Interactive and Scalable Evaluation of LLM Math Tutoring via Simulated Teacher-Student Dialogues

Gihyeon Choi and Harksoo Kim*
Konkuk University
Seoul, Republic of Korea
{pluto32,nlpdrkim}@konkuk.ac.kr

Abstract

Evaluating LLMs as math tutors is challenging because many automated protocols either score a single response against a reference or place the model in a fixed dialogue context, without measuring its ability to *build and steer* tutoring interactions. We present a fully automated, simulation-based framework that generates controlled multi-turn teacher–student dialogues with turn-level answer verification under a fixed interaction budget. This budget specifies the maximum number of student–teacher rounds allowed per problem ($K = 5$ in our main experiments). To prevent score inflation from answer-revealing feedback, we introduce an *overinformative* eligibility filter: a hint-only student test applied to the teacher utterance immediately preceding the student's first correct attempt. Only eligible corrections receive credit in our effectiveness and efficiency metrics, both of which are normalized by the number of initially incorrect sessions. We also propose a diagnostic metric suite separating tutoring effectiveness/efficiency (ECR/FEI), answer reachability (ARG), and teacher judgment (correctness/incorrectness F1). Experiments on GSM8K and MATH with diverse teacher and student models show that problem-solving accuracy is an informative but incomplete proxy for interactive tutoring ability, and that filtering overinformative feedback can substantially alter apparent performance. Our framework enables scalable, reproducible, and fine-grained evaluation of LLM math tutoring beyond outcome-only scoring.

1 Introduction

Large language models (LLMs) are increasingly deployed as interactive math tutors (Dinucu-Jianu et al., 2025; Team et al., 2025b), raising a practical question: *how can we evaluate LLM math tutor models in a way that more faithfully reflects their underlying tutoring capability, while remaining reliable and scalable (Gupta et al., 2025)?* Prior work often mixes automatic scoring with human review by comparing tutor utterances to reference responses in existing teacher–student dialogues and asking raters to judge educational appropriateness (Macina et al., 2023; Maurya et al., 2025; Macina et al., 2025). However, human-in-the-loop evaluation is costly and difficult to standardize: judgments vary across raters and time, and results are hard to reproduce.

Fully automated evaluation is appealing, but common protocols have two structural limitations (Mohammadi et al., 2025): (i) scoring against a single reference response and (ii) evaluating within a pre-written dialogue context. Single-reference comparisons can penalize valid tutoring that differs in wording, explanation order, or teaching style (e.g., guiding questions vs. direct hints), making scores sensitive to surface form rather than instructional quality (Zhang et al., 2025; Gudipati, 2025). Pre-written contexts also make evaluation overly clean: real tutoring requires interpreting imperfect student answers, choosing the next move, handling misunderstandings and repetition, and recovering when feedback fails. As a result, such setups often measure how well a model *continues someone else's dialogue* rather than how well it can *build and steer* an interaction, and scores may depend on the dataset's dialogue flow or revealed solution steps (Daheim et al., 2024; Wang et al., 2025). Moreover, turn-level scoring is only weakly tied to what tutoring ultimately cares about: whether the student is corrected and how efficiently they reach the correct answer.

We propose a simulation-based, fully automated framework for evaluating LLM math tutoring through interactive teacher–student dialogues on verifiable math problems.[1] A tutor model plays

*Corresponding author.

[1]Code is available at https://github.com/KUNLP/

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 3300–3313
November 6-10, 2026 ©2026 Association for Computational Linguistics

the teacher role and interacts with a fixed student model under a controlled multi-turn protocol, while student correctness is automatically verified at each turn. Our goal is to provide a controlled and reproducible protocol for evaluating interaction-level tutoring behavior, rather than to replace studies with real learners or directly measure long-term learning gains. By fixing the student model and experimental conditions—including the problem set, prompts, interaction budget, decoding settings, verification rules, and stopping conditions—and varying only the teacher model, we enable standardized teacher comparisons. We treat each problem as a multi-turn tutoring session and evaluate whether an initially incorrect simulated student reaches the correct answer within the fixed budget and how efficiently this occurs. We also apply a conservative overinformative eligibility rule to reduce credit for answer-revealing shortcut success, rather than treating every correction as evidence of effective guidance. These interaction-level behaviors are difficult to capture through single-response scoring or evaluation within fixed, pre-written dialogue contexts.

Our contributions can be summarized as follows:

- **Interaction-aware, fully automated tutoring evaluation** We introduce a simulation-based protocol that evaluates math tutoring via controlled multi-turn teacher–student dialogues with turn-level answer verification under a fixed interaction budget.
- **Overinformative-aware eligibility to prevent shortcut success** We propose a hint-only overinformative test on the teacher utterance immediately preceding the student's first correct attempt and assign credit only to eligible corrections when measuring tutoring effectiveness and efficiency.
- **Complementary diagnostics and empirical analyses** We quantify interaction-level tutoring behavior through Error Correction Rate (ECR), Feedback Efficiency Index (FEI), Answer Reachability Gain (ARG), and teacher-judgment F1 scores (CJ-F1/IJ-F1). Experiments on GSM8K and MATH quantify the relationship between standalone problem-solving accuracy and these tutoring metrics, showing that accuracy is an informative but incomplete proxy for interactive tutoring ability. We also assess agreement between automatic overinformativeness and teacher-judgment labels and human annotations.

LLM-MATH-TUTOR-EVAL.

2 Related Work

LLM teachers for dialogue-based math tutoring Dialogue tutoring aims to scaffold learning through structured guidance rather than merely provide correct answers. Recent work shapes LLM teacher behavior via prompting, finetuning, and post-training alignment. Prompt-based methods range from handcrafted pedagogical prompts (Sonkar et al., 2023; Kargupta et al., 2024) to more structured, decision-oriented prompting that mirrors teacher practices (e.g., diagnosing error types before selecting pedagogical intent) (Wang et al., 2024). However, prompts can be costly to design and brittle, complicating systematic comparison (Jurenka et al., 2025). Finetuning trains teachers on pedagogical conversations or synthetic teacher–student dialogues (Xu et al., 2024), but collecting high-quality human-created tutoring data remains expensive (Macina et al., 2023), motivating mixtures of synthetic and human data or alignment to implicit pedagogical preferences (Team et al., 2025b). Across these approaches, a recurring bottleneck is the lack of scalable, reproducible protocols for comparing interaction-level tutoring ability.

