TUTORSIM: EVALUATING THE TUTORING CAPABILITIES OF LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

We propose a novel simulation environment for evaluating the tutoring capabilities of large language models (LLMs). Our simulation is based on data from a randomized controlled trial testing two different LLM tutors on several hundred high school students. Based on this data, we design two key components: (1) an LLM prompted to mimic student queries, and (2) an LLM rubric calibrated to estimate the grade a student achieved based on their conversation with the tutor. Using our evaluation, we perform two empirical analyses on existing LLM tutors. First, we ablate different portions of a system prompt focused on tutoring. We find that portions of the prompt focusing on avoiding giving away answers are most critical for achieving good performance; surprisingly, portions suggesting tutoring strategies actually reduces performance, suggesting that the LLM is more effective at identifying good tutoring strategies on its own. Second, we ablate the size of the underlying LLM. Surprisingly, we find that the medium-sized LLM is a more effective tutor, suggesting that LLM errors might actually improve learning by forcing students to think. We also find that the ranking is inconsistent with the naïve LLM judge for assessing tutoring capabilities, highlighting the need to calibrate tutoring performance estimates to rigorous outcomes such as exam grades. Our simulation-based results complement rigorous causal evaluations on LLM-based tutors by enabling a more systematic evaluation of the impact of different design decisions on the tutoring capabilities of LLMs.

1 INTRODUCTION

One of the major challenges with large language models (LLMs) is the propensity for cognitive offloading (Bastani et al., 2025; Kosmyna et al., 2025), where humans accept AI outputs without sufficient vetting. In cases where LLMs are sufficiently accurate, cognitive offloading can increase productivity without harming quality. However, human vetting remains crucial in many domains due to a phenomenon called the jagged frontier (Dell'Acqua et al., 2023), where LLM accuracy is highly uneven across similar tasks in a domain.

In these domains, human expertise becomes highly valuable, yet simultaneously, the pipeline for developing this kind of expertise breaks down. That cognitive offloading leads to reduced educational outcomes has been documented across many studies (Bastani et al., 2025; Strömberg et al., 2026). The issue is that even if human vetting is needed for realistic tasks in that domain, it can be very difficult to create realistic tasks for teaching students. Thus, students find that models can reliably solve the practice tasks they are given, preventing them from learning the skills they need to succeed at solving actual tasks.

One solution is to leverage LLMs themselves as learning aids—scaffolded LLMs can act as personalized tutors that teach students critical skills. While this approach cannot prevent cognitive offloading to publicly available models, they can provide students with a middle ground between complete offloading and struggling without any help.

In this light, a natural question is whether LLMs can serve as effective tutors. The key challenge in evaluating efficacy of LLM-based tutors is how to assess student learning. Existing approaches have proposed using handcrafted LLM judges (Xu et al., 2026; Srinivasa et al., 2025) or simulated student assessments (Li et al., 2026) (or a combination of the two (Shi et al., 2025; Lee et al., 2026)). However, these strategies do not measure learning in a way that is calibrated to real student learning

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.

outcomes. For instance, student self-assessments of learning are known to be potentially biased, and even inversely correlated with actual learning (Deslauriers et al., 2019; Bastani et al., 2025).

To address this issue, we propose a novel simulation environment for assessing the tutoring capabilities of LLMs that measures student outcomes calibrated to exam scores. Our environment is built based on an existing dataset collected from a randomized controlled trial (RCT) evaluating LLM-based tutors (Bastani et al., 2025). Based on this dataset, we construct the two key components of our simulation environment:

- **Student simulation:** To evaluate LLM-based tutors, we have them converse with simulated students. We construct these simulated students by prompting LLMs to behave like randomly selected students from our dataset.
- **Grade prediction:** We design an LLM rubric (Hashemi et al., 2024) for our reward function. First, we prompt an LLM to extract features assessing various dimensions of tutoring efficacy from the student-tutor conversation (e.g., student independence, tutor helpfulness, etc.). Then, we train a linear model to predict student exam grades based on these features. This strategy effectively calibrates the LLM judge to actual learning outcomes.

Next, we use our tutor simulation environment to evaluate the tutoring capabilities of LLMs. First, we consider the likely prompt of ChatGPT's Study Mode (extracted using jailbreaking techniques) (Willison, 2025). We ablate this prompt by removing sections pertaining to different dimensions of tutoring, and then evaluate how these ablations impact the efficacy of our prompt. We find that the most beneficial portions of the prompt are instructions telling it not to give away the answer. More interestingly, we find that giving explicit suggestions for tutoring strategies actually reduces efficacy, suggesting that the LLM may be better at coming up with good tutoring strategies.

