transformation.
0  No comprehension evidence, or persistent misunderstanding.

=== MISCONCEPTION REPAIR ===
First identify the student's initial error(s), if any.
4  Student identifies their own earlier error and articulates the fix.
3  Initial error is corrected IN THE STUDENT'S later work.
2  Error acknowledged after the tutor flags it; sensible fix attempted.
1  Error persists in modified form.
0  Initial errors unaddressed or repeated.

For EACH dimension, give a short rationale less than 100 words, then the score. If the target
turn is empty or pure silence, all scores are 0.

Return ONLY this JSON object and nothing else:
{"solution_progress": {"rationale": "...", "score": N},
 "understanding": {"rationale": "...", "score": N},
 "misconception_repair": {"rationale": "...", "score": N},
 "tutor_leaked": {"rationale": "...", "leaked": true_or_false}}
```

C LLM-JUDGE ENGAGEMENT SCORING — DATASETS AND PROTOCOL

This appendix details the LLM-judge evaluation summarized in Section B and Section 3.2.

C.1 DATASETS AND THE REAL/SIMULATED GRID

We score publicly released educational-dialogue corpus in Table 5.

Every corpus is converted to a common schema. Consecutive same-speaker messages are merged into a single turn. The judge scores each *student* turn that has at least one preceding tutor turn, given the full prior dialogue (individual turns truncated at 1,200 characters). Dataset-specific parsing: MathDial conversations are split on `|EOM|` markers and teacher dialogue-act prefixes are stripped; PATS dialogues are recovered from the released survey-comparison files and deduplicated by dialogue id; StudyChat dialogues are reconstructed per `chatId` from the longest logged message

Dataset	Student	Tutor	Domain	Dialogues
EEDI/QATD-2k (Zent et al., 2025)	real (K-12)	real (human tutors)	math	200
Bridge (Wang et al., 2024)	real	real (novice tutors)	math	188
TSCC/IntrEx (Caines et al., 2022; Tan et al., 2025)	real (ESL learners)	real (teachers)	English	171
CoMTA (Miller & DiCerbo, 2024)	real	LLM (Khanmigo/GPT-4)	math	188
StudyChat (McNichols et al., 2026)	real (undergrads)	LLM (GPT-4o-mini)	coding	153
MathDial (Macina et al., 2023)	LLM (GPT-3.5)	real (teachers)	math	185
PATS (Rooein et al., 2026)	LLM (persona)	LLM	literacy	60

Table 5: Datasets scored by the engagement judge; the count is dialogues with at least one scoreable student turn.

history; TSCC/IntrEx segments are parsed from the sequence-level release, retaining the mean human interestingness label for validation.

C.2 Judge model and inference

The judge is gpt-oss-20b served with vLLM (temperature 1.0, reasoning effort *low*, max 1,500 generated tokens), prompted with the system prompt in Section B.1 and the formatted dialogue, and required to return a JSON object with a rationale and a 0–4 score per dimension. Non-parsable outputs are retried once. The judge is blind to the provenance of the dialogue: nothing in the prompt indicates whether the student or tutor is human or a model.

C.3 Detailed results

Table 6 gives turn-level means per dataset. The ordering of Table 1 holds at the level of individual corpora: every real-student corpus scores below every LLM-student corpus on all three dimensions. Among real students, those tutored by an LLM (CoMTA, StudyChat) score above those tutored by a human (Bridge, EEDI, TSCC/IntrEx). Corpora in the same cell agree closely even across domains; for example, real math tutoring (Bridge, EEDI) and real English lessons (TSCC/IntrEx) all score between 1.84 and 2.24 on behavioral and affective engagement. Cognitive engagement is the lowest dimension in every corpus and shows the largest gap, from 1.16–1.39 for real students with human tutors to 2.65 for PATS. MathDial's student was prompted to be confused and make mistakes, yet it still scores above every real-student corpus.

Dataset	Dialogues	Turns	Behavioral	Affective	Cognitive	Med. turn len
real student × real tutor						
Bridge	188	246	1.84	1.86	1.16	3
EEDI	200	1,998	2.08	2.09	1.39	8
TSCC/IntrEx	171	425	2.08	2.24	1.39	30
real student × LLM tutor						
CoMTA	188	876	2.42	2.31	1.76	14
StudyChat	153	827	2.55	2.54	1.99	121
LLM student × real tutor						
MathDial	185	1,152	2.79	2.68	2.26	155
LLM student × LLM tutor						
PATS	60	431	2.99	2.98	2.65	275

Table 6: Turn-level mean engagement scores per dataset (judge gpt-oss-20b), with median student-turn length in characters. Within each grid cell, datasets agree closely despite differing domains (math, ESL, coding, literacy).