Automatic and human evaluation of tutoring dialogue Tutoring quality is inherently open-ended: there is rarely a single best teacher utterance per turn, so reference matching (or generic NLG metrics) can be noisy and unreliable (Papineni et al., 2002; Zhang et al., 2020; Jurenka et al., 2025). Task-specific classifiers (e.g., active listening) cover only narrow facets and often ignore long-range context (Demszky et al., 2021). Many recent evaluations therefore rely on human experts or strong LLM judges (e.g., GPT-4) along dimensions such as helpfulness, presentation, or Socratic behavior (Liu et al., 2024; Liang et al., 2024), but these are costly, hard to reproduce, and may depend on closed models. Taxonomies and guidelines provide useful structure but are typically static rather than an automated protocol for continual comparison (Tack and Piech, 2022; Maurya et al., 2025); learner studies directly measure learning outcomes but are expensive and practically constrained (Schmucker et al., 2024). Our work addresses this gap with a unified automatic evaluation: we evaluate teacher models via controlled

LLM–LLM interactions and report session-level outcomes and judgment behavior, while explicitly controlling for answer-revealing (overinformative) feedback to prevent shortcut "success" from inflating tutoring-driven metrics.

3 Methodology

3.1 Dialogue Data Construction

We build an evaluation dialogue dataset by simulating math tutoring between a teacher LLM and a student LLM. Each benchmark problem (e.g., GSM8K, MATH) yields one self-contained teacher–student session. Both agents follow fixed role instructions: the student attempts the problem with explicit reasoning, revises only the erroneous part when corrected, and reports the final answer as "So, the answer is ..."; the teacher assesses each attempt and, when the student is incorrect, provides concise guidance (hints or guiding questions) rather than directly giving the final answer or a full solution. The full problem, student, teacher, and hint prompt templates are provided in Appendix B.

Each session begins with a fixed problem prompt. At round k, the student generates an attempt s_k, which is automatically verified to obtain y_k, and the teacher then responds with t_k. The teacher responds even when s_k is correct, allowing us to evaluate correctness judgment. The full simulation protocol is provided in Algorithm 2 (Appendix A).

The interaction budget K denotes the maximum number of student–teacher rounds allowed per problem, where each round consists of one student attempt followed by one teacher response. If the student first becomes correct at round k, the session ends after the teacher response t_k; otherwise, it ends after t_K.

We use $K = 5$ in the main experiments as a controlled evaluation condition, rather than an optimal or required value. More generally, K is a configurable parameter of the framework. For each student model, we fix the problem set, prompts, interaction budget, decoding settings, verification rules, and stopping conditions across runs and vary only the teacher model, so that all teachers are compared under the same tutoring opportunity.

3.2 Evaluation Setup and Notation

We evaluate tutoring from four complementary aspects: (i) effectiveness (whether interaction helps an initially incorrect student reach a correct answer), (ii) efficiency (eligible correction success weighted by the reciprocal number of feedback steps), (iii) answer reachability (how the simulated student's conditional likelihood of the correct final answer changes following teacher feedback), and (iv) teacher judgment (whether the teacher recognizes correct answers and flags incorrect ones, reported as two F1 scores).

Each problem instance q_i yields one dialogue session D_i generated by Algorithm 2. We denote the student's k-th attempt and the teacher's reply by $s_{i,k}$ and $t_{i,k}$, respectively, for $k = 1, \ldots, K$ (where $K = 5$ is the maximum number of rounds in our main experiments). For each student response, we define a binary correctness signal $y_{i,k} \in \{0, 1\}$ by $y_{i,k} = \textsc{Verify}(s_{i,k}, a_i)$, where VERIFY uses the Hugging Face `math_verify` library (Kydlíček, 2025) to extract the final answer and check equivalence to the ground truth.

3.3 Eligibility and Overinformative Filtering

Our effectiveness/efficiency metrics are intended to capture tutoring-driven correction while reducing credit for answer-revealing shortcuts. However, these metrics can be spuriously inflated when the teacher reveals the answer (or sufficiently decisive information), allowing the student to become correct without meaningful reasoning. We therefore introduce an *overinformative* filter as an *eligibility* rule for computing tutoring-driven metrics.

Rather than measuring global verbosity, we evaluate a single teacher utterance per successful session: the message t_{i,k^*-1} immediately preceding the student's *first* correct attempt s_{i,k^*}. Our operational test asks whether the same student model can produce the correct answer from the original problem and the final pre-success teacher utterance, without the preceding dialogue history. We use only the teacher utterance immediately preceding the student's first correct attempt as a minimal attribution window. This design targets the feedback most directly associated with the transition from failure to success, while avoiding dilution of responsibility across earlier turns whose contribution is harder to isolate.

Algorithm 1 applies a context-stripped hint-only test to successful sessions. For the final pre-success teacher utterance $h = t_{i,k^*-1}$, we re-query the same student model with the original problem and h only, without the preceding dialogue history, and pre-fill the output prefix "So, the answer is". If the student still produces the correct answer, the session is flagged as *overinformative*. Thus, a second-

round correction ($y_{i,2} = 1$) is not excluded automatically; it is flagged only if the first teacher response, when paired with the original problem but without the preceding student attempt or dialogue history, is sufficient for the same student model to produce the correct answer. We use this filter as a conservative eligibility rule for reducing shortcut success, not as a complete judgment of pedagogical quality. The exact HINTPROMPT template is provided in Appendix B.

Among initially incorrect sessions, overinformative corrections receive zero credit in both ECR and FEI but remain in their denominators. This prevents corrections flagged as overinformative from increasing either metric. We additionally report the Overinformative Rate (Overinfo.), defined as the fraction of successful sessions whose final pre-success teacher utterance is flagged as overinformative.

