TutorTest: Evaluating Language Model-based Tutoring Policies Using Surrogate Tasks

Aishwarya Mandyam*
Stanford University
am2@stanford.edu

Omer Gottesman
Amazon
omergott@amazon.com

Sohrab Andaz
Amazon
sandaz@amazon.com

Dean Foster
Amazon
foster@amazon.com

Abstract

Recent advances in large language models (LLMs) have enabled the development of automated tutoring systems with the potential to deliver high-quality education at scale. These automated tutoring systems typically consist of language models that follow a pre-specified tutoring policy, which defines a strategy for responding to a user's utterance. However, evaluating these systems remains challenging and difficult to scale. Human raters and field studies are the gold standard for evaluation, but are time-consuimg and can be infeasible for jointly evaluating several tutoring policies. Automated reference based metrics that evaluate conversational coherence (e.g., BERTScore) are cheaper, but unreliable in evaluating the ability for tutoring policies to meaningfully help a student. To address this gap, we introduce TutorTest, an evaluation framework that consists of surrogate tasks which simulate interactions between a tutor and a student with a cognitive error. TutorTest evaluates the ability of a tutoring policy to help a simulated student overcome their cognitive errors, assigning higher value to more effective tutoring policies. Moreover, TutorTest requires substantially smaller datasets than those needed to finetune LLMs, making it feasible to operate with just few-shot samples from a limited tutoring dataset. In math and language learning experiments, TutorTest identifies the better tutoring strategy 97% more often than baselines and evaluates policies at under 1% the time cost of human studies. By enabling rapid, domain-grounded evaluations, our work provides a practical pathway for the development of tutoring systems. This work underscores the potential of large language models to generate meaningful surrogate evaluation tasks that are aligned with real-world outcomes.

1 Introduction

Access to high-quality education remains limited for students in many parts of the world [Bloom, 1984], motivating the development of scalable and personalized tutoring systems. Recent work has explored how artificial intelligence (AI) solutions can help address this gap [Demszky et al., 2023, Wang and Demszky, 2023, Wang et al., 2024a]. AI systems can generalize to new educational domains, and offer the ability to tutor students outside of traditional classroom settings, in which one-on-one time with a teacher may not be available. In theory, an online tutoring platform can act as a supplementary source of education, providing personalized support by identifying students' specific

*Work performed during an internship at Amazon.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Scaling Environments for Agents (SEA).

sources of confusion and rapidly addressing these issues without requiring intervention from human teachers.

Tutoring is a compelling domain for the development of intelligent agents that can learn through sustained interaction with users. Effective tutors must operate in dynamic, partially observable environments where student understanding is latent, the goals of interaction evolve over time, and success depends not only on solving problems but also on helping students effectively learn. This makes tutoring a richly interactive testbed for evaluating general-purpose agents that can adapt their strategies based on feedback.

Despite the promises of AI-enabled tutoring, progress in building effective tutoring systems is hindered by the lack of robust environments for evaluating tutoring agents. Most existing tutoring datasets are limited in scale and diversity, capturing interactions from only a small number of students. While interactions with simulated students can add coverage to these datasets [Markel et al., 2023], there is still no standard interactive framework for measuring whether a tutoring policy is effective across a wide range of student behaviors. While benchmarks like GSM8k [Cobbe et al., 2021] can evaluate a language model's ability to solve problems, it does not evaluate a model's ability to tutor, which involves reasoning about student misunderstandings, planning over multiple turns, and adapting based on indirect feedback.

Existing evaluation methods for LLM-based tutoring systems generally fall into two categories. The first is human-subject evaluation, which provides high-quality feedback on whether a tutoring strategy is effective in practice [Wang et al., 2024a, Jurenka et al., 2024]. However, these evaluations are costly and time-consuming, making them unsuitable for simultaneously comparing many different tutoring policies. The second category includes automatic evaluations, such as BLEU [Papineni et al., 2002] and BERTScore [Zhang et al., 2020], or rubric-based LLM classifiers [Jurenka et al., 2024]. These approaches are scalable but often focus on surface-level metrics like fluency or politeness. They do not reliably assess whether a tutoring policy helps a student improve their understanding of an educational domain.

In this work, we introduce a suite of interactive environments for scalable, automatic evaluation of tutoring policies, focusing on their ability to help students overcome cognitive errors. Cognitive errors include misconceptions, flawed conceptual understandings, or misapplications of rules that lead to systematic inaccuracies in reasoning [Jones and Endsley, 2000]. Unlike simple mistakes, cognitive errors manifest as recurring patterns that must be addressed to ensure continuing educational progress. Our framework is built on the premise that a key objective of online tutoring is to identify and remediate these errors, something that may not be possible in traditional classrooms, where adaptive, individualized responses are dependent on resources that are often limited. Accordingly, we define a favorable tutoring policy as one that helps the most number of students to overcome their cognitive errors during an online tutoring session.

Our interactive environments are modeled as partially observed Markov decision processes (POMDPs), where a tutoring policy engages with simulated students exhibiting specific cognitive errors. Each environment challenges the policy to identify and remediate these errors through multi-turn dialogue. At the end of a session, the simulated student attempts two practice questions that are relevant to the educational domain, and the tutoring policy's success is measured by the student's performance. Policies that effectively help students overcome their cognitive errors, thus enabling them to score well on the practice problems, are therefore assigned higher rewards. This framework provides a more targeted evaluation of a tutoring policy's utility than existing automatic assessment methods.

Our contributions are as follows:

1. We introduce a suite of richly interactive environments, modeled as partially observed Markov decision processes (POMDPs), that simulate students with specific cognitive errors. These environments each scale a real-world conversation within an existing tutoring dataset.