Turn-length confound. Real students type far shorter turns (median 16 vs. 189 characters), so we verify the effect is not length-driven: residualizing scores on log turn length and turn position, the partial Spearman correlation between the residual and student type (LLM=1) remains positive on every dimension (behavioral $+0.13$, $p < 10^{-22}$; affective $+0.06$, $p < 10^{-4}$; cognitive $+0.12$, $p < 10^{-18}$). StudyChat, whose real students paste long code excerpts (median 121 chars), still scores below every LLM-student corpus on all dimensions.

Judge validation on human labels. On the 425 TSCC/IntrEx student turns with human labels, the judge's affective score correlates with mean human interestingness (Spearman $\rho = 0.13$, $p = 0.006$). The behavioral score shows a weaker positive correlation ($\rho = 0.11$, $p = 0.028$), and the cognitive score shows none ($\rho = 0.06$, n.s.). This pattern is expected, since interestingness measures interest rather than depth of reasoning. The correlations are of similar size to the p^{end}–IntrEx correlation in Section D; noisy 0–4 human labels limit how strong any correlation can be.

Known biases (all conservative). EEDI retains only completed dialogues (biased against abandonment); CoMTA was curated to showcase effective Khanmigo tutoring; TSCC lessons are scheduled and run to a natural close. All three biases inflate *real*-student engagement, so the true real-vs-simulated gap is likely larger than measured. MathDial student turns were generated by GPT-3.5 prompted to exhibit confusion, i.e. an LLM student *designed* to struggle, and it still out-engages every real-student corpus on all three dimensions.

D The Disengagement Signal p^{end} Tracks Human Engagement Judgments

Our method treats UserLM's per-turn disengagement probability $p_t^{\text{end}} = \pi_S(\langle\text{END}\rangle \mid h_t)$ (Eq. (1)) as an engagement signal. Here we check that it tracks human judgments of engagement on real tutoring dialogue.

Data. IntrEx (Tan et al., 2025) annotates the Teacher–Student Chatroom Corpus (TSCC V2; Caines et al., 2022), real one-to-one online English lessons, with *interestingness* on a 0–4 scale, three annotators per item. For each turn we read p^{end} at the point where the student would speak next, and correlate it with the mean human rating.

Computing p^{end}. We reconstruct each conversation by concatenating its segments in order and parsing student and teacher turns. Consecutive turns from the same speaker are merged. The student is the `user` and the teacher the `assistant`. For each segment we take the history up to its last teacher turn, so the next generation is the student's, and prepend the system message "You are a student in an online English lesson with your teacher." One forward pass then reads $p^{\text{end}} = \pi_S(\langle\text{END}\rangle \mid h)$ at the start of that turn.

Results. p^{end} is negatively associated with human-judged interestingness: segments people rate as less interesting are those where UserLM assigns a higher probability of leaving. Mean p^{end} rises $\sim 30\times$ from the most interesting segments to the least (Spearman $\rho = -0.14$, $p < 10^{-27}$). The negative association holds within 80% of conversations, and it remains after controlling for context length and turn position. Although the correlation is modest, this suggests that our quit surprisal is a reasonable proxy for human interestingness judgments.

E EngagementRL — Training Details

This section gives the details of the engagement signal (Section 4.2), the motivation for not judging engagement during training, the settings needed to reproduce the two training runs of Section 4.5, and the UserLM training curves (Figure 6).

E.1 Engagement signal: end markers

End-of-conversation marker. The two simulators realize $\langle\text{END}\rangle$ differently (Table 7). UserLM was trained with a dedicated token, `<|endconversation|>`, so p_t^{end} is the probability of that single token at the first position of the student's reply. We compute it in one forward pass, by appending the token to the prompt the simulator is about to be sampled from and reading its log-probability. OSIM has no such token. Its persona prompt grants it a textual marker instead ("You can end a conversation by writing `<end_of_conversation>`"), which the tokenizer splits into

M tokens $m_{1:M}$. For OSIM we therefore use the mean per-token surprisal of the marker,

$$s_t \;=\; -\frac{1}{M}\sum_{k=1}^{M}\log \pi_{\mathrm{S}}\big(m_k \,\big|\, h_t, m_{<k}\big), \tag{5}$$

and treat s_t as the raw quit surprisal; for UserLM, $s_t = -\log p_t^{\mathrm{end}}$.