Algorithm 1 Overinformative Filtering (Hint-Only Student Test)

Require: Dialogue D_i with turns $\{(s_{i,k}, t_{i,k})\}_{k=1}^{K}$, problem q_i, answer a_i, student model $\mathcal{S}$

$k^* \leftarrow \min\{k \le K \mid y_{i,k} = 1\}$ {first correct student attempt}
if k^* is undefined **then**
  return N/A
end if
if $k^* = 1$ **then**
  return NOTOVERINFORMATIVE
end if
$h \leftarrow t_{i,k^*-1}$ {final pre-success teacher utterance}
$\hat{s} \leftarrow \mathcal{S}(\text{HINTPROMPT}(q_i, h))$ {answer-only; prefix `So, the answer is` pre-filled}
if $\text{VERIFY}(\hat{s}, a_i) = 1$ **then**
  return OVERINFORMATIVE
else
  return NOTOVERINFORMATIVE
end if

3.4 Metrics

We define metrics for the four evaluation aspects introduced in Section 3.2.

Overinformative handling Overinformative handling follows Section 3.3: only eligible corrections receive credit in ECR and FEI, and both metrics are normalized by the number of initially incorrect sessions. Teacher-judgment metrics are computed on all turns. For ARG, if the final pre-success teacher turn is labeled overinformative, we set the post-feedback reachability equal to the pre-feedback reachability for that turn, i.e., the gain Δ is set to zero.

3.4.1 Tutoring Effectiveness

We measure *tutoring effectiveness* by whether teacher interaction helps an initially incorrect student reach a correct answer within K rounds. Restricting to sessions with $y_{i,1} = 0$, we define

$$\text{SUCCESS}_i = \mathbf{1}\{\exists\, k \le K : y_{i,k} = 1\}.$$

For successful sessions, let $k_i^* = \min\{k \le K \mid y_{i,k} = 1\}$ denote the first correct round, and define

$$\text{OVERINF}_i = \mathbf{1}\{t_{i,k_i^*-1} \text{ is labeled } \textit{overinformative}\},$$

using the hint-only test in Section 3.3. A session is eligible iff

$$\text{ELIGIBLE}_i = \text{SUCCESS}_i \cdot (1 - \text{OVERINF}_i).$$

We then define the filtered *Error Correction Rate* as

$$\text{ECR} = \frac{\sum_i \mathbf{1}\{y_{i,1} = 0\}\ \text{ELIGIBLE}_i}{\sum_i \mathbf{1}\{y_{i,1} = 0\}}.$$

Higher ECR indicates more reliable tutoring-driven correction under the fixed interaction budget.

3.4.2 Tutoring Efficiency

We measure tutoring efficiency using reciprocal weighting inspired by mean reciprocal rank (MRR) (Voorhees and Tice, 2000), giving greater credit to eligible corrections achieved with fewer feedback steps. Let

$$\mathcal{I}_0 = \{i \mid y_{i,1} = 0\}$$

denote all initially incorrect sessions, and let

$$\mathcal{I}_{\text{succ}} = \{i \in \mathcal{I}_0 \mid \text{ELIGIBLE}_i = 1\}$$

denote the eligible successful subset. Using the first-correct round k_i^* defined above, the number of teacher feedback steps preceding the first correct student attempt is $k_i^* - 1$, since the initial attempt is made before any teacher feedback. For $|\mathcal{I}_0| > 0$, we define the Feedback Efficiency Index (FEI) as

$$\text{FEI} = \frac{1}{|\mathcal{I}_0|} \sum_{i \in \mathcal{I}_{\text{succ}}} \frac{1}{k_i^* - 1}.$$

Unsuccessful sessions and overinformative corrections receive zero credit but remain in the denominator; initially correct sessions are excluded. If no eligible correction occurs, FEI is zero provided that $|\mathcal{I}_0| > 0$. Higher FEI reflects both eligible correction success and speed, rather than correction speed conditional on success alone.

3.4.3 Answer Reachability

We use *Answer Reachability Gain* (ARG) as an answer-reachability diagnostic, tracking changes in the simulated student's conditional log-likelihood of producing the ground-truth final answer under the dialogue context following teacher feedback. ARG is not intended as a complete measure of pedagogical feedback quality and does not replace human judgments of pedagogical quality or external validation on tutoring datasets.

The per-turn gain provides a local feedback-level signal, which we aggregate over teacher turns preceding the first correct student attempt. In contrast, ECR measures eligible correction frequency, while FEI additionally weights each eligible correction by the reciprocal number of feedback steps. ARG provides a complementary diagnostic and need not vary monotonically with either ECR or FEI. We compute ARG only over sessions that are initially incorrect and subsequently become correct within the interaction budget, excluding directly correct sessions ($y_{i,1} = 1$).

Let a_i be the ground-truth answer for problem q_i, and let $A_i = (a_{i,1}, \ldots, a_{i,|A_i|})$ denote its tokenized form. Let P be the fixed student output prefix `"So, the answer is"`. For each teacher turn k, let $H_{i,k}$ denote the dialogue history up to and including $t_{i,k}$, with $H_{i,0} = \textsc{ProblemPrompt}(q_i)$. Using the same student model $\mathcal{S}$, we append P to $H_{i,k}$ and compute the log-likelihood of only the ground-truth answer continuation A_i:

$$\begin{aligned}\ell_{i,k} &= \log p_{\mathcal{S}}(A_i \mid H_{i,k}, P) \\ &= \sum_{j=1}^{|A_i|} \log p_{\mathcal{S}}\big(a_{i,j} \mid H_{i,k}, P, a_{i,<j}\big).\end{aligned} \tag{1}$$

We define the per-turn gain as

$$\Delta_{i,k} = \ell_{i,k} - \ell_{i,k-1}. \tag{2}$$

Let $k_i^* = \min\{k \le K \mid y_{i,k} = 1\}$ denote the first correct round. We then define session-level ARG by aggregating gains over teacher turns before the first correct student attempt:

$$\mathrm{ARG}_i = \sum_{k=1}^{k_i^*-1} \Delta_{i,k}. \tag{3}$$

If the final pre-success teacher utterance t_{i,k_i^*-1} is labeled overinformative by the detector in Section 3.3, we set $\Delta_{i,k_i^*-1} = 0$.