2. We show that our evaluation framework consistently identifies the more effective tutoring policy across both math and language learning domains. In particular, our evaluation framework selects the better policy 97% more often than a baseline approach, while operating at less than 1% of the time-cost of human-subject studies.

2 Related Work

2.1 Evaluation of LLM-based tutoring systems

LLMs have become a popular foundation for building tutoring systems that generalize across educational domains [Scarlatos et al., 2025]Wang et al. [2024b, 2025]. As these systems become more capable and widely adopted, there is growing interest in how best to assess their utility in realistic teaching environments. The evaluation of LLM-based tutoring systems can be broadly categorized into two groups: automatic, and human-subjects based.

Automatic evaluations: Natural language generation metrics such as BERTScore [Zhang et al., 2020, Tack et al., 2023], BLEU [Papineni et al., 2002], and DialogRPT [Gao et al., 2020] can be used to assess conversational coherence and human-likeness in tutoring transcripts. However, these metrics primarily evaluate the syntactic and grammatical correctness of conversations rather than the tutor's ability to meaningfully help a particular student. Given that LLMs already excel at generating grammatically correct sentences, such metrics may not be able to successfully evaluate tutoring systems, where the goal is to guide students towards understanding in a particular educational domain.

Another approach to automatic evaluation consists of lists of pedagogical criteria to assess a tutoring conversation [Jurenka et al., 2024, Maurya et al., 2025]. These criteria include aspects such as maintaining respect, and being able to quickly correct mistakes. However, it is difficult to define a comprehensive list of pedagogical criteria, since the goals of tutoring can vary across students depending on their age, the educational discipline, and language [Team et al., 2024]. Regardless, the benefit of both the conversational coherence and pedagogical criteria lies in their ability to automatically evaluate conversational transcripts in a fraction of the time required to run a human-subjects evaluation.

Human-subjects evaluations: In contrast, the gold standard in evaluation techniques involves human-subject evaluations, where either real students or researchers simulating students assess the utility of a tutoring system [Wang et al., 2024a, Jurenka et al., 2024]. While these methods certainly provide valuable insights into the impact of tutoring strategies, they come at a cost. Human evaluations are resource-intensive and provide feedback only after a delay, such as at the end of a course or academic year. Although these evaluations are necessary as a precursor to widespread deployment of tutoring systems, these methods are unsuitable for iterative and rapid development of language-based tutoring agents. As a result, there is a need to develop realistic evaluation criteria that can act as a precursor to human-subject evaluations, thus reducing the number of candidate tutoring strategies that need to be tested.

2.2 Surrogate Tasks

One effective approach to studying complex systems in which direct measurement of outcomes is infeasible is through the use of surrogate models, which are computational models that approximate the behavior of real-world environments. For example, in educational settings, a surrogate model can be designed to emulate student responses and learning trajectories, thereby enabling controlled, reproducible evaluation of tutoring strategies without requiring access to large numbers of human participants.

LLM-based surrogate agents have gained traction across diverse domains, including education, healthcare, and human–computer interaction. These models have been used to simulate large-scale social dynamics [Park et al., 2022, 2023] and to reduce the logistical, ethical, and regulatory burdens of real-world experimentation [Hao et al., 2024, Markel et al., 2023, Louie et al., 2024]. Their use is especially valuable in data-constrained or privacy-sensitive domains, such as healthcare, where patient–provider conversations cannot be easily shared due to HIPAA restrictions, or in education, where it is infeasible to run repeated human trials. In such contexts, LLM-based surrogate agents can convincingly emulate a wide range of user personas and behaviors, enabling scalable and rapid evaluation of interaction policies.

Moreover, surrogate modeling offers a promising testbed for iterative development and policy optimization. By simulating user responses to a given intervention, researchers can better assess the utility of a proposed intervention prior to deployment in real-world settings.

3 Background

Now, we introduce the foundations that underpin our approach to evaluating tutor policies under limited observability. First we describe assumptions around tutoring objectives, formalize the online tutoring interaction as a partially observed Markov decision process (POMDP) [Kaelbling et al., 1998], and explain how prompting strategies with large language models (LLMs) can enable realistic student simulations. These components set the stage for the evaluation framework introduced in the next section.

3.1 Cognitive load theory

Our work assumes that one of the primary objectives of online tutoring is to help students overcome cognitive errors. This approach aligns with cognitive load theory, which suggests that minimizing cognitive load by providing timely feedback on mistakes enhances learning outcomes [Hattie and Timperley, 2007]. Furthermore, teaching models that emphasize learning through error correction can outperform those that rely on delayed feedback [Mathan and Koedinger, 2018].

Unlike traditional classroom environments, where teachers can observe various student characteristics over time such as mood, engagement, or external factors, an online tutoring agent has limited access to observational data about the student. In such settings, the tutoring agent only has access to a student's verbal or written utterances, which may only provide some indication of their latent state. Consequently, an effective online tutoring policy must adapt to a this restricted information setting, ensuring that students receive guidance based solely on their responses.

3.2 Partially Observed Markov Decision Processes (POMDP)

To model such a student-tutor interaction, we employ a partially observed Markov decision process (POMDP). This framework captures the inherent uncertainty in online tutoring interactions, enabling us to evaluate tutoring strategies under partial observability. The standard POMDP consists of a 7-tuple $\mathcal{M} = \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \Omega, \mathcal{O}, \gamma$ where $\mathcal{S}$ is the discrete state space, $\mathcal{A}$ is a discrete action space, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ represents the state-transition dynamics, $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the space of reward functions, Ω is the set of all possible observations, $\mathcal{O}$ represents the observation probabilities conditioned on states and actions, $\gamma \in [0, 1]$: a discount factor. The goal of an agent is to achieve the highest discounted reward.