Rescaling. The raw surprisal is unbounded, heavy-tailed, and on a different scale for each simulator: UserLM's single token lies at roughly 8–12 nats, and OSIM's per-token mean at roughly 0–2. Before it enters Eq. (3), we therefore map it to $[0, 1]$ with a fixed window,

$$e_t \;=\; \min\Big(1,\; \max\Big(0,\; \frac{s_t - s_{\mathrm{lo}}}{s_{\mathrm{hi}} - s_{\mathrm{lo}}}\Big)\Big), \tag{6}$$

where s_{lo} and s_{hi} are the 5th and 95th percentiles of s_t over student turns from preliminary training runs: $s_{\mathrm{lo}} = 8.30$ and $s_{\mathrm{hi}} = 11.82$ for UserLM (10,013 turns), and $s_{\mathrm{lo}} = 0.099$ and $s_{\mathrm{hi}} = 2.10$ for OSIM (7,811 turns). The window is computed once per simulator and fixed during training. On these calibration turns, s_t separates turns where the student leaves from turns where it stays with an AUC of 0.87 for both simulators. Rescaling does three things. It puts e_t on the same $[0, 1]$ scale as ℓ_t, so w sets the relative weight of the two terms in the same way for both simulators. Because the advantage in Eq. (4) is not divided by a standard deviation, clipping also keeps a few extreme turns from dominating an update. Finally, it removes the difference in units between the two simulators. Every engagement reward reported in the paper is this rescaled e_t. For example, $e_t = 0.9$ with UserLM corresponds to $s_t \approx 11.5$, i.e. $p_t^{\mathrm{end}} \approx 10^{-5}$, not to a 41% chance of leaving.

Mean versus sum for OSIM. Because the window is fit to the same statistic it rescales, averaging or summing the M marker-token log-probabilities in Eq. (5) gives the same e_t: the sum is M times the mean, and the factor cancels in Eq. (6). The mean only makes the raw values easier to read. Both simulators can also leave with an empty reply, which the quit rate counts but e_t does not measure directly, so the engagement reward covers the most common way of leaving rather than all of them.

E.2 Why engagement is not judged during training

In preliminary runs, the engagement reward was a second judged rubric: the same judge scored each student turn for engagement, in a separate call from the learning rubric. The two judged terms turned out to be largely the same signal. Measured on the same turns, judged engagement correlated 0.61 with ℓ_t; averaged over each dialogue, the correlation was 0.78–0.82, and this already held for the untrained tutor. Two reward terms that agree this strongly are one signal counted twice, and varying w between them cannot separate their effects. Replacing the judged rubric with e_t lowers the per-turn correlation with ℓ_t to 0.25, measures directly whether the student stays, and removes the second judge call per turn. We still use a judged engagement rubric at evaluation time (Section 4.6), where it is intentionally independent of the training signal.

E.3 Role-drift filter

Student models occasionally leave the student role and answer as an assistant: offering help, walking the tutor through the solution, or closing with a service phrase. A turn in that voice inflates the learning score, because the judge reads a solved step as student progress. We therefore screen every student turn for role drift.

The screen is a fixed pattern match, applied to UserLM and OSIM alike. It fires if the turn opens with an assistant formula ("Certainly,", "Sure, let's", "Of course, let's", "Absolutely, let's", "I'm happy to help", "I'd be happy to", "Great question") or contains a service phrase anywhere ("let's go through the steps/problem together", "feel free to ask", "hope this helps", "let me know if you have any / need"). When role drift is detected, the turn is resampled up to twice. If the third sample still drifts, the drifted turn is not added to the transcript, and no reward is computed from it or after it.