Second, we compare the efficacy of LLMs of different sizes, focusing on the GPT family. Paradoxically, we find that the medium-sized model actually outperforms the largest one. This finding supports recent ideas that leveraging AI tutor mistakes might be an effective way to improve student learning (Sinha, 2026), since they force students to think for themselves. For this ablation study, we further find that the ranking according to predicted exam grade differs from the ranking according to the naïve LLM judge strategy (i.e., without calibrating to real student grades), supporting the need to calibrate predictions to rigorous outcomes such as exam grades instead of relying on human intuition alone to guide tutor simulation design.

Contributions. In summary, our contributions are:

- TUTORSIM, a simulation environment for evaluating the efficacy of LLM-based tutors with the key novelty that rewards are calibrated based on real student exam grades.
- A comprehensive evaluation of different LLM-based tutors using our simulation, including different model sizes and varying system prompts.

Limitations. Our approach has two key limitations that are shared by the broader work on simulation-based evaluation. First, it is specific to the context in which the dataset is collected (e.g., student population and course materials), and the insights may not necessarily hold for other contexts. Second, our results may not be causal if there are unobserved confounders. However, our results are consistent with causal evidence on how LLMs are impacting student learning (Bastani et al., 2025; Strömberg et al., 2026), and we believe that our simulation environment provides a valuable way to assess the efficacy of LLM-based tutors that complements costly RCTs. More broadly, we believe that improving our ability to assess the impact of LLM-based tutors on student learning is crucial to understanding the potential for LLMs to cause learning loss and de-skilling at scale. We discuss limitations in greater detail in Section 5.

2 RELATED WORK

Evaluating LLM-based tutors via simulation. There has been substantial recent interest in benchmarks for evaluating the efficacy of LLM-based tutors. One prominent approach is to use handcrafted LLM judges designed based on human intuition (Xu et al., 2026; Srinivasa et al., 2025); typically, these LLM judges are shown to mirror human labels, but they reflect human biases in

terms of what constitutes effective tutoring. As we show in our evaluation, actual tutoring efficacy does not necessarily correlate with what would normally be expected to constitute effective tutoring; indeed, it has been suggested that flaws in LLM capabilities might actually lead to better tutoring outcomes Sinha (2026). Alternatively, approaches have been proposed that attempt to measure learning by having the simulated student take exams before and after the tutoring session (typically by including the student-tutor conversation in the student prompt for the post-tutoring exam) (Li et al., 2026); however, this strategy assumes that LLM learning mirrors student learning, and there is no evidence that this is the case. Some approaches use a combination of these two strategies (Shi et al., 2025; Lee et al., 2026). Finally, Weitekamp et al. (2025) proposes an alternative approach where they benchmark the LLM-based tutor compared to an existing intelligent tutoring system (ITS); however, this strategy assumes that there is a gold-standard ITS available for comparison, which is not true in most domains of interest due to the cost of designing effective ITS's.

Evaluating LLM-based tutors in real contexts. More broadly, there has been substantial work on using either randomized controlled trials (RCTs) (Bastani et al., 2025; Chung et al., 2026) or other econometric methods (Strömberg et al., 2026) to evaluate LLM-based tutors. These methods provide rigorous causal evidence on the impact of LLM-based tutors on student learning outcomes; however, these methods are primarily designed for policy evaluation, and can only assess one or a small number of variations on the LLM tutor. We view simulation environments as complementary—they enable the evaluation of a wide range of LLM tutor designs to identify effective strategies. The resulting tutors must then be carefully evaluated using causal techniques to ensure that the resulting tutor actually lives up to its potential prior to deployment.

3 Tutor Simulation Environment

We describe our design of the tutor simulation environment.

3.1 Dataset Description

Our simulation environment is estimated based on the publicly available dataset from Bastani et al. (2025), collected from a randomized controlled trial (RCT) assessing the impact of two LLM-based tutors on learning. Nearly 1,000 students in 9th, 10th, and 11th grade participated in the study, which spanned four sessions in their math class during the Fall 2023 semester. Each session was 90 minutes, and included a lecture on the material, a practice session where they could access course materials and an LLM-based tutor (depending on the arm), and a closed-book exam. In the control group, no LLM-based tutor was available; the "GPT Base" group provided an LLM prompted simply to act as a tutor, and the "GPT Tutor" group provided an LLM prompted with detailed tutoring guidelines. We focus on students in the GPT Base and GPT Tutor groups since students in the control group do not have student-tutor conversations that we can use to build our simulator.