Finally, we report the mean session-level ARG over the error-corrected population:

$$\begin{aligned}\mathrm{ARG} &= \frac{1}{|D_{\mathrm{ec}}|} \sum_{i \in D_{\mathrm{ec}}} \mathrm{ARG}_i, \\ D_{\mathrm{ec}} &= \{\, i : y_{i,1} = 0 \wedge \exists\, k \le K,\ y_{i,k} = 1 \,\}.\end{aligned} \tag{4}$$

3.4.4 Teacher Judgment Performance

We evaluate teacher judgment by whether the teacher endorses the student's latest attempt as correct, and withholds endorsement when the attempt is incorrect. For each teacher turn $t_{i,k}$ responding to $s_{i,k}$, we predict a binary label $\hat{j}_{i,k} \in \{\mathrm{YES}, \mathrm{NO}\}$, where YES indicates that the teacher expresses belief that $s_{i,k}$ is correct.

We obtain $\hat{j}_{i,k}$ using an LLM-based checker (gemma-3-27b-instruct). The checker uses dialogue history for reference resolution but bases its decision only on the teacher utterance(s), prioritizing the most recent explicit judgment or, otherwise, the strongest implied one. Conservatively, praise followed by correction, requests to verify, or hedging (e.g., "but ...", "check", "are you sure") is labeled NO. The checker outputs a JSON label and 1–2 short evidence spans.

Using the student's verified correctness $y_{i,k} \in \{0, 1\}$ on $s_{i,k}$ as ground truth, we report two F1 scores: (i) correct-answer recognition F1 (positive class: $y_{i,k} = 1$, predicted positive: $\hat{j}_{i,k} = \mathrm{YES}$), and (ii) incorrect-answer detection F1 (positive class: $y_{i,k} = 0$, predicted positive: $\hat{j}_{i,k} = \mathrm{NO}$). These metrics are computed on all teacher turns, since they measure recognition behavior rather than tutoring-driven success.

3.5 Aggregation and Reporting

Because teacher performance can vary with the student model, we use two-stage aggregation. Let $\mathbb{S}$ and $\mathbb{T}$ denote the sets of student and teacher models. For each pairing $(\mathcal{T}, \mathcal{S}) \in \mathbb{T} \times \mathbb{S}$, we compute each metric over the sessions/turns relevant to that metric, yielding $\mathrm{Score}(\mathcal{T}, \mathcal{S})$.

For metrics directly comparable across students (ECR, FEI, and teacher-judgment F1), we report the teacher-level score by macro-averaging across students:

$$\text{Score}(\mathcal{T}) = \frac{1}{|\mathbb{S}|} \sum_{\mathcal{S} \in \mathbb{S}} \text{Score}(\mathcal{T}, \mathcal{S}).$$

ARG is log-likelihood-based and therefore not directly comparable across student models. Let $\bar{x}_{\mathcal{T},\mathcal{S}}$ denote teacher $\mathcal{T}$'s mean session-level ARG with student $\mathcal{S}$ over error-corrected sessions. For each student $\mathcal{S}$, we define the pairwise win-rate against other teachers as

$$w_{\mathcal{S}}(\mathcal{T}, \mathcal{T}') = \begin{cases} 1, & \bar{x}_{\mathcal{T},\mathcal{S}} > \bar{x}_{\mathcal{T}',\mathcal{S}}, \\ 0.5, & \bar{x}_{\mathcal{T},\mathcal{S}} = \bar{x}_{\mathcal{T}',\mathcal{S}}, \\ 0, & \bar{x}_{\mathcal{T},\mathcal{S}} < \bar{x}_{\mathcal{T}',\mathcal{S}}. \end{cases} \quad (5)$$

We then average these pairwise scores within each student:

$$\text{WinRate}_{\mathcal{S}}(\mathcal{T}) = \frac{1}{|\mathbb{T}| - 1} \sum_{\mathcal{T}' \in \mathbb{T},\, \mathcal{T}' \neq \mathcal{T}} w_{\mathcal{S}}(\mathcal{T}, \mathcal{T}'). \quad (6)$$

We then report the teacher-level ARG score as

$$\text{ARGScore}(\mathcal{T}) = \frac{1}{|\mathbb{S}|} \sum_{\mathcal{S} \in \mathbb{S}} \text{WinRate}_{\mathcal{S}}(\mathcal{T}),$$

where $\text{ARGScore}(\mathcal{T}) \in [0, 1]$ and 0.5 indicates an average tie under the within-student pairwise comparison. In summary, we macro-average raw metric values across student models for ECR, FEI, and teacher-judgment F1, whereas for ARG we macro-average a within-student relative win-rate to avoid artifacts from student-specific likelihood scaling.

4 Experiments

4.1 Experimental Setup

We evaluate teacher models on two mathematical reasoning benchmarks, GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021). For GSM8K, we use the test split. For MATH, we follow Lightman et al. (2024) and use its fixed public 500-problem subset of the test split.

We consider eight teacher models: seven open-weight models—Gemma-3-Instruct (12B, 27B) (Team et al., 2025a), Qwen2.5-Instruct (7B, 14B) (Yang et al., 2025), Phi-3-Medium-128K-Instruct (Abdin et al., 2024a), Phi-4 (Abdin et al., 2024b), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024)—and one proprietary model, GPT-4.1 (Fachada et al., 2025). We select these models to cover multiple model families and parameter scales, including both open-weight and proprietary systems.

We use six student models: Llama-3.2-Instruct (1B, 3B), Qwen2.5-Instruct (1.5B, 3B), and Gemma-3-Instruct (1B, 4B). These students span three model families and multiple model scales, capturing variation in model family and capacity rather than explicitly controlled learner personas, prior knowledge, misconceptions, or engagement styles. All open-weight inference runs on an NVIDIA A100 (80GB).