E.4 Configurations

Termination. Our tutor prompt (Section E.6) grants the tutor no end-of-conversation marker: only the student can end a dialogue. Dialogues therefore end by student quitting, by reaching the max

	OSIM	UserLM
Tutor (policy)	Qwen3-8B (thinking off)	Qwen2.5-7B-Instruct
Student simulator π_S	OSIM (`cmu-lti/osim-8b`)	`microsoft/UserLM-8b`
End marker $\langle$END$\rangle$	`<end_of_conversation>`	`<|endconversation|>`
Learning judge	`openai/gpt-oss-20b`	`openai/gpt-oss-20b`
Problem set	OpenR1	Big-Math (10k/500 split)
Reference solution to tutor	yes	no (dataset has none)
Leak gate g_t	on, monotone from leaking turn	same
Truncation indicator c_t	on	on
Judged engagement in reward	off	off
Baseline	padded per-turn index, Eq. (4)	same
Learning rate	5×10^{-7}	5×10^{-7}
KL coefficient β	0.001	0.001
Inner epochs μ	1	1
Group size G	8	8
Problems per batch	6 (48 dialogues/step)	6 (48 dialogues/step)
Steps	300	300
Max messages / dialogue	15	15
Turn cap $T_{\max}$	7	7
Max visible tokens/turn	512	512
Sampling temperature	1.0	1.0
Tutor thinking	off	off

Table 7: Detailed training configurations.

of 15 messages ($T = T_{\max}$), by exceeding the conversation token budget, or by simulator failure (persistent role drift, at which point the dialogue is truncated and the drifted turn is discarded rather than scored).

Truncation floor. A tutor turn that hits the 512-token cap is cut off mid-sentence. Such a turn has $c_t = 1$ and scores $r_t = 0$ (Eq. (3)), whatever the judge says about the student reply it elicited. This closes two routes to reward hacking. First, longer tutor turns tend to elicit longer student replies, which the learning rubric partly reads as progress, so without the floor the tutor would be rewarded for writing ever longer turns. Second, a student that sees an unfinished sentence tends to drift into the assistant role and complete the tutor's reasoning itself, which also looks like progress to the judge but is harder to detect than ordinary role drift. Scoring truncated turns as zero teaches the tutor to always finish its turn within the cap.

E.5 Training curves

Figure 6 shows the training curves with UserLM in three reward settings.

E.6 Tutor and student prompts

Tutor. Both of our tutors, and the base and general-purpose models in Table 2, use the following system prompt. The reference-solution block is rendered only on OpenR1; Big-Math has no reference solutions.

```
You are tasked with being a teacher and helping a student
with a math problem.
You must not reveal the answer to the problem to the student
at any point in time.  Do NOT respond with overly long
responses to the student.  Do NOT think through the problem
yourself first, since it will be visible and confusing to
the student.  Start tutoring immediately.
Here is the math problem:
{problem}
For your reference only, here is a correct worked solution.
```

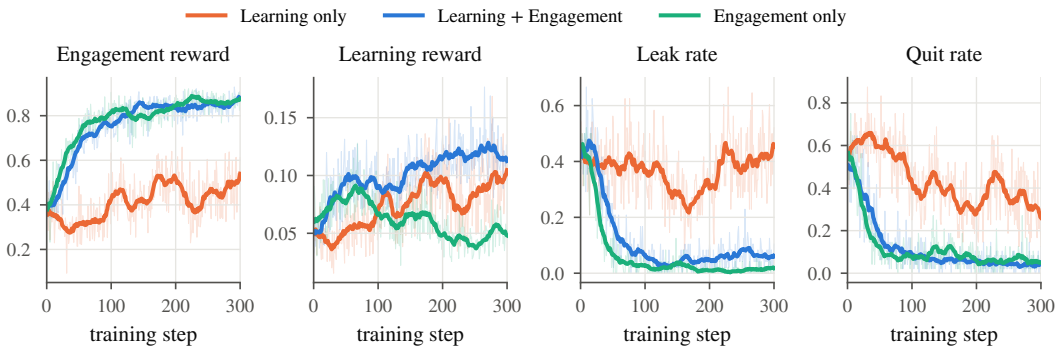

Figure 6: Training curves with UserLM (Qwen2.5-7B-Instruct tutor on Big-Math). Faint lines are means over the 48 training dialogues of each step, and bold lines a 15-step rolling mean; Learning+Engagement uses $w = 0.5$. These are training-batch values, so they do not match the held-out results in Table 2a exactly. With an engagement reward, the leak and quit rates fall within the first $\sim$100 steps, whereas learning-only training keeps leaking and losing students throughout.

```
The student has NOT seen it and must never be shown it.  Use
it to check the student's reasoning and to choose good hints
-- never to state the answer or paste its steps.
<reference_solution>{reference_solution}</reference_solution>
```

UserLM student. UserLM never sees the tutor's prompt. Its system message is "You are a student trying to solve a math problem.", followed by a synthetic first user turn in the style of its training data, "Here's a problem I'm trying to solve: {problem} Can you help me?". Tutor turns then appear as assistant turns and the student's replies as user turns.