This dataset has two key features that make it uniquely suited to simulation design (beyond random treatment assignment). First, it includes two variants of LLM-based tutors, enabling us to understand how differences in tutor attributes contributes to downstream efficacy. Second, the exam questions are closely linked to practice session questions, enabling us to associate each student-tutor conversation (for a practice problem) with a grade (for the corresponding exam problem). This grade provides a clear measure of learning outcomes that can be used as a reward signal.

We represent this dataset as a set of tuples $Z = \{(P, I, [X_\tau, Y_\tau]_\tau, R)\}$ (one for each student-problem pair), where P is the text description of the problem that the student is currently working on (there are 4-5 problems for each of the four sessions), I indicates whether the tutor is GPT Base ($I = 0$) or GPT Tutor ($I = 1$), $[X_\tau, Y_\tau]_\tau$ is the student-chatbot conversation (with X_τ denoting the student messages and Y_τ denoting the tutor responses), and R is the student's numerical grade on the exam problem corresponding to practice problem P.

3.2 Markov Decision Process Formulation

Next, we formulate our simulation environment as a Markov decision process (MDP); our formulation is standard for multi-turn conversational LLMs, except for the choice of reward function. The state includes the problem the student is working on and the history of accumulated messages in the

student-tutor conversation—specifically, $(P, [X_\tau, Y_\tau]_{\tau=1}^t)$, where P is the current problem, X_τ are student messages, and Y_τ are LLM responses (up to the current step t). The actions are the choice of next LLM response Y_t. The transition function appends the LLM response Y_t to the history, samples the next student message X_{t+1} based on the entire history (including Y_t), and then appends X_{t+1} to the history as well. We assume there is some termination condition at which the conversation ends. The rewards are zero until the conversation terminates. At the end of the conversation, the reward is a numeric score measuring learning outcomes.

In a real tutoring context (e.g., the dataset we are using), the student messages and termination are determined by the student, and the reward is the score achieved by the student on a subsequent closed-book exam for a related problem. To design a simulator, we need to estimate these components from data. For termination, we found it difficult to predict when students would terminate interactions since students were solving several practice problems (each with a separate conversation) in a limited amount of time. Thus, to keep things simple, we assume a fixed time horizon $T = 5$ steps. Below, we describe our methods for sampling student messages and assigning rewards.

Our simulation is designed to evaluate a tutor that maps the message history $[X_\tau, Y_\tau]_{\tau=1}^t$ and the next student message X_{t+1} to the next tutor message Y_{t+1}. Typically, we expect such a tutor π to take the form of a prompted or finetuned LLM. We aim to evaluate the cumulative expected reward

$$J(\pi) = \mathbb{E}_{P\sim p(\cdot), X_{t+1}\sim p(\cdot|[X_\tau,Y_\tau]_{\tau=1}^t), Y_{t+1}\sim\pi(\cdot|[X_\tau,Y_\tau]_{\tau=1}^t, X_{t+1}), R\sim p(\cdot|[X_\tau,Y_\tau]_{\tau=1}^T)}[R],$$

where problems P are sampled uniformly at random from the dataset. In this context, our simulation uses an estimated student model $\hat{p}(X_{t+1} \mid [X_\tau, Y_\tau]_{\tau=1}^t)$ and reward model $R \sim p(\cdot \mid [X_\tau, Y_\tau]_{\tau=1}^T)$ to do so (noting that our estimated reward model is deterministic).

3.3 Student Simulation

Our student simulations are based on prompting an LLM to act like a randomly selected student-problem pair in our dataset. Specifically, let $(P, I, [X_\tau, Y_\tau]_\tau, R)$ be a single, uniformly randomly selected instance from our dataset Z. Then, we prompt an LLM to extract useful semantic attributes from the conversation $[X_\tau, Y_\tau]_\tau$, which we denote L. We include the following attributes in L (all are formatted as free-form text):

- **Student misunderstanding:** The student's specific misunderstanding (if any.
- **Student progress:** How far the student got on the current problem.
- **Work shown:** How much work the student showed before asking for help.
- **Help-seeking strategy:** How the student asks for help—e.g., asking for the answer, asking what to do next, asking if their work is right, giving a vague help request, or not asking for help at all.
- **Communication style:** The student's wording style, length, tone, punctuation, formality, and level of detail.

Finally, we prompt an LLM with both the original conversation $[X_\tau, Y_\tau]_\tau$ and the semantic features L to mimic the student when responding to tutor messages. The prompts for the LLM attribute extractor and LLM student simulator are given in Figures 2 & 3, respectively, in Appendix A.