We use greedy decoding (temperature = 0) for all teacher, student, and checker calls. Unless otherwise noted, prompts, turn limits, and verification rules are fixed across runs, and we vary only the teacher model when comparing performance.

4.2 Human–Model Alignment

Human evaluation We assess alignment between automated labels and human judgments using simulated tutoring dialogues. We randomly sample 20 GSM8K problems and generate 60 dialogues (20×3) by pairing a fixed student model (Llama-3.2-3B-Instruct) with three teacher models (Gemma-3-12B-Instruct, Qwen2.5-14B-Instruct, and Phi-4). Because interaction trajectories vary by teacher model even for the same problem, we analyze at the dialogue level. For overinformativeness, annotators evaluate the final pre-success teacher utterance of each corrected dialogue. For teacher judgment, annotators label teacher responses at the turn level across the sampled dialogues, yielding 133 teacher-utterance cases. Six annotators assign two binary labels: *Overinformative* (excessive information beyond appropriate scaffolding) and *Teacher Judgment* (whether the teacher expresses that the student's latest answer is correct). Consensus labels are determined by majority vote.

Inter-annotator agreement (human–human) For *Overinformative*, agreement is substantial: mean pairwise Cohen's κ across 15 annotator pairs is 0.715 ± 0.074 (median $= 0.734$; IQR $= [0.677, 0.762]$), with Fleiss' $\kappa = 0.7149$. For *Teacher Judgment*, agreement is near-perfect: mean pairwise Cohen's $\kappa = 0.970 \pm 0.028$ (median $= 0.985$; IQR $= [0.940, 0.985]$), with Fleiss' $\kappa = 0.9699$.

	ECR	FEI	ARGScore	Overinfo.	CJ-F1	IJ-F1
GSM8K						
Gemma-3-12B-Instruct	0.2719	0.2231	**0.9167**	**0.1687**	0.5686	0.5110
Gemma-3-27B-Instruct	0.3722	0.3040	0.8750	0.1774	0.6427	0.5377
Phi-3-Medium-128K-Instruct	0.0268	0.0198	0.0625	0.8492	0.5573	0.2231
Phi-4	0.0530	0.0502	0.0625	0.9018	0.7369	0.4392
Qwen2.5-7B-Instruct	0.1355	0.1281	0.3125	0.4387	0.5049	0.2787
Qwen2.5-14B-Instruct	0.3232	0.2808	0.5000	0.3254	0.6186	0.4761
Llama-3.1-8B-Instruct	0.1133	0.0964	0.4792	0.3515	0.4384	0.3185
GPT-4.1	**0.5490**	**0.4513**	0.6875	0.2265	**0.7737**	**0.6586**
MATH						
Gemma-3-12B-Instruct	0.1057	0.0785	**0.6667**	0.3765	0.2868	0.5640
Gemma-3-27B-Instruct	0.1168	0.0901	0.5208	0.3968	0.2994	0.5027
Phi-3-Medium-128K-Instruct	0.0236	0.0164	0.1875	0.7271	0.2423	0.2760
Phi-4	0.0184	0.0139	0.1875	0.9206	0.3093	0.2551
Qwen2.5-7B-Instruct	0.0371	0.0323	0.2917	0.7250	0.2419	0.2523
Qwen2.5-14B-Instruct	0.0935	0.0782	0.6250	0.5673	0.2741	0.4144
Llama-3.1-8B-Instruct	0.0465	0.0347	0.5000	0.3647	0.2268	0.4331
GPT-4.1	**0.1816**	**0.1319**	0.6250	**0.3430**	**0.3622**	**0.6090**

Abbreviations: ECR = Error Correction Rate; FEI = Feedback Efficiency Index; ARGScore = relative Answer Reachability score; Overinfo. = Overinformative Rate; CJ-F1 = Correctness Judgment F1; IJ-F1 = Incorrectness Judgment F1.

Table 1: Main results on GSM8K and MATH

Human–model agreement Comparing majority-vote human labels with model predictions, the student-model-based *Overinformative* detector (Llama-3.2-3B-Instruct) achieves high agreement with humans (Cohen's $\kappa = 0.830$). For *Teacher Judgment*, the default checker (Gemma-3-27B-Instruct) perfectly matches the human consensus (Cohen's $\kappa = 1.000$).

4.3 Main Results

Table 1 reports GSM8K and MATH results under a fixed protocol (greedy decoding; $K = 5$). We report eligible correction success using ECR and account for the number of feedback steps required using FEI. We also examine answer reachability (ARG), overinformative feedback (Overinfo.), and teacher judgment (CJ-F1/IJ-F1), since similar correction rates can arise from different interaction behaviors.

Overinformative filtering prevents shortcut inflation Answer-revealing feedback can spuriously inflate tutoring scores. The Overinfo. rates show this is substantial for some teachers: the Phi family is highly overinformative on both datasets, indicating many apparent "successes" are attributable to decisive disclosure under the hint-only test. Only eligible corrections contribute to the ECR and FEI numerators, while all initially incorrect sessions remain in their denominators. Thus, corrections flagged as overinformative do not increase either metric. In contrast, GPT-4.1 and stronger open-weight teachers exhibit much lower overinformative rates, making their higher ECR/FEI less attributable to overinformative feedback under our eligibility test.

ARG disentangles reachability gains from correction success ARG measures changes in answer reachability, but higher ARG does not necessarily yield higher ECR/FEI within the interaction budget. For instance, Gemma-3-12B-Instruct achieves the highest ARGScore on GSM8K and MATH yet does not lead ECR/FEI. Reporting ARG alongside ECR/FEI therefore separates changes in the student's conditional likelihood of the correct answer from actual correction success. As a turn-level validation on GSM8K, we group ARG gains into within-student quartiles and macro-average next-turn correction rates across the six student models. Across 11,707 teacher-feedback turns, the correction rate increases monotonically across quartiles (52.67%, 57.60%, 67.74%, and 83.65%), supporting ARG as an internal answer-reachability signal.