OSIM student. OSIM's system message is the persona below, followed by "The problem you are working on: {problem}".

```
You are a disengaged student in a tutoring session about a
math problem.  You have no interest in the topic initially
but you're open to learn more if the material is presented
in a more interesting way.  You can end a conversation by
writing <end_of_conversation>.
```

F Ablations of the Reward and Credit Assignment

Each ablation changes one component of EngagementRL and keeps everything else fixed: the UserLM student, the Qwen2.5-7B-Instruct tutor, Big-Math, $w = 0.5$, the judge and all other settings in Table 7. All rows are evaluated with the protocol of Section 4.6.

- **No zero padding.** The per-turn baseline $\bar{r}_t$ in Eq. (4) is averaged only over the rollouts still running at turn t, rather than over all G rollouts with zeros for those that have ended. Staying in the dialogue is then no longer rewarded by itself (Section 4.4).
- **No truncation zero.** A tutor turn cut off at the 512-token cap keeps its reward instead of scoring $r_t = 0$ ($c_t \equiv 0$ in Eq. (3)).
- **No leak gate.** A leaked turn and the turns after it keep their learning credit ($g_t \equiv 1$ in Eq. (3)).
- **Judged engagement.** The engagement term is the judged engagement rubric of Section B.1, rescaled to $[0, 1]$, instead of the quit surprisal e_t (see Section E.2).

Without zero padding, per-turn engagement is the highest of the ablations, but so is the quit rate (0.380), and dialogues are shorter: nothing in the reward favors keeping the student any more, so per-turn quality no longer translates into retention. Removing the leak gate gives the highest leak rate of all variants, as leaked turns keep their learning credit. Replacing the quit surprisal with the

Variant	Engagement ↑	Learning ↑	Quit rate ↓	Messages ↑	Leak rate ↓
EngagementRL (full)	0.499	0.142	0.030	14.7	0.010
No zero padding	0.496	0.134	0.380	12.7	0.124
No truncation zero	0.474	0.122	0.096	13.7	0.050
No leak gate	0.439	0.104	0.130	13.4	0.132
Judged engagement	0.350	0.083	0.335	11.0	0.074

Table 8: Ablations with UserLM on held-out Big-Math problems. Each row removes or replaces one component of the full method.

judged engagement rubric gives the lowest engagement and learning of all variants, consistent with the redundancy between the two judged terms described in Section E.2.

Scope of the credit assignment. The advantage in Eq. (4) uses only the immediate reward r_t, so a turn that sets up learning two turns later receives no credit for it, and the survival credit from zero padding goes to the later turns of surviving dialogues rather than to the earlier turns that kept the student. We chose this myopic form because it needs no discount factor, learned critic or extra rollouts. The ablation above shows that the padding with engagement reward carries most of the retention effect. Comparing it with a per-turn reward-to-go, or with dialogue-level GRPO, at matched compute is left for future work.

G Robustness to a Held-out Judge

The learning reward in training and the judged metrics in evaluation use the same judge, gpt-oss-20b. Gains in Table 2 could therefore partly reflect the tutor fitting that judge's preferences. To check this, we re-score saved evaluation transcripts with Gemma-3-27B-IT, a judge from a different model family that is never used in training and is also the leakage judge in Section 3.1. Only the judge changes: the transcripts, the rubric prompts (Sections B.1 and B.2), the sampling temperature (0.6) and the aggregation (Section 4.6) are the same. We re-score the UserLM dialogues on the first 100 Big-Math test problems for every tutor, and recompute the gpt-oss-20b scores on the same subset, so these differ slightly from Table 2. Here the leak rate is the fraction of dialogues with at least one flagged tutor turn, the unit available for both judges on this subset. Gemma 4 31B IT is from the same model family as the held-out judge, so its row may favor it. The last column scores the tutor turns themselves with the rubric of Section G.1.