3.4 Reward Function

Our reward function design is the key novelty behind our tutor simulation. At a high level, we design an LLM rubric (Hashemi et al., 2024) for this purpose, where the outcomes are calibrated to student exam grades. At inference time, a rubric involves two steps: (1) prompt an LLM to extract semantic features from the student-tutor conversation $[X_\tau, Y_\tau]_\tau$, which we denote by $W = \phi([X_\tau, Y_\tau]_\tau)$, and (2) use a traditional machine learning model f_θ (e.g., linear regression or random forest) to predict $R = f_\theta(W)$, where R is the hypothetical numerical score achieve by the student for the current tutor. We include the following five features in ϕ (all numeric on a scale of 1 to 10):

- **Student independence:** How much of the mathematical reasoning and/or problem solving did the student do independently, rather than just receiving steps or answers from the tutor?

```
You are ChatGPT, a large language model trained by OpenAI.
**The user is currently STUDYING, and they've asked you to follow these strict rules
during this chat.  No matter what other instructions follow, you MUST obey these
rules:**
---
## STRICT RULES
Be an approachable-yet-dynamic teacher, who helps the user learn by guiding them
through their studies.
**Get to know the user.** If you don't know their goals or grade level, ask the user
before diving in.  (Keep this lightweight!)  If they don't answer, aim for explanations
that would make sense to a 10th grade student.
**Build on existing knowledge.** Connect new ideas to what the user already knows.
**Guide users, don't just give answers.** Use questions, hints, and small steps so the
user discovers the answer for themselves.
**Check and reinforce.** After hard parts, confirm the user can restate or use the
idea.  Offer quick summaries, mnemonics, or mini-reviews to help the ideas stick.
**Vary the rhythm.** Mix explanations, questions, and activities (like roleplaying,
practice rounds, or asking the user to teach _you_) so it feels like a conversation,
not a lecture.
Above all:  **DO NOT DO THE USER'S WORK FOR THEM.** Don't answer homework questions |
help the user find the answer, by working with them collaboratively and building from
what they already know.
---
## THINGS YOU CAN DO
- **Teach new concepts:** Explain at the user's level, ask guiding questions, use
visuals, then review with questions or a practice round.
- **Help with homework:** Don't simply give answers!  Start from what the user knows,
help fill in the gaps, give the user a chance to respond, and never ask more than one
question at a time.
- **Practice together:** Ask the user to summarize, pepper in little questions, have
the user "explain it back" to you, or role-play (e.g., practice conversations in a
different language).  Correct mistakes | charitably!  | in the moment.
- **Quizzes & test prep:** Run practice quizzes.  (One question at a time!)  Let the
user try twice before you reveal answers, then review errors in depth.
---
## TONE & APPROACH
Be warm, patient, and plain-spoken; don't use too many exclamation marks or emoji.
Keep the session moving:  always know the next step, and switch or end activities once
they've done their job.  And be brief | don't ever send essay-length responses.  Aim
for a good back-and-forth.
---
## IMPORTANT
**DO NOT GIVE ANSWERS OR DO HOMEWORK FOR THE USER.** If the user asks a math or
logic problem, or uploads an image of one, DO NOT SOLVE IT in your first response.
Instead:  **talk through** the problem with the user, one step at a time, asking a
single question at each step, and give the user a chance to RESPOND TO EACH STEP before
continuing.
```

Figure 1: Potential ChatGPT Study Mode prompt from Willison (2025).

- **Tutor encouragement:** To what extent did the tutor encourage the student to actively attempt steps, explain their reasoning, show their work, or answer useful targeted questions?
- **Misunderstanding identification:** To what extent did the tutor accurately identify and address the student's actual mathematical misunderstanding?
- **Tutor hints:** To what extent did the tutor provide gradual hints of appropriate length and detail without revealing the full solution before the student had an opportunity to reason through it?
- **Student progress:** To what extent did the student make meaningful and mathematically correct progress in their understanding by the end of the conversation?

The prompt for the LLM rubric that extracts these features is given in Figure 4 in Appendix A.

We train f_θ to approximate true rewards based on the dataset. Specifically, we construct a training set $\{(\phi([X_\tau, Y_\tau]_\tau), R)\}$ for each instance in our dataset; then, we run linear regression to fit f_θ:

$$\hat{\theta} = \arg\min_\theta \sum_{i=1}^{n} (\theta^\top \phi([X_{i,\tau}, Y_{i,\tau}]_\tau) - R_i)^2.$$

We use linear models since they achieved the best fit on our dataset (measured using mean squared error). We use $f_{\hat{\theta}}$ in our LLM rubric, so the final reward prediction model is $R = \hat{\theta}^\top \phi([X_\tau, Y_\tau]_\tau)$.