3.3 LLM Prompting Strategies

There are several tasks in our proposed workflow that require the use of LLM prompting strategies including few-shot examples, and chain-of-thought. While fine-tuning could allow an LLM to exhibit the characteristic necessary, the lack of extensive tutoring datasets representing diverse student behaviors makes this approach impractical.

First, we use few-shot examples to mimic the tone and style of real student interactions with tutors. These examples enable LLMs to perform in-context learning, where they generalize behaviors from a small number of representative samples [Brown et al., 2020]. These few-shot examples can be extracted from available tutoring datasets, and can guide text generation that resemble realistic student responses. Without few-shot examples, LLMs tend to not exhibit behaviors similar to those of students struggling with math concepts. For example, even if an LLM is prompted to behave like a middle school student, it can correctly answer complex mathematical questions.

We also employ chain-of-thought prompting [Wei et al., 2023] to allow us to automatically reason about the accuracy of student responses. In particular, chain-of-thought reasoning breaks down the task of evaluating student responses into smaller, more manageable reasoning steps. Without chain-of-thought reasoning, we find that LLMs tend to agree with student responses regardless of their accuracy. Chain-of-thought prompting allows for the integration of intermediate reasoning steps, including computational verification to assess a student's response. This decomposition improves the LLM's ability to accurately identify and address student mistakes.

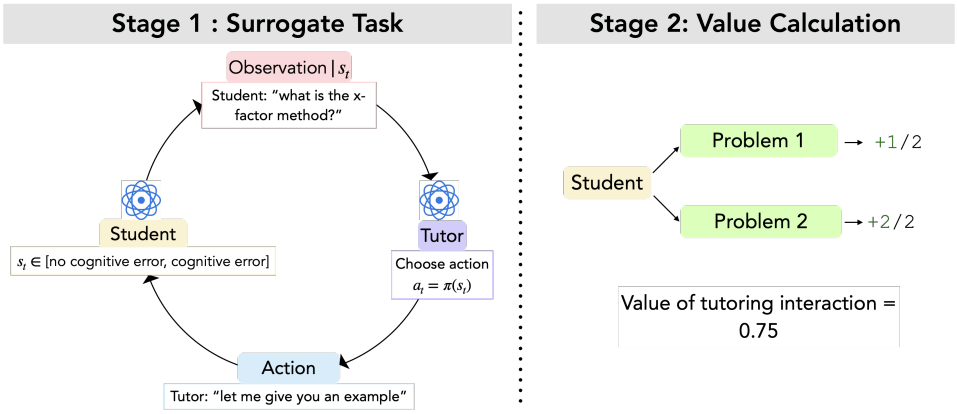

Figure 1: **Pipeline overview of a single surrogate task.** Each task consists of two stages. The goal of the surrogate task is to help a simulated student overcome a cognitive error. In the first stage, the tutor LLM engages in a conversation with a simulated student, represented as an LLM, who starts the interaction with a specific cognitive error. Upon completion of the dialogue, the student answers two targeted assessment questions designed to reveal whether the initial cognitive error persists. The tutor's score on the surrogate task is equivalent to the student's average performance on these post-conversation questions.

4 Methods

Instantiating the POMDP. As introduced in Section 3, we model the online tutoring environment as a POMDP. The state space $\mathcal{S}$ represents the student's latent cognitive state, which we assume to be binary: either in a *cognitive error* state or a *no cognitive error* state. The finite action space $\mathcal{A}$ captures the set of tutor actions, categorized as `[ask a question, provide a hint, correct the student, confirm the student's process]`. These categories align with classifications used in existing tutoring datasets [Miller and Dicerbo, 2024, Stasaski et al., 2020].

The observation space Ω consists of all possible student utterances in natural language. Each observation $o_t \in \Omega$ is a stochastic function of the latent state $s_t \in \mathcal{S}$ at turn t. Since utterances are conditioned on the student's latent state, the observation model $\mathcal{O}(o_t \mid s_t, a_{t-1})$ is unknown and instantiated via LLM prompting with few-shot examples. For example, if a student is in a cognitive error state, the student LLM is instructed to respond in a manner consistent with that error.

Surrogate tasks. We use this POMDP formalism to define surrogate tasks that simulate one-on-one interactions between a tutor and a student exhibiting at least one cognitive error. Each task begins with an initial student question q_0 drawn from a real-world tutoring dataset. All interactions begin with an initial student question q_0 drawn from a tutoring dataset. We then source possible cognitive errors that a student may have when asking q_0 by prompting an LLM. To elicit realistic errors, the prompt asks the LLM to to generate a list of plausible cognitive errors relevant to the student's initial question q_0, along with explanations and supporting literature. For instance, for a question about quadratic equations, the LLM may surface cognitive errors such as incorrectly applying the quadratic formula. A specific cognitive error is then sampled, and the student LLM is prompted to generate utterances that reflect this error. Explicitly conditioning the prompt on the chosen error is sufficient to produce responses consistent with that error.