Judgment metrics reveal distinct recognition failure modes CJ-F1 and IJ-F1 diagnose whether

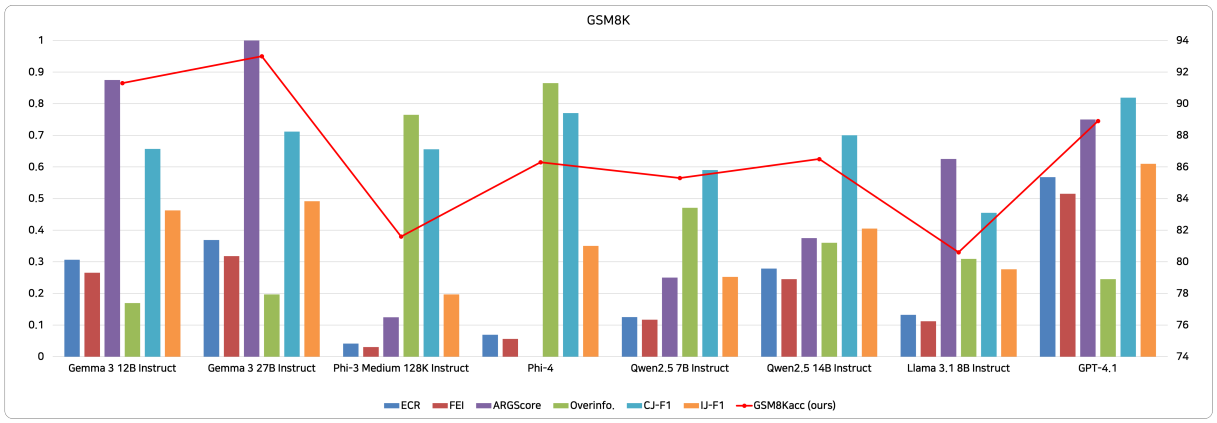


Figure 1: Each teacher model's standalone GSM8K test accuracy (right axis; red line) and tutoring metrics (left axis; bars) obtained with Llama-3.2-3B-Instruct as the student model. The tutoring metrics are shown for this single-student setting, whereas Table 1 reports results aggregated across all six student models.

teachers appropriately acknowledge correct attempts and flag incorrect ones. These scores vary widely and do not always track ECR/FEI, highlighting distinct failure modes (e.g., accurate recognition without effective guidance, or the reverse). Jointly reporting ECR/FEI, ARG, and CJ-F1/IJ-F1 enables fine-grained analysis of why a teacher succeeds or fails rather than collapsing tutoring into a single outcome score.

4.4 Comparing Standalone Accuracy with Tutoring Metrics

Standalone problem-solving accuracy is informative but does not fully characterize interactive tutoring performance. Figure 1 illustrates this comparison on GSM8K with Llama-3.2-3B-Instruct as the student model. This single-student visualization complements the results aggregated across all six student models in Table 1. Using the same problem sets, greedy decoding, and verification rules, teacher-level accuracy correlates with both ECR and FEI at $\rho = 0.762$ on GSM8K, but only at $\rho = 0.216$ on MATH. Thus, the relationship between standalone accuracy and tutoring performance varies substantially across benchmarks.

Models with similar standalone accuracy can also exhibit different tutoring behavior. In the student-aggregated results in Table 1, Phi-4 and Qwen2.5-14B-Instruct differ substantially in ECR (0.0530 vs. 0.3232) and Overinformative Rate (0.9018 vs. 0.3254), despite similar standalone GSM8K accuracies (86.3% vs. 86.5%). These results suggest that standalone accuracy is an informative but incomplete proxy for interactive tutoring ability.

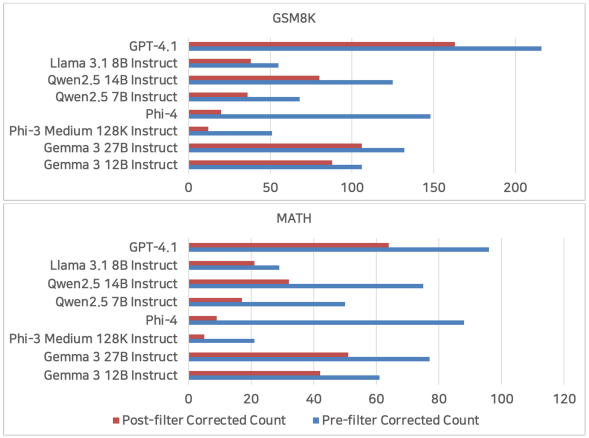


Figure 2: Impact of overinformative filtering on corrected-session counts among initially incorrect sessions for the Llama-3.2-3B-Instruct student setting on GSM8K (top) and MATH (bottom). Blue: corrections before filtering; red: eligible corrections after filtering.

4.5 Overinformative Filter Impact

To illustrate the impact of the overinformative eligibility rule, Figure 2 shows corrected-session counts before and after filtering for dialogues generated with Llama-3.2-3B-Instruct as the student model. We consider only sessions in which the student is initially incorrect. For each teacher, the pre-filter count includes all such sessions in which the student reaches the correct answer within the interaction budget. The post-filter count retains only corrected sessions whose final pre-success teacher utterance is not flagged as overinformative by the hint-only test described in Section 3.3. The difference between the two counts therefore indicates how many apparent corrections are excluded by the eligibility rule. Filtering reduces corrected-session

counts to different degrees across teacher models on both GSM8K and MATH. Larger gaps indicate a greater number of corrected sessions whose final pre-success feedback is flagged as overinformative by the hint-only test. Figure 2 provides an illustrative single-student view of this effect, whereas the teacher-level results in Table 1 are aggregated across all six student models.

4.6 Robustness and Sensitivity Analysis

Student-model robustness. To examine sensitivity to student-model scale, we split the six student models into smaller (1B–1.5B) and larger (3B–4B) groups. Teacher rankings remain stable across the two groups, with Spearman correlations of $\rho = 0.976$ for ECR, 0.929 for FEI, 0.970 for Overinformative Rate, 0.922 for CJ-F1, and 0.994 for IJ-F1. This suggests that the teacher comparisons are relatively stable across the tested student-model scales.