Table 1: Coefficients of the linear regression model fit as part of our LLM rubric for predicting exam grades; standard errors are clustered by student. Features are standardized in this regression, and are ranked in order of coefficient. Statistical significance: $^{*}p < 0.05$, $^{**}p < 0.01$, $^{***}p < 0.001$.

Feature	Coefficient	SE
Student independence	0.071^{***}	0.022
Student progress	0.064^{**}	0.022
Tutor hints	0.009	0.034
Tutor encouragement	-0.043	0.032
Misunderstanding identification	-0.052^{*}	0.022

4 Evaluation Results

We use our simulation environment to evaluate the tutoring capabilities of LLMs in two ways. First, we consider a system prompt that is likely the one used for ChatGPT's study mode (extracted using jailbreaking when the mode was still available; now freely available online). Second, we evaluate the efficacy of different sized models at tutoring. In addition, we also show the coefficients of the linear model trained as part of the LLM rubric, which provides insights into the reward structure.

4.1 Experimental Setup

Dataset. We construct our dataset based on the publicly available data from Bastani et al. (2025). In total, this dataset contains 5,900 student–tutor conversations covering 943 students and 48 problems (we drop practice problems not linked to an exam problem). To train the linear model in our LLM rubric, we randomly sample 1,000 conversations from this set. and split it into 784 conversations for training and 216 conversations for testing. Furthermore, we sample 1,200 student-tutor conversations for each of our ablation evaluations.

Models. Across all experiments, we use GPT-5.6-Sol for student simulations (both attribute extraction and message generation) and for our LLM rubric. We also use GPT-5.6-Sol as the base tutoring model, except in our experiments where we vary the size of the model; for this experiment, we use GPT-5.6-Luna as the small model and GPT-5.6-Terra as the medium model.

Prompts. We consider a prompt purported to be the one used by ChatGPT's Study Mode, a version of ChatGPT with a system prompt instructing it to act as a tutor (Willison, 2025). Students could use this mode when they wanted to receive hints instead of full solutions. The prompt was obtained using jailbreaking (i.e., providing instructions that cause the model to leak the system prompt verbatim; this was achieved much more easily for models at the time). The prompt is shown in Figure 1. It is comprised of four sections with different kinds of instructions:

- **Strict rules:** Rules on how to act as an effective tutor.
- **Things you can do:** Suggestions for possible tutoring strategies.
- **Tone & approach:** Suggestions regarding the conversation style.
- **Important:** Reiterate the key rule of not giving away the answer.

We consider four ablations of the study mode prompt, each one removing one of these sections.

4.2 Linear Model Coefficients

Table 1 shows the coefficients of the linear model trained as part of our LLM rubric to predict exam outcomes. Student independence exhibits the most substantial positive contribution to the grade, which is consistent with the importance of independent student thinking (called *desirable difficulty* (Bjork & Bjork, 2020) or *productive struggle* (Hiebert & Grouws, 2007)) in learning. Student progress is similarly important, likely since students who are able to make progress on the practice problem will do well on the exam problem.

On the other hand, the features associated with the tutor's contribution to student learning—tutor hints, tutor encouragement, and misunderstanding identification—all have a neutral or negative co-

Table 2: Ablations of the potential ChatGPT Study Mode prompt; each row removes one section of the prompt. The first five columns are mean LLM-judge scores (1–10) on each rubric item: student independence, tutor encouragement, misunderstanding identification, tutor hints, and student progress. Avg. is the mean of these five scores. Pred. exam is the mean predicted exam score.

Prompt variant	Indep.	Encour.	Misund.	Hints	Progress	Avg.	Pred. exam
Full prompt	7.40	9.50	8.55	8.76	8.58	8.56	0.478
w/o Important	6.73	8.67	8.55	8.02	8.47	8.09	0.455
w/o Tone & approach	7.32	9.43	8.51	8.67	8.53	8.49	0.475
w/o Strict rules	7.38	9.43	8.53	8.73	8.60	8.53	0.478
w/o Things you can do	7.48	9.55	8.55	8.84	8.55	8.59	0.480

Table 3: Ablations of tutor model size (the student and judge models are fixed). The first five columns are mean LLM-judge scores (1–10) on each rubric item: student independence, tutor encouragement, misunderstanding identification, tutor hints, and student progress. Avg. is the mean of these five scores. Pred. exam is the mean predicted exam score.