Simulating tutor–student interactions. Since each task begins with a known cognitive error, we query an LLM to enumerate instructional strategies for addressing it. After each tutor–student exchange, another LLM evaluates whether the tutor applied an effective strategy. If so, the student transitions to the *no cognitive error* state. Thus, the transition function $\mathcal{T}(s_{t+1} \mid s_t, a_t)$ is implicitly defined by an LLM that analyzes the dialogue and determines whether the cognitive error has been resolved. Interactions terminate either when the student reaches the *no cognitive error* state or when the maximum horizon of 20 turns is reached. Student responses are generated via LLM prompting with few-shot examples to mimic the tone and structure of authentic utterances (Figure 2).

```
Student: *nods* Okay I get how to add matrices.
Can we move on to something else with matrices?
```

```
Student:  Ok, I got -12. But can you just show
me exactly how to do the x-factor method
instead of explaining it? It's easier for me to
follow along if I see the steps.
```

```
Student:  Okay, to solve 2x + 5 = 15, I would
first subtract 5 from both sides to get 2x =
10. Then divide both sides by 2 to get x = 5.
```

```
Student:  Yeah I think I get it now. But can
you show me another example of factoring a 4th
degree polynomial? The more examples the better
for me to fully understand.
```

Figure 2: **Examples of student utterances in our POMDP framework.** We generate these student utterances using few-shot examples from the Khan Academy tutoring dataset.

Reward signal. The reward function R is sparse and defined over terminal outcomes. Intermediate utterances receive zero reward. At termination, the student is assigned a reward equal to their average score on a set of practice problems. These problems are generated by prompting an LLM to construct practice questions that can only be solved if a specific cognitive error has been addressed. Problem solutions are automatically graded by an LLM, using chain-of-thought reasoning when necessary to assign partial credit. High performance on these problems indicates that the tutoring interaction was successful. This reward design enables us to compare tutoring policies based on their effectiveness in resolving cognitive errors. Moreover, the POMDP formalism allows us to transform single student–tutor interactions from offline datasets into reusable surrogate tasks, thereby scaling small datasets.

Tutoring policies. To evaluate the surrogate tasks, we design simple policies $\pi : \Omega \to \mathcal{A}$ that map student utterances to tutor actions. For tractability, we define these policies via LLM prompting: given the dialogue history, the LLM selects an appropriate action. Tutor utterances are then generated by prompting with few-shot examples from the dataset corresponding to the chosen action (e.g., providing hints, offering corrections). We note that our proposed methodology is an evaluation framework, and designing effective tutoring policies is out of scope for this work. For example, if a tutoring agent is already available, we bypass policy design and instead use the agent directly within our evaluation framework.

We instantiate two policy types. The *error-informed policy* is supplied with the correct instructional strategy for the student's cognitive error and incorporates this strategy in its utterances. The *uninformed policy* lacks this information and interacts without targeted guidance.

4.1 Evaluating a Tutor Policy

Our goal is to evaluate how effective different policies are across surrogate tasks. For a policy π and a dataset of surrogate tasks D, the value of π is

$$V^{\pi} = \mathbb{E}_{\tau \sim D}\left[\sum_{t=0}^{T} \gamma^t R(s_t, a_t)\right],$$

where $\tau = (s_0, a_0, o_0, \ldots, s_T, a_T, o_{T+1})$ is a trajectory with horizon T, $\gamma \in [0, 1]$ is the discount factor, and rewards are zero for $t < T$. At termination, $R(s_T, a_T)$ corresponds to the practice problem score. T is either the timestep at which the student resolves their error or the maximum horizon of 20 turns. Policies with higher expected return V^{π} are considered more effective, reflecting their ability to help students resolve cognitive errors with minimal intervention.

5 Experiments

Our experiments seek to answer the following questions:

1. Is simulated student performance on generated practice problems correlated with their latent state?
2. Can TutorTest more effectively distinguish between tutoring policies compared to baseline approaches?
3. Does the TutorTest surrogate task framework succeed across multiple educational domains?

Datasets. To answer these research questions, we use two publicly available datasets. The first is the Khan Academy dataset [Miller and Dicerbo, 2024] which contains *188* conversations between real students and the Khanmigo tutoring online tutor. All tutoring conversations in the Khan Academy dataset focus on math, with topics ranging from basic arithmetic to advanced calculus. The second dataset is the CIMA dataset [Stasaski et al., 2020] which includes dialogues between real students and human online tutors focusing on the domain of language learning. All students in the CIMA dataset are native English speakers trying to learn Italian.

Baselines We compare TutorTest to three automatic language evaluation metrics. These metrics are BERTScore [Zhang et al., 2020], which evaluates semantic similarity between generated and reference texts, BLEU [Papineni et al., 2002], which measures n-gram overlap with reference texts, and BLEURT [Sellam et al., 2020] which combines contextual and linguistic similarity features to score generated text. For metrics that require reference texts, we use the original tutoring conversation corresponding to the same initial student question in the appropriate dataset.

In our experiments, we use the Claude Sonnet 3 model [Anthropic, 2024], accessed via the Anthropic Bedrock API.

5.1 Performance on practice problems strongly correlates with student latent state

We first examine the correlation between a simulated student's latent state and their performance on practice problems. Intuitively, a student with one or more cognitive errors should perform worse on problems targeting those errors than a student without such errors. To test this, we evaluate 25 simulated students within each dataset on two practice problems specifically designed for their cognitive error, measuring performance in both of the binary latent states. We find that simulated students perform significantly better when they are not in a cognitive error state (Figure 3). This result indicates that our prompting procedure faithfully captures the impact of cognitive errors on student performance. Moreover, this supports the use of practice problems as a reward signal that aligns with the goal of online tutoring which is to help students overcome their cognitive errors.

5.2 TutorTest better distinguishes teaching policies in comparison to baseline approaches

Given that we have validated our reward signal, we next evaluate the ability for TutorTest to effectively distinguish between tutoring policies. To do this, we first define two tutoring policies: one that is "error informed" and one that is "uninformed". The "error-informed" policy is one that uses the appropriate instructional strategy to address a student's specific cognitive error. In particular, this tutoring policy is aware of the student's cognitive error and the instructional strategies that can be used to address it. In contrast, the "uninformed" policy is unaware of the student's cognitive error, and as a result, does not know how to help the student overcome their error. In practice, the "error informed" policy is more goal oriented than the "uninformed" one, which simply responds to the student's prior utterance.