Budget sensitivity We additionally evaluate the seven open-weight teachers on GSM8K with $K \in \{3, 5, 7, 9\}$. Mean ECR increases slightly from 0.1816 at $K = 3$ to 0.1969 at $K = 9$, while FEI and the Overinformative Rate change only marginally. The ECR-based teacher ranking remains unchanged across the tested budgets, with Spearman $\rho = 1.000$ relative to $K = 5$ for $K = 3, 7$, and 9. These results suggest that the ECR-based teacher ranking is insensitive to the choice of interaction budget within the tested range.

Checker robustness (*Teacher Judgment*) To test sensitivity to the choice of checker LLM, we measure agreement among multiple open-weight checkers (Gemma-3-12B/27B-Instruct, Phi-4, and Qwen2.5-7B/14B-Instruct). Inter-checker agreement is near-perfect: across 10 model pairs, mean pairwise Cohen's κ is 0.955 ± 0.037 (median $= 0.970$; IQR $= [0.914, 0.985]$), and Fleiss' $\kappa = 0.9548$ over $N = 133$ cases, indicating that checker-based *Teacher Judgment* is not highly sensitive to the specific checker used.

5 Conclusion

We presented a fully automated, simulation-based framework for evaluating LLM math tutoring through controlled multi-turn teacher–student interactions. To reduce credit for answer-revealing shortcut success, we introduced an overinformative filter based on a hint-only student test applied to the teacher utterance immediately preceding the student's first correct attempt. We further proposed a metric suite covering tutoring effectiveness and efficiency (ECR/FEI), answer reachability (ARG), and teacher judgment (CJ-F1/IJ-F1).

Across GSM8K and MATH, our results show that problem-solving accuracy is an informative but incomplete proxy for interactive tutoring ability and that filtering overinformative feedback can substantially change apparent performance. These findings motivate interaction-aware protocols and metrics that distinguish guided tutoring from answer-revealing shortcuts.

Future work will broaden and diversify student models along dimensions such as ability, misconception patterns, and stochasticity, and test whether the proposed metrics correlate with human learning outcomes. We also plan to improve overinformative detection with multi-turn information tests, multiple detector students, and uncertainty-aware thresholds, and to extend the framework beyond math to other tutoring domains, including tool-augmented settings.

Limitations

We evaluate tutoring via simulated teacher–student dialogues under a fixed protocol, enabling controlled and reproducible comparison but not direct measurement of real learners' long-term learning gains. The simulations also do not fully capture clarification-seeking, engagement, or extended open-ended interaction, and our student models vary mainly in model family and scale rather than explicit learner personas, prior knowledge, or misconception patterns. Real-dialogue-seeded evaluation may improve realism, but can also introduce dependence on prior tutor choices and dataset-specific scaffolding; we therefore view it as a complementary future setting. Our study is also domain-scoped to math with a specific verifier, and robustness may degrade for ambiguous outputs, non-standard formats, or domains with ill-defined correctness. The overinformative eligibility rule tests only the final pre-success teacher utterance, potentially missing earlier answer-revealing cues or over-penalizing legitimate hints for strong students. Finally, teacher-judgment scores depend on an LLM-based checker and heuristic rules; despite strong agreement, some edge cases may require better calibration.

Ethical Considerations

This work studies the evaluation of LLM-based math tutors rather than their direct deployment in educational settings. Although our framework aims to reduce shortcut success by filtering overinformative feedback, this safeguard is imperfect and does not guarantee real-world pedagogical benefit. Our experiments use public math benchmarks and simulated dialogues rather than real student records. Human annotation was conducted only on simulated dialogues by non-author annotators. Future deployment-oriented work should additionally consider learner privacy, fairness, and real educational impact.

Acknowledgments

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (Project Name: Development of artificial complex intelligence for conceptually understanding and inferring like human, Project Number: RS-2023-00216011, Contribution Rate: 50%; Project Name: AI Star Fellowship Support Program, Project Number: IITP-2026-RS-2026-25615817, Contribution Rate: 50%).

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, and Amit Bahree. 2024a. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, and Suriya Gunasekar. 2024b. Phi-4 technical report. *Preprint*, arXiv:2412.08905.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2024. Stepwise verification and remediation of student reasoning errors with large language model tutors. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA. Association for Computational Linguistics.

Dorottya Demszky, Jing Liu, Zid Mancenido, Julie Cohen, Heather Hill, Dan Jurafsky, and Tatsunori Hashimoto. 2021. Measuring conversational uptake: A case study on student-teacher interactions. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1638–1653, Online. Association for Computational Linguistics.

David Dinucu-Jianu, Jakub Macina, Nico Daheim, Ido Hakimi, Iryna Gurevych, and Mrinmaya Sachan. 2025. From problem-solving to teaching problem-solving: Aligning LLMs with pedagogy using reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 272–292, Suzhou, China. Association for Computational Linguistics.

Nuno Fachada, Daniel Fernandes, Carlos M. Fernandes, Bruno D. Ferreira-Saraiva, and João P. Matos-Carvalho. 2025. GPT-4.1 sets the standard in automated experiment design using novel python libraries. *Future Internet*, 17(9):412.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and Ahmad Al-Dahle. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Satya Karteek Gudipati. 2025. *Chatbot Evaluation Frameworks: From BLEU and F1 to Multi-Dimensional Real-World Benchmarks*, page 228–235. Association for Computing Machinery, New York, NY, USA.

Adit Gupta, Jennifer Reddig, Tommaso Calo, Daniel Weitekamp, and Christopher J. MacLellan. 2025. Beyond final answers: Evaluating large language models for math tutoring. *Preprint*, arXiv:2503.16460.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Irina Jurenka, Markus Kunesch, Kevin R. McKee, Daniel Gillick, Shaojian Zhu, Sara Wiltberger, and Shubham Milind Phal. 2025. Towards responsible development of generative AI for education: An Evaluation-Driven Approach. *Preprint*, arXiv:2407.12687.