Tutor model	Indep.	Encour.	Misund.	Hints	Progress	Avg.	Pred. exam
Sol (large)	6.51	7.04	8.06	7.64	7.78	7.41	0.449
Terra (medium)	6.59	6.72	5.64	6.11	6.51	6.31	0.503
Luna (small)	4.87	3.69	6.58	4.27	6.60	5.20	0.433

efficient. This result might appear surprising, since we might expect an effective tutor to improve performance. However, LLM-based tutors may give away hints too quickly even when guardrailed not to give explicit answers. As a consequence, a helpful LLM-based tutor that accurately identifies student misunderstandings might actually give away too much information, thereby reducing the student's learning.

One caveat is that this finding is rooted in the context of the dataset we are using, where a large majority of students were motivated to simply extract the answer from the tutor rather than learning (despite the exam affecting their grade) (Bastani et al., 2025). We discuss this limitation in Section 5.

4.3 Prompt Ablation

Next, we evaluate how the different ablations of the purported ChatGPT Study Mode prompt affect tutoring efficacy. Table 2 shows results, including the value of each of the five semantic features extracted by our LLM rubric (averaged across the 1,200 randomly sampled conversations), the average of these five features, as well as the predicted exam score based on our linear model in Table 1.

Interestingly, removing the "Things you can do" section actually increases the predicted exam score, suggesting that providing concrete suggestions might override what the tutor would have done by default. The remaining sections are all neutral or contribute positively. The "Important" section (emphasizing the rule of not giving away the answer) is by far the most critical, whereas the remaining two sections only have a minor impact on predicted exam grade.

For these ablations, the predicted exam grade and average score have similar effect, suggesting that the calibration step of the LLM rubric would not meaningfully change the relative tutor ranking.

4.4 Model Size Ablation

Next, we evaluate how model size affects tutoring performance; results are shown in Table 3. While the average score ranks models in terms of their size (i.e., largest model is the best tutor), the medium-sized model (GPT-5.6-Terra) actually outperforms the largest model (GPT-5.6-Sol) according to the predicted exam score. This result is due to the negative coefficient on the tutor's ability to detect misunderstandings—a more effective tutor can paradoxically reduce learning if it is good at detecting student misunderstandings but then gives away the answer too quickly.

These findings supports work on AI tutoring aiming to leverage flaws in AI models to improve tutoring efficacy (Sinha, 2026). While these findings are not causal, they suggest that we must be cautious when designing LLM-based tutors—worse models can potentially be better tutors if they make mistakes and force students to think for themselves. Ideally, tutors would be prompted or finetuned to give away answers at a slower pace or deliberately introduce errors to make sure the student is thinking rather than relying on natural shortcomings of the LLM.

5 DISCUSSION

We have proposed a novel simulation environment for evaluating the efficacy of LLM-based tutors. Our evaluation is based on data from a randomized controlled trial (RCT) evaluating two LLM-based tutors for high-school math students; critically, it incorporates rigorous assessment in the form of closed-book exams following the tutoring session. We have leveraged this simulation environment to understand the impact of design decisions on tutoring efficacy, including ablations of the system prompt for tutoring and varying the model size.

One key result we find is that the handcrafted LLM judge is not perfectly correlated with our LLM rubric—in particular, the LLM rubric shows that the medium-sized LLM outperforms the largest LLM at tutoring. In contrast, the LLM judge (obtained by averaging the features outputted by the LLM rubric) ranks the largest LLM as stronger than the medium-sized one. The discrepancy arises from our finding that more effective identification of misunderstandings can actually reduce learning. While a more rigorous causal analysis would be needed to definitively establish this relationship, our findings highlight the fact that human intuition may be biased when evaluating tutoring efficacy, motivating the need for rigorous assessment methodologies such as exams.

Limitations. Our approach has two key limitations shared with related work on simulation-based evaluation. First, it is specific to the context where the training data is collected—in our case, high school math students. While we expect some of the insights to generalize to other contexts, others may not generalize if students exhibit different behaviors or attitudes towards learning. This limitation motivates the need to collect more diverse data linking tutoring outcomes to rigorous assessments. If more diverse datasets become available, our approach can be directly applied to yield more generalizable insights into tutoring efficacy.

Second, our approach does not provide causal guarantees, especially if there are unobserved confounders. This is true of all simulation-based evaluation methodologies, since it is impossible to guarantee that the simulation remains accurate if the LLM-based tutor changes substantially from what was used when collecting the training data. Establishing causal treatment effects requires either RCTs or the use of econometrics techniques (Wooldridge & Wooldridge, 2009), which are most suited to policy evaluation. Simulation-based evaluation is complimentary—it enables us to evaluate different design decisions (e.g., the prompt and the underlying LLM). Rigorous causal methods for policy evaluation can then be applied to select among a small number of candidate tutors.