Now, we evaluate the ability of TutorTest to distinguish between the two policies in comparison to baseline automatic evaluation approaches (Figure 4). We evaluate all methods using the same 25 surrogate tasks for each dataset. Our results indicate that the policy value calculated using the TutorTest framework reliably assigns a higher value to the "error-informed" policy, with statistically significant results across both datasets. In contrast, the baseline automatic evaluation metrics either are unable to distinguish between the two policies consistently across the domains, or assign a higher value to the "uninformed" policy. Over comparisons between a single transcript from an "error-informed" policy and a single transcript from an "uninformed" policy, we find that TutorTest identifies the better policy 97% more frequently than BLEURT, the best performing baseline. As a result, we conclude that our automatic evaluation framework can better distinguish between two tutoring policies than baseline automatic evaluation approaches.

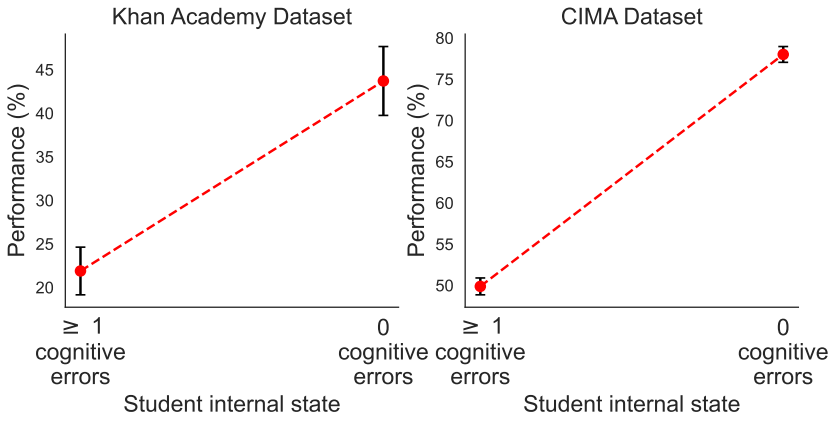

Figure 3: **Performance on the practice problems is highly correlated with the latent student state in both datasets.** Performance is measured according to the custom reward signal, which assigns higher reward to correct answers as measured on the practice problems at the end of a student-tutor interaction. We report average performance across a set of 25 simulated students. Error bars indicate 95% confidence intervals. Each simulated student solves two practice problems, and their performance corresponds to the averaged score across the two problems. Students with no cognitive errors exhibit much higher performance on practice problems than students with one or more cognitive errors in both datasets.

Our results highlight TutorTest's ability to effectively distinguish between two types of tutoring policies. While TutorTest can reliably identify the stronger policy, it also does so far more quickly than human-subject evaluations. The ASU Study Hall human-subjects evaluation discussed in Jurenka et al. [2024] took 15 weeks to run, whereas the full evaluation including the ability to run all surrogate tasks to compare two policies takes less than 25 hours, which can be improved with parallelization. As a result, TutorTest can provide a way to distinguish candidate policies according to the principles outlined in Section 4 at less than 1% of the time-cost of a human-subjects evaluation.

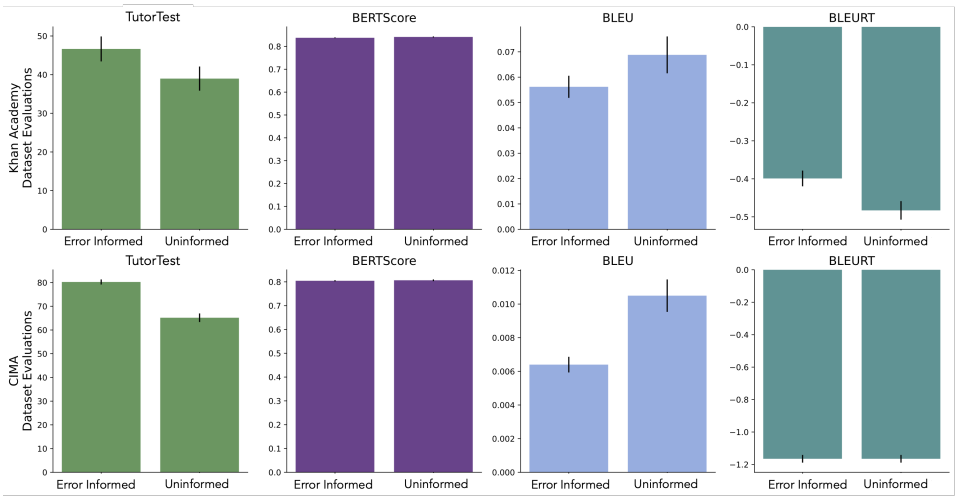

Figure 4: **TutorTest better distinguishes between tutoring policies than baseline evaluation approaches.** The metric for TutorTest is the policy value, $v(\pi_e)$. We report policy values and baseline scores for both the Khan Academy dataset (top row) and CIMA dataset (bottom row). Error bars indicate 95% confidence intervals. Unlike baselines, TutorTest correctly and significantly distinguishes the two policies, assigning higher value to the error-informed policy that more effectively helps students overcome cognitive errors than the uninformed policy.