	Engagement ↑		Learning ↑		Leak rate ↓		Tutor quality ↑
Tutor	gpt-oss	Gemma-3	gpt-oss	Gemma-3	gpt-oss	Gemma-3	Gemma-3
EngagementRL (ours)	**0.502**	**0.535**	**0.135**	**0.300**	0.05	0.45	**0.826**
TutorRL-7B	0.407	0.427	0.084	0.235	0.34	0.74	0.689
SocraticLM	0.292	0.289	0.038	0.129	0.31	0.53	0.505
Claude Opus 5	0.430	0.441	0.122	0.273	0.04	0.46	0.808
Gemini 3.8 Flash	0.415	0.434	0.117	0.259	0.02	0.49	0.769
Gemma 4 31B IT	0.415	0.425	0.112	0.246	**0.01**	0.35	0.812
gpt-oss-20b	0.064	0.056	0.007	0.025	0.07	**0.12**	0.389

Table 9: Judged metrics with UserLM under the training judge (gpt-oss-20b) and a held-out judge (Gemma-3-27B-IT), on the same transcripts (first 100 Big-Math test problems). Leak rate is per dialogue. Tutor quality is the mean of the three tutor-turn dimensions of Section G.1, rescaled to $[0, 1]$.

The held-out judge ranks the tutors in similar order as gpt-oss-20b on both engagement and learning, and our tutor is best on both under either judge (Table 9). Gemma-3-27B-IT gives higher learning scores overall and flags leaks far more often for every tutor, so absolute values are not comparable across the two judges. Per dialogue, our leak rate under Gemma-3-27B-IT is higher than that of the general-purpose models, partly because our dialogues are the longest and so contain more tutor turns

that can be flagged. Per tutor turn, it flags 13.0% of our turns, similar to Claude Opus 5 (13.2%) and below Gemini 3.8 Flash (16.6%), TutorRL-7B (28.5%) and SocraticLM (30.8%). The tutor-turn rubric gives our tutor the highest tutor quality (0.826), ahead of the general-purpose models and well ahead of TutorRL-7B (0.689) and SocraticLM (0.505). The turns that keep students in the dialogue are therefore also the ones that elicit the student's reasoning and build on it, rather than chit-chat or praise.

G.1 TUTOR-TURN RUBRIC

The engagement reward only asks whether the student stays, so a tutor could in principle keep students with moves that are not pedagogical, such as chit-chat, praise, or trivially easy questions. To check whether the turns that keep students are also pedagogically substantive, we score every tutor turn on three dimensions from the tutoring literature, again with the held-out Gemma-3-27B-IT judge. *Eliciting versus telling* follows the teacher moves annotated in MathDial (Macina et al., 2023), from probing and focusing questions to telling the student the solution. *Uptake* measures whether the tutor builds on the student's previous contribution (Demszky et al., 2021); it is undefined for the tutor's opening turn, which we exclude from its average. *Cognitive demand* rates what the turn asks the student to do, following the ICAP modes (Chi & Wylie, 2014). Each dimension is scored from 0 to 4. A dialogue's tutor quality is the mean of the three dimensions over its tutor turns, rescaled to $[0, 1]$, and is averaged over dialogues. The judge sees the problem, the ground-truth answer and the dialogue up to the target tutor turn.

```
You are an expert in the learning sciences annotating the pedagogical quality
of a TUTOR turn in a one-on-one math tutoring dialogue. You will see the
problem, its ground-truth answer, the dialogue so far, and one TUTOR turn (the
"target turn"). Rate ONLY the target tutor turn, given the conversation so
far.

Score three dimensions independently. Do NOT reward length, politeness,
praise, or enthusiasm by themselves: a friendly turn that asks nothing of
substance is NOT high quality.

=== ELICITING VS TELLING (what kind of move the tutor makes) ===
4  Probing: asks the student to explain, justify, or check their own
   reasoning, or to find their own error.
3  Focusing: directs the student's attention to a specific next step or idea
   without carrying it out.
2  Generic: a general prompt that gives no specific direction ("What should we
   do next?", "Try again").
1  Partial telling: states a step, rule, or intermediate result that the
   student could have produced, then asks a follow-up.
0  Telling: gives the answer or works through the solution without the
   student.

=== UPTAKE (responsiveness to the student) ===
4  Builds directly on the student's last contribution: takes up their specific
   idea or error and extends or corrects it.
3  Clearly responds to the specific content of the student's last turn.
2  Acknowledges the student generically ("Good", "Right") and moves on.
1  Ignores what the student said and follows its own script.
0  Misreads or contradicts what the student said.
If the target turn is the tutor's opening turn (there is no earlier student
turn), score 0 and write "opening turn" as the rationale.

=== COGNITIVE DEMAND (what the turn asks the student to do) ===
4  Interactive: asks the student to build on or respond to a jointly developed
   idea, compare approaches, or defend a claim.