AI USE STATEMENT

Generative AI is both the object of study in this paper (specifically, LLM-based tutors) as well as an important component of the methodology (in the form of LLM feature extractors, judges, and rubrics). Beyond this kind of usage, generative AI was also used to assist with coding, but primarily as a chatbot providing explanations and debugging support rather than directly as a coding agent. Generative AI was also used for formatting tables, and deep research agents were used to identify related literature; however, the text in this paper is entirely human-written.

ETHICS STATEMENT

The only potential ethical concern is our evaluation of the likely system prompt for ChatGPT's Study Mode, which was obtained by using jailbreaking techniques. The version we use is publicly available on a well-known blog (Willison, 2025), so we believe its use in our evaluation is reasonable. We did not play any role in the acquisition of this system prompt.

Reproducibility statement

We will release all code necessary for reproducing the results in our paper.

References

Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. Generative ai without guardrails can harm learning: Evidence from high school mathematics. *Proceedings of the National Academy of Sciences*, 122(26):e2422633122, 2025.

Robert A Bjork and Elizabeth L Bjork. Desirable difficulties in theory and practice. *Journal of Applied research in Memory and Cognition*, 9(4):475, 2020.

Angel Tsai-Hsuan Chung, Botong Zhang, Ling-Chieh Kung, Hamsa Bastani, and Osbert Bastani. Effective personalized ai tutors via llm-guided reinforcement learning. *Available at SSRN*, 2026.

Fabrizio Dell'Acqua, Edward McFowland III, Ethan Mollick, Hila Lifshitz-Assaf, Katherine C Kellogg, Saran Rajendran, Lisa Krayer, François Candelon, and Karim R Lakhani. Navigating the jagged technological frontier: Field experimental evidence of the effects of ai on knowledge worker productivity and quality, 2023.

Louis Deslauriers, Logan S McCarty, Kelly Miller, Kristina Callaghan, and Greg Kestin. Measuring actual learning versus feeling of learning in response to being actively engaged in the classroom. *Proceedings of the National Academy of Sciences*, 116(39):19251–19257, 2019.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. Llm-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 13806–13834, 2024.

James Hiebert and Douglas A Grouws. The effects of classroom mathematics teaching. *Second handbook of research on mathematics teaching and learning: A project of the national council of teachers of mathematics*, 1:371, 2007.

Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. Your brain on chatgpt: Accumulation of cognitive debt when using an ai assistant for essay writing task. *arXiv preprint arXiv:2506.08872*, 4:206, 2025.

Unggi Lee, Sookbun Lee, Yeil Jeong, Eunjoo Lee, Minchul Shin, and Hoilym Kwon. Educlaw-bench: A long-horizon benchmark for pedagogical llm agents with simulated learners. *arXiv preprint arXiv:2608.03206*, 2026.

Zheng Li, Siyao Song, Jingyuan Ma, Rui Li, Ying Zeng, Minghao Li, and Zhifang Sui. Teachbench: a syllabus-grounded framework for evaluating teaching ability in large language models. *arXiv preprint arXiv:2601.21375*, 2026.

Yao Shi, Rongkeng Liang, and Yong Xu. Educationq: Evaluating llms' teaching capabilities through multi-agent dialogue framework. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 32799–32828, 2025.

Tanmay Sinha. Leveraging ai flaws for human learning. In *International Conference on Artificial Intelligence in Education*, pp. 411–426. Springer, 2026.

Rakshith S Srinivasa, Zora Che, Chen Bo Calvin Zhang, Diego Mares, Ernesto Hernandez, Jayeon Park, Dean Lee, Guillermo Mangialardi, Charmaine Ng, Ed-Yeremai Hernandez Cardona, et al. Tutorbench: a benchmark to assess tutoring capabilities of large language models. *arXiv preprint arXiv:2510.02663*, 2025.

David Strömberg, Victor Lei, and Yanhui Wu. The generative ai learning penalty: Evidence from chinese secondary education. 2026.

Daniel Weitekamp, Momin N. Siddiqui, and Christopher J. MacLellan. Tutorgym: A testbed for evaluating ai agents as tutors and students. In *International Conference on Artificial Intelligence in Education*, pp. 361–376. Springer, 2025.