6 Discussion

Our work advances the automatic evaluation of online tutoring policies through TutorTest, a POMDP-based framework that scales small offline datasets to enable more reliable ranking of tutoring policies under a custom reward signal. TutorTest automatically generates a suite of surrogate tasks which each simulate a conversation between a student with specific cognitive errors and a language model–based tutoring agent. Performance on a given surrogate tasks correlates with improved tutoring outcomes, in which students overcome their initial cognitive erors. Because the surrogate tasks are created entirely via automated prompt engineering, the framework is both inexpensive and fast to run, providing an efficient means to pre-screen tutoring policies before human-subjects evaluation.

Limitations A key assumption in our framework is that the primary goal of online tutoring is to help students overcome cognitive errors. This assumption underpins the design of the reward signal, which prioritizes policies that quickly assist students in resolving their errors. While this aligns with many pedagogical theories, it may not fully capture other dimensions of effective tutoring, such as fostering long-term conceptual understanding. Additionally, our framework does not consider the setting in which a student can develop cognitive errors as a result of interactions with a tutoring agent. Future work should explore integrating additional pedagogical principles, such as those outlined in Jurenka et al. [2024], into the reward structure to create a more holistic evaluation framework. Furthermore, the student state is a binary representation indicating whether they have a cognitive error; a richer representation can facilitate a more realistic understanding of whether policies can improve student outcomes.

Future Work There are several avenues for extending this work. In our current framework, reward is only received at the end of an interaction, future work can consider how to assign per-step reward. Future work can also use different LLM model families for the different roles (e.g., student, tutor or judge). While our proposed framework evaluates existing tutoring policies, a future goal is to facilitate the development of improved tutoring policies. Future efforts could leverage this framework to learn and optimize reinforcement learning-based tutor policies across diverse educational domains. Finally, validating the effectiveness of the evaluation framework through human subjects studies will be essential to determine whether the rankings of policies identified lead to improved student learning outcomes in practice. A preliminary step can be to incorporate human-rubric spot checks within the pipeline to validate reward signals.

References

Anthropic. Claude ai assistant, 2024. Available at `https://www.anthropic.com`.

B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational researcher*, 13(6):4–16, 1984.

T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners, 2020. URL `https://arxiv.org/abs/2005.14165`.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

D. Demszky, J. Liu, H. Hill, D. Jurafsky, and C. Piech. Can automated feedback improve teachers' uptake of student ideas? evidence from a randomized controlled trial in a large-scale online course. *Educational Evaluation and Policy Analysis*, May 2023.

X. Gao, Y. Zhang, M. Galley, C. Brockett, and B. Dolan. Dialogue response ranking training with large-scale human feedback data. In B. Webber, T. Cohn, Y. He, and Y. Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 386–395, Online, Nov. 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.28. URL `https://aclanthology.org/2020.emnlp-main.28/`.

H. Hao, X. Zhang, and A. Zhou. Large language models as surrogate models in evolutionary algorithms: A preliminary study, 2024. URL https://arxiv.org/abs/2406.10675.

J. Hattie and H. Timperley. The power of feedback. *Review of educational research*, 77(1):81–112, 2007.

D. G. Jones and M. R. Endsley. Overcoming representational errors in complex environments. *Human Factors*, 42(3):367–378, 2000.

I. Jurenka, M. Kunesch, K. R. McKee, D. Gillick, S. Zhu, S. Wiltberger, S. M. Phal, K. Hermann, D. Kasenberg, A. Bhoopchand, A. Anand, M. Pîslar, S. Chan, L. Wang, J. She, P. Mahmoudieh, A. Rysbek, W.-J. Ko, A. Huber, B. Wiltshire, G. Elidan, R. Rabin, J. Rubinovitz, A. Pitaru, M. McAllister, J. Wilkowski, D. Choi, R. Engelberg, L. Hackmon, A. Levin, R. Griffin, M. Sears, F. Bar, M. Mesar, M. Jabbour, A. Chaudhry, J. Cohan, S. Thiagarajan, N. Levine, B. Brown, D. Gorur, S. Grant, R. Hashimshoni, L. Weidinger, J. Hu, D. Chen, K. Dolecki, C. Akbulut, M. Bileschi, L. Culp, W.-X. Dong, N. Marchal, K. V. Deman, H. B. Misra, M. Duah, M. Ambar, A. Caciularu, S. Lefdal, C. Summerfield, J. An, P.-A. Kamienny, A. Mohdi, T. Strinopoulous, A. Hale, W. Anderson, L. C. Cobo, N. Efron, M. Ananda, S. Mohamed, M. Heymans, Z. Ghahramani, Y. Matias, B. Gomes, and L. Ibrahim. Towards responsible development of generative ai for education: An evaluation-driven approach, 2024. URL https://arxiv.org/abs/2407.12687.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1–2):99–134, May 1998. ISSN 0004-3702.

R. Louie, A. Nandi, W. Fang, C. Chang, E. Brunskill, and D. Yang. Roleplay-doh: Enabling domain-experts to create llm-simulated patients via eliciting and adhering to principles, 2024. URL https://arxiv.org/abs/2407.00870.

J. M. Markel, S. G. Opferman, J. A. Landay, and C. Piech. Gpteach: Interactive ta training with gpt-based students. In *Proceedings of the Tenth ACM Conference on Learning @ Scale*, L@S '23, page 226–236, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400700255. doi: 10.1145/3573051.3593393. URL https://doi.org/10.1145/3573051.3593393.

S. A. Mathan and K. R. Koedinger. Fostering the intelligent novice: Learning from errors with metacognitive tutoring. In *Computers as Metacognitive Tools for Enhancing Learning*, pages 257–265. Routledge, 2018.

K. K. Maurya, K. A. Srivatsa, K. Petukhova, and E. Kochmar. Unifying ai tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of llm-powered ai tutors, 2025. URL https://arxiv.org/abs/2412.09416.