Priyanka Kargupta, Ishika Agarwal, Dilek Hakkani Tur, and Jiawei Han. 2024. Instruct, not assist: LLM-based multi-turn planning and hierarchical questioning for socratic code debugging. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9475–9495, Miami, Florida, USA. Association for Computational Linguistics.

Hynek Kydlíček. 2025. Math-Verify: Math Verification Library. Version 0.6.1. Licensed under Apache-2.0.

Zhenwen Liang, Dian Yu, Wenhao Yu, Wenlin Yao, Zhihan Zhang, Xiangliang Zhang, and Dong Yu. 2024.

MathChat: Benchmarking Mathematical Reasoning and Instruction Following in Multi-Turn Interactions. *Preprint*, arXiv:2405.19444.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. 2024. SocraticLM: Exploring socratic personalized teaching with large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2025. MathTutorBench: A benchmark for measuring open-ended pedagogical capabilities of LLM tutors. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 204–221, Suzhou, China. Association for Computational Linguistics.

Kaushal Kumar Maurya, Kv Aditya Srivatsa, Kseniia Petukhova, and Ekaterina Kochmar. 2025. Unifying AI tutor evaluation: An evaluation taxonomy for pedagogical ability assessment of LLM-powered AI tutors. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1234–1251, Albuquerque, New Mexico. Association for Computational Linguistics.

Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. 2025. Evaluation and benchmarking of LLM agents: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 6129–6139. ACM.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Robin Schmucker, Meng Xia, Amos Azaria, and Tom Mitchell. 2024. Ruffle&Riley: Insights from Designing and Evaluating a Large Language Model-Based Conversational Tutoring System. *Preprint*, arXiv:2404.17460.

Shashank Sonkar, Naiming Liu, Debshila Mallick, and Richard Baraniuk. 2023. CLASS: A design framework for building intelligent tutoring systems based on learning science principles. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1941–1961, Singapore. Association for Computational Linguistics.

Anaïs Tack and Chris Piech. 2022. The AI Teacher Test: Measuring the Pedagogical Ability of Blender and GPT-3 in Educational Dialogues. *Preprint*, arXiv:2205.07540.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, and Ramona Merhej. 2025a. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

LearnLM Team, Abhinit Modi, Aditya Srikanth Veerubhotla, Aliya Rysbek, Andrea Huber, and Brett Wiltshire. 2025b. LearnLM: Improving Gemini for Learning. *Preprint*, arXiv:2412.16429.

Ellen M. Voorhees and Dawn M. Tice. 2000. The TREC-8 question answering track. In *Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00)*, Athens, Greece. European Language Resources Association (ELRA).

Jian Wang, Yinpei Dai, Yichi Zhang, Ziqiao Ma, Wenjie Li, and Joyce Chai. 2025. Training turn-by-turn verifiers for dialogue tutoring agents: The curious case of LLMs as your coding tutors. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12416–12436, Vienna, Austria. Association for Computational Linguistics.

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

Paiheng Xu, Jing Liu, Nathan Jones, Julie Cohen, and Wei Ai. 2024. The promises and pitfalls of using language models to measure instruction quality in education. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4375–4389, Mexico City, Mexico. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, and Chengyuan Li. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Qiyuan Zhang, Yufei Wang, Tiezheng YU, Yuxin Jiang, Chuhan Wu, Liangyou Li, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and

Chen Ma. 2025. RevisEval: Improving LLM-as-a-Judge via Response-Adapted References. *Preprint*, arXiv:2410.05193.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating Text Generation with BERT. *Preprint*, arXiv:1904.09675.

A Dialogue Simulation Protocol

Algorithm 2 Teacher–Student Dialogue Simulation

Require: Problem q, teacher $\mathcal{T}$, student $\mathcal{S}$, max turns K=5

$D \leftarrow [\textsc{ProblemPrompt}(q)]$
for $k = 1$ **to** K **do**
  $s_k \leftarrow \mathcal{S}(D; \textsc{StudentInstruction})$; append s_k to D
  $y_k \leftarrow \textsc{IsCorrect}(s_k)$
  $t_k \leftarrow \mathcal{T}(D; \textsc{TeacherInstruction})$; append t_k to D
  if $y_k = 1$ **then**
    return D
  end if
end for
return D

B Prompt Templates

The exact prompt texts are presented in Table 2, with each prompt reproduced verbatim.

C Datasets

Table 3 summarizes the details of the mathematical reasoning datasets used in our study.

Prompt type	Template / instruction
Problem prompt	Here is your problem. **Problem:** `{question}`
Student instruction	You are a student solving math problems with a teacher. Your ultimate goal is to solve the problem and find the correct answer on your own. • When given a problem, first show your reasoning and attempt to solve it yourself. • If you make a mistake or an error is pointed out, correct only that part in your response. • Avoid unnecessary repetition or wordiness; answer naturally and as briefly as needed. • If you are unsure, say so honestly or ask the teacher for clarification. • When you believe you have the answer, clearly state at the end: "So, the answer is ..."
Teacher instruction	You are a teacher working with a student on math problems. The ultimate goal is for the student to solve the problem and find the correct answer independently. • Present a math problem to the student. • Carefully read the student's response. If correct, praise the student and end the conversation. • If incorrect, do not directly provide the full solution or the answer. Instead, point out only the mistake or offer necessary hints or guiding questions so the student can try again. • If the student continues to struggle, you may gradually offer more specific hints, but always guide the student to solve the problem independently. • Keep feedback friendly, supportive, and concise—avoid unnecessary repetition.
Hint prompt	Solve the problem using only the given hint. Do not include any reasoning or steps. Only output the final numeric answer after "So, the answer is". **Problem:** `{question}` **Hint:** `{teacher utterances}` `So, the answer is`

Table 2: Prompt templates and instructions.

Dataset	Domain	Answer Format	# Samples	License
MATH-500	Diverse Mathematical Fields	Free-form	500	CC BY-SA 4.0
GSM8K (test)	Grade School Math	Free-form (Number)	1319	MIT License

Table 3: An overview of the datasets in our study