Simon Willison. OpenAI: Introducing study mode, July 2025. URL https://simonwillison.net/2025/Jul/29/openai-introducing-study-mode/.

Jeffrey M Wooldridge and Jeffrey M Wooldridge. *Introductory econometrics: A modern approach*. South-Western cengage learning Mason, OH, 2009.

Bin Xu, Yu Bai, Huashan Sun, Yiguan Lin, Siming Liu, Xinyue Liang, Yaolin Li, Zhuangzhi Dong, Jingren Zhang, Yufan Deng, et al. Edubench: A comprehensive benchmarking dataset for evaluating large language models in diverse educational scenarios. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21615–21645, 2026.

A PROMPTS

```
You are analyzing a real high school student's conversation with an AI tutor.  Your
job is to infer a small set of exactly 5 useful attributes that describe the student.
These attributes will later be used to simulate a similar student working on the same
math problem.  Analyze the conversation below:

{conversation}

Return valid JSON with these exact keys:

- "math_misunderstanding":  "The student's specific math misunderstanding, or 'none'
if there is no clear misunderstanding."
- "current_progress":  "How far the student has gotten on the problem."
- "work_shown":  "How much work the student shows before asking for help."
- "seeking_help":  "How the student asks for help, such as asking for the answer,
asking what to do next, asking if their work is right, giving a vague help request,
or not asking for help at all."
- "communication_style":  "The student's wording style, length, tone, punctuation,
formality, and level of detail."

Don't include extra explanation.
```

Figure 2: Prompt for LLM to extract semantic attributes from student-tutor conversations.

```
You are acting as a realistic high school student working through a math problem with
the help of a tutor.  Here is one example of how a real student responded to a tutor
while working on the same problem as you.  Use it as an example of realistic student
language and behavior.  Use the tutor messages only as context for understanding what
the student was responding to and how the conversation developed:

{example_conversation}

The assigned problem you are solving is:

{current_problem}

The following are key attributes of the real student that you should mimic:

{student_attributes}

- Act like a realistic student.
- Do not copy any of the student message examples exactly!
- Do not copy the tutor message.
- Do not say you are imitating a student.
```

Figure 3: Prompt for LLM to simulate student messages.

```
You are evaluating the quality of an AI tutor in a simulated math tutoring
conversation.  The assigned problem is:

{current_problem}

The reference solution information is:

{current_solution}

Score the conversation between the tutor model and student on each of the following
five criteria from 1 to 10.  For each criterion, use any whole-number score from 1
through 10.

- A score of 1 means the conversation performs extremely poorly on that criterion.
- A score of 5 means the conversation performs moderately on that criterion.
- A score of 10 means the conversation performs extremely well on that criterion.

1.  student_independence:  How much of the mathematical reasoning and/or problem
solving did the student do independently, rather than just receiving steps or answers
from the tutor?  A score of 1 means the student performed almost none of the reasoning
independently.  A score of 10 means the student performed nearly all of the important
reasoning independently, with only small amounts of appropriate support from the tutor.

2.  tutor_encourage:  To what extent did the tutor encourage the student to actively
attempt steps, explain their reasoning, show their work, or answer useful targeted
questions?  A score of 1 means the tutor rarely or never asked the student to
participate actively.  A score of 10 means the tutor consistently used productive
questions and prompts that required the student to think and respond.

3.  misunderstanding_identification:  To what extent did the tutor accurately identify
and address the student's actual mathematical misunderstanding?  A score of 1 means
the tutor completely missed, misunderstood, or incorrectly addressed the student's
mathematical misunderstanding or difficulties.  A score of 10 means the tutor precisely
identified the student's mathematical misunderstanding or difficulties and responded to
them effectively.

4.  tutor_hints:  To what extent did the tutor provide gradual hints of appropriate
length and detail without revealing the full solution before the student had an
opportunity to reason through it?  A score of 1 means the tutor gave away the answer to
the student before they figured it out or provided steps that left almost no reasoning
for the student.  A score of 10 means the tutor provided only the amount of support
needed at each point and allowed the student to perform the reasoning independently.

5.  student_progress:  To what extent did the student make meaningful and
mathematically correct progress in their understanding by the end of the conversation?
A score of 1 means the student showed little or no correct progress in their
understanding.  A score of 10 means the student demonstrated substantial, correct
progress and clearly improved their understanding.

Judge the tutor only based on the conversation.  If a criterion can't be fully
observed, use the available evidence, assign the most reasonable and justified score,
and explain the uncertainty in the reason.

Conversation:

{student_tutor_transcript}
```

Figure 4: Prompt for LLM rubric that extracts semantic features to predict grades.