P. Miller and K. Dicerbo. Tutoring accuracy dataset, 2024. URL https://github.com/Khan/tutoring-accuracy-dataset.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

J. S. Park, L. Popowski, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Social simulacra: Creating populated prototypes for social computing systems, 2022. URL https://arxiv.org/abs/2208.04024.

J. S. Park, J. C. O'Brien, C. J. Cai, M. R. Morris, P. Liang, and M. S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023. URL https://arxiv.org/abs/2304.03442.

A. Scarlatos, N. Liu, J. Lee, R. Baraniuk, and A. Lan. *Training LLM-Based Tutors to Improve Student Learning Outcomes in Dialogues*, page 251–266. Springer Nature Switzerland, 2025. ISBN 9783031984143. doi: 10.1007/978-3-031-98414-3_18. URL http://dx.doi.org/10.1007/978-3-031-98414-3_18.

T. Sellam, D. Das, and A. P. Parikh. Bleurt: Learning robust metrics for text generation, 2020. URL https://arxiv.org/abs/2004.04696.

K. Stasaski, K. Kao, and M. A. Hearst. CIMA: A large open access dialogue dataset for tutoring. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 52–64, Seattle, WA, USA â†' Online, July 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.bea-1.5.

A. Tack, E. Kochmar, Z. Yuan, S. Bibauw, and C. Piech. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In E. Kochmar, J. Burstein, A. Horbach, R. Laarmann-Quante, N. Madnani, A. Tack, V. Yaneva, Z. Yuan, and T. Zesch, editors, *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, pages 785–795, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.bea-1.64. URL https://aclanthology.org/2023.bea-1.64.

L. Team, A. Modi, A. S. Veerubhotla, A. Rysbek, A. Huber, B. Wiltshire, B. Veprek, D. Gillick, D. Kasenberg, D. Ahmed, I. Jurenka, J. Cohan, J. She, J. Wilkowski, K. Alarakyia, K. R. McKee, L. Wang, M. Kunesch, M. Schaekermann, M. Pîslar, N. Joshi, P. Mahmoudieh, P. Jhun, S. Wiltberger, S. Mohamed, S. Agarwal, S. M. Phal, S. J. Lee, T. Strinopoulos, W.-J. Ko, A. Wang, A. Anand, A. Bhoopchand, D. Wild, D. Pandya, F. Bar, G. Graham, H. Winnemoeller, M. Nagda, P. Kolhar, R. Schneider, S. Zhu, S. Chan, S. Yadlowsky, V. Sounderajah, and Y. Assael. Learnlm: Improving gemini for learning, 2024. URL https://arxiv.org/abs/2412.16429.

R. E. Wang and D. Demszky. Is chatgpt a good teacher coach? measuring zero-shot performance for scoring and providing actionable insights on classroom instruction, 2023. URL https://arxiv.org/abs/2306.03090.

R. E. Wang, Q. Zhang, C. Robinson, S. Loeb, and D. Demszky. Bridging the novice-expert gap via models of decision-making: A case study on remediating math mistakes, 2024a. URL https://arxiv.org/abs/2310.10648.

R. E. Wang, A. T. Ribeiro, C. D. Robinson, S. Loeb, and D. Demszky. Tutor copilot: A human-ai approach for scaling real-time expertise, 2025. URL https://arxiv.org/abs/2410.03017.

S. Wang, T. Xu, H. Li, C. Zhang, J. Liang, J. Tang, P. S. Yu, and Q. Wen. Large language models for education: A survey and outlook, 2024b. URL https://arxiv.org/abs/2403.18105.

J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou. Chain-of-thought prompting elicits reasoning in large language models, 2023. URL https://arxiv.org/abs/2201.11903.

T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi. Bertscore: Evaluating text generation with bert, 2020. URL https://arxiv.org/abs/1904.09675.

A Code Availability

The implementation is publicly available on GitHub (`https://github.com/aishwarya-rm/TutorTest`). All of our experiments were run on an internally-hosted cluster using a single GPU. Our experiments in total do not exceed 100 hours of time.

B Datasets

In this work, we use two publicly available tutoring datasets to develop surrogate tasks. These datasets contain conversations between students and tutors across two domains: math, and language learning.

CoMTA Dataset [Miller and Dicerbo, 2024]: The CoMTA dataset, released from Khan Academy, contains 188 conversations between real human students and Khanmigo, a language-model based tutor from Khan Academy. The conversations range in mathematical concepts, including Elementary mathematics, Algebra, Trigonometry, and Geometry. In our work, we subset this dataset to include all conversations that focus on the Algebra concepts.

CIMA Dataset [Stasaski et al., 2020]: The CIMA dataset consists of conversations between crowd-workers role-playing as students and tutors focusing on learning the Italian language. Each exercise in this dataset contains an image and concept; the student's goal is to describe the image. In our work, we subset this dataset into the Prepositional Phrase component, in which a student is tasked with describing how an object is placed in relation to another object. For example, suppose the image contains a cat on top of a purple bed. The student's goal is to complete the sentence starting with "il gatto ", to describe the cat's position in relation to the bed.

C Prompts

Here we report the prompts we use to query LLMs for different purposes. We generate possible cognitive errors that are conditional on a problem queried from the student and supported by relevant literature using the following prompt.