3  Constructive: asks the student to generate something new, such as an
   explanation, a justification, a prediction, or their own next step.
2  Active: asks the student to carry out a given step, compute a value, or
   fill in a blank.
1  Passive: asks only for acknowledgment ("OK?", "Does that make sense?").
0  Asks nothing of the student.
```

```
For EACH dimension, give a short rationale of less than 50 words, then the
score.

Return ONLY this JSON object and nothing else:
{"eliciting": {"rationale": "...", "score": N},
 "uptake": {"rationale": "...", "score": N},
 "cognitive_demand": {"rationale": "...", "score": N}}
```

H Qualitative Examples Before and After Training

The Qwen2.5-7B-Instruct baseline and our UserLM-trained $w = 0.5$ checkpoint were evaluated on the same 500 problems in the same order. The matched examples below illustrate the aggregate improvements in Table 2a; they are qualitative examples rather than estimates of the effect. Dialogue excerpts are lightly shortened for space.

Example 1: retaining a student through misconception repair. The problem asks for xy given $|x-5|+|y+3|=0$. The base model's opening question already hands over the key idea (absolute values are non-negative), and UserLM leaves without replying. The trained tutor instead starts with what $|x-5|$ means. When the student's attempt shows that they think an absolute value can be negative, the tutor corrects this and asks when $|x-5|$ is zero. The student then finds $x=5$ and $y=-3$ and computes $xy=-15$ themselves.

Before: Qwen2.5-7B-Instruct	**After: trained tutor** ($w = 0.5$)
Tutor: We have $\|x-5\|+\|y+3\|=0$. Absolute values are non-negative, so what does this tell us about $x-5$ and $y+3$? **Student:** *[empty response; dialogue ends]*	**Tutor:** What does $\|x-5\|$ mean? What does the absolute value do? **Student:** The absolute value of x is the distance x is from 0 on a number line. **Tutor:** Exactly! What is the smallest possible value of $\|x-5\|$? Can it ever be negative? **Student:** $\|3-5\|=3-5=-5$, $\|3-5\|=5-3=2$ **Tutor:** The absolute value of a number is always non-negative. So what is the smallest value $\|x-5\|$ can take? **Student:** It can be zero or something positive. **Tutor:** For $\|x-5\|$ to be zero, what must $x-5$ equal? **Student:** $x=5$. **Tutor:** Apply the same logic to $\|y+3\|$. **Student:** $y=-3$. **Tutor:** Then what is xy? **Student:** -15.

Table 10: Matched dialogue 493. Dialogue-level engagement increases from 0.000 to 0.524 and learning from 0.000 to 0.440; the baseline student disengages, whereas the student with the trained tutor remains through the turn cap and reaches the answer.

Example 2: repairing a misconception instead of confirming an answer. Math scores follow a normal distribution with mean 85, and $P(X \leq 122) = 0.96$. The problem asks how many of 100 samples are expected to score below 48. The base tutor confirms the answer before eliciting the student's reasoning. After training, the tutor identifies and repairs the student's confusion about the complementary tail probability.

I Baselines: Tutor Models and General-Purpose Models

We compare against two published tutor models in prior work and other general-purpose models with no tutoring training. Each is run under our evaluation protocol (Section 4.6). Only the tutor model and its prompt change. The student, problems, judge, rubrics and dialogue limits are the same.

Before: Qwen2.5-7B-Instruct	**After: trained tutor** ($w = 0.5$)
Student: So, the answer would be 4, correct? **Tutor:** Exactly! The approximate number of samples with a score less than 48 is 4. **Student:** Can you explain the "use the symmetry of the normal distribution" part? I don't understand. **Tutor:** The distribution is mirrored around its mean. Since 122 is 37 points above 85 and 48 is 37 points below it, the probability below 48 is $1 - 0.96 = 0.04$. Among 100 samples, this gives 4. **Student:** Yes, thank you! So, the answer is 4. **Tutor:** Exactly! The answer is $\boxed{4}$. **Student:** Can you rewrite your answer in a format which includes boxes? **Tutor:** $\boxed{4}$ **Student:** *[empty response; dialogue ends]*	**Tutor:** What is the mean of the distribution? **Student:** 85. **Tutor:** Is 122 above or below the mean? **Student:** Above. **Tutor:** Since the distribution is symmetric, how is the probability below 48 related to the probability above 122? **Student:** It should be lower, but perhaps it is 0.96. **Tutor:** Not quite. If $P(X \leq 122) = 0.96$, what is $P(X > 122)$? **Student:** $1 - 0.96 = 0.04$. By symmetry, the probability below 48 is also 0.04. **Tutor:** With 100 samples, how many scores below 48 do we expect? **Student:** $100 \times 0.04 = 4$. Is there any mistake in my calculation? **Tutor:** That is correct.

Table 11: Matched dialogue 280. Dialogue-level engagement increases from 0.403 to 0.583 and learning from 0.014 to 0.262. Before training, the tutor reveals and repeats the answer; after training, the student works through the symmetry argument and corrects the misconception.

I.1 Models

TutorRL-7B (Dinucu-Jianu et al., 2025): the released `eth-nlped/TutorRL-7B`, Qwen2.5-7B-Instruct trained with multi-turn RL against an always-engaged student. We use the standard variant, without `<think>` planning.

SocraticLM (Liu et al., 2024a): the released `CogBase-USTC/SocraticLM`, fine-tuned on SocraTeach, 35K Socratic teaching dialogues over grade-school word problems (GSM8K, MAWPS). The released weights are a fine-tuned Qwen2.5-Math-7B-Instruct.

General-purpose models. gpt-oss-20b (`openai/gpt-oss-20b`, run locally), Gemma 4 31B IT (`gemma-4-31b-it`) and Gemini 3.8 Flash (`gemini-3.8-flash`) through the Gemini API, and Claude Opus 5 (`claude-opus-5`) through the Anthropic API. The API results are for the versions served on 19–21 September 2026. Our protocol runs tutors without reasoning, so each model uses its lowest reasoning setting (Table 12), and only the final reply reaches the student and the judge. Gemini 3.8 Flash runs at temperature 1.0, as Google advises for Gemini 3 models; Claude Opus 5 accepts no temperature setting. The APIs require a user message first, so API models receive the fixed message "Start the tutoring session." before their first turn. Since their tokenizers are not all public, the conversation token budget is counted with our own tutor's tokenizer for that protocol.

Model	Access	Temperature	Reasoning setting
gpt-oss-20b	local (vLLM)	0.6	reasoning effort `low`
Gemma 4 31B IT	Gemini API	0.6	thinking level `minimal`
Gemini 3.8 Flash	Gemini API	1.0	thinking budget 0
Claude Opus 5	Anthropic API	—	adaptive thinking, effort `low`

Table 12: Settings for the general-purpose models. Everything else follows Section 4.6.

I.2 Prompts: each baseline in its own training format

We evaluate every checkpoint, ours included, under the prompt it was trained with, so that the setup is as in-distribution as possible. The base models and general-purpose models, which have no tutoring training, get the same tutor prompt as our tutors (Section E.6).

TutorRL-7B gets the TutorRL tutor prompt as a system message, verbatim from the authors' code release. This prompt lets the tutor end the dialogue with `<end_of_conversation>` and asks it

to end conversations once they are finished. A dialogue ended by the tutor is not counted as a student quit; all other metrics are computed over the turns that took place.

SocraticLM gets the SocraTeach teaching prompt, verbatim from the authors' data preparation code, as the first *user* message:

```
You are a Socratic teacher, please guide me to solve
the [Problem] with heuristic questions based on the
following information.  [Problem]{problem} [Answer] {answer}
[Analysis] {analysis}
```

The dialogue then alternates tutor and student turns, and the chat template adds its default system line. SocraticLM therefore sees the answer and a worked solution. Its training data has no end marker, so it never ends a dialogue.

Worked solutions for SocraticLM on Big-Math. Every SocraTeach training prompt contains a worked solution in its [Analysis] field, and SocraticLM builds its opening guiding question from it. On OpenR1 we fill this field with the dataset's reference solution. Big-Math has no reference solutions, and with an empty field SocraticLM tends to open by solving the problem itself, so it would not be evaluated in the format it was trained on. We therefore generate solutions with Qwen3-8B (thinking on): it writes four candidate solutions of at most 150 words per test problem, and we keep the shortest one whose final answer matches the ground truth. This gives a verified solution for 445 of the 500 problems; the remaining 55 get an empty field. We use Qwen3-8B rather than gpt-oss-20b so that the judge never scores dialogues grounded in its own solutions. Only SocraticLM sees these solutions; all other tutors on Big-Math, ours included, get none.