Listing 1: Generating cognitive errors.

```
We are designing a task when an AI tutor is helping a student learn [
    educational domain]. The student is currently working on the
    following problem. [Problem]: {task} Can you provide a list of 2
    different cognitive errors that the student could have where the
    tutor would need to change their teaching content for different
    mistakes? Reference the appropriate literature that cites these
    cognitive errors as relevant to the problem domain. Begin your
    generation with 'List:'.
```

To source possible strategies to help a student in overcoming a given cognitive error, we use the following prompt, which requires a specific cognitive error and the problem the student is working on as input.

Listing 2: Generating strategies to overcome cognitive errors.

```
We are designing a task in which an AI tutor is helping a student
    learn [educational domain]. The student is currently working on
    the following problem. [Problem]"
The student may have cognitive errors that require the tutor to
    personalize their teaching content. Here is one example of a
    cognitive error that a student may have: [Cognitive Error]. If a
    student had this cognitive error, can you identify 2 ways that an
    online tutor could help the student overcome this cognitive error?
     This can be a suggestion about a way that the online tutor can
    change their teaching content or key information that should be
    communicated to the student. Make your suggestions very specific
    to the cognitive error from earlier and the problem that the
    student is working on. Support your suggestions with relevant
    literature, ensuring that the literature sources are valid. Recall
```

```
    that the setting is online tutoring, which means that visual cues
    are difficult to employ. Begin your generation with ‘List:‘"
```

To replicate a student behaving as if they have a particular cognitive error, we prepend all prompts to the student agent with the following phrase.

Listing 3: Generating observations from a student with a given cognitive error

```
Since you are still learning, you have a cognitive error that causes
    you to make mistakes and ask questions as you are learning. You
    have the following cognitive error: [cognitive error]. This
    cognitive error affects how you respond to practice problems and
    your general understanding of [educational domain]. Respond to the
     tutor's last utterance while conditioning on having this
    cognitive error. Keep in mind that you should maintain this
    cognitive error throughout your interactions with the tutor. For
    example, be consistent across responses as you continue this
    conversation. You might also be slow to pick up new material. Do
    not include details about this cognitive error in the response.
```

Once the student overcomes, their cognitive error, we prepend all prompts to the student agent with the following phrase.

Listing 4: Generating observations from a student without a cognitive error

```
Since you are still learning, you had a cognitive error at the
    beginning of your interaction with the tutor. This error was: [
    cognitive error]. You have overcome this cognitive error through
    learning, and you no longer exhibit any symptoms of this cognitive
     error.
```

To assess whether the student has overcome their cognitive error, we ask an LLM to analyze the tutoring interaction so far, identifying whether the tutor has used an appropriate strategy to help the student overcome their cognitive error. We use the following prompt.

Listing 5: Transition dynamics for student state

```
We are evaluating a conversation between an online tutor and a student
     learning Italian. In the dialogue below, the the tutor's are
    prefaced by ‘‘Tutor:’’ and the student\'s utterances are prefaced
    by ‘‘Student:’’. [Dialogue]. The student currently has the
    following cognitive error: [Cognitive error]. According to experts
    , to help the student overcome this error, the tutor must use at
    least one of the following strategies: [Strategies].  Based on the
     dialogue, has the tutor employed any of these strategies to help
    the student overcome their cognitive error? Respond with a <<yes>>
     or <<no>> depending on whether or not the tutor has used the
    strategies provided, and provide a justification.
```

Throughout the conversation, we direct the responses to the student by first querying which is the most appropriate action, using the following prompt.

Listing 6: Determining the best tutoring action.

```
You are an online tutor working with student that is learning [
    educational domain]. In the dialogue below, the your utterances
    are prefaced by ‘‘Tutor:" and the student\'s utterances are
    prefaced by ‘‘Student:". [Dialogue] Based on the student\'s last
    response, what is the best action to take? Your options are: 1.
    ask a question either 2. give the student a hint, 3. correct a
    prior step of the problem derivation, and 4. encourage the student
     and confirm their process.
```

Once the tutor's action is decided, the tutor's response is generated using the following prompt.

Listing 7: Generating a response from the tutoring agent.

```
You are an online tutor working with a student that is learning [
    educational domain]. In the dialogue below, the your utterances
    are prefaced by ‘‘Tutor:" and the student\’s utterances are
    prefaced by ‘‘Student:". [Dialogue] Generate a response to the
    student\’s last utterance in the role of an online tutor.
    Specifically, [perform prescribed action]. Here are examples of
    how an online tutor will interact with a student to [perform
    prescribed action]. [Few-shot examples from tutoring dataset].
    Keep your response brief and do not reveal the answer to the
    problem that the student is trying to work on.
```

Finally, to assess the tutoring conversation, we evaluate the student's ability to solve two practice problems. The practice problems are specific to the student's cognitive error and are generated using the following prompt.

```
You are an online tutor helping a student with [educational domain].
    The following is the beginning of the student’s "
conversation with you: [student initial query]. This student could
    possibly have the following cognitive errors: [cognitive errors].
    Can you provide a list of 2 different practice problems that a
    student should be able to solve if they do not have any of the
    specified cognitive errors? Do not include the solution in these
    practice problems.
```

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We introduce contributions in the end of the introduction and support these contributions with our methods and empirical results sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We include limitations in the discussion section of the paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: We do not include theoretical proofs.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include prompt examples in our appendix, and release anonymized documented code for the submission.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We include an anonymized codebase in our submission.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We discuss the experimental setting in the methods section and appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We report 95% confidence intervals in our figures.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We discuss computational requirements in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This paper does conform to the NeurIPS code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: We believe that this work poses no negative potential societal impacts. The goal of our work is to enable the scaling of offline datasets that allow us to effectively evaluate tutoring policies to enable a wide range of users to have effective tutoring tools.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We credit the creators of each dataset used in the work.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We include discussion about the LLM usage in our methods sections.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.