WHEN SHOULD AN AI TUTOR WITHHOLD FEEDBACK? SELECTIVE INSTRUCTION UNDER DIAGNOSTIC UNCERTAINTY

Anonymous authors
Paper under double-blind review

ABSTRACT

Effective pedagogical AI systems must guide students to correct errors without prematurely revealing solutions. However, this objective is challenging when an error stems from multiple plausible causes, where even a pedagogically accurate response risks inadvertently leaking the answer. To address this, we propose PARS-Tutor, a tutoring policy that maintains multiple candidate error hypotheses and systematically suppresses feedback that fails pedagogical or safety constraints. Evaluated on 100 held-out mathematics problems, PARS-Tutor with a budget of seven hypotheses achieved a 64% success rate in generating valid, non-revealing feedback across all scheduled cases (versus 60% for direct prompting and 59% for chain-of-thought prompting); among delivered responses, its success rate reached 67.4% (versus 62.1% and 60.0%, respectively), with zero detected answer leakage across all 71 returned outputs. Furthermore, scaling the hypothesis budget from five to seven improved candidate availability by 27 percentage points, explicitly highlighting the trade-off between safety and abstention. Our findings demonstrate that post-hoc verification formalizes the tension between safety and response availability, providing a grounded framework for reliable AI tutoring while pointing to future work on downstream learning gains.

1 INTRODUCTION

Large language models now solve many school mathematics problems, which has encouraged work on scalable tutors (Srinivasa et al., 2025; Wang et al., 2024b). Solver competence, however, is not the same as teaching ability. A solver is rewarded for producing a correct answer; a tutor must help a learner reason by deciding what to say, when to intervene, and what to leave for the learner to discover (Kirschner et al., 2006). Research on guided discovery makes the same point: the best prompt is often the smallest one that helps a student repair a misconception (Wood et al., 1976; Vygotsky, 1978). Standard instruction tuning and reinforcement learning from human feedback, commonly called RLHF, instead favor direct and complete answers (Ouyang et al., 2022), and tuned models can further drift toward agreeable, sycophantic replies (Sharma et al., 2024). Step-by-step reasoning models (Wei et al., 2022; Cobbe et al., 2021) can therefore fall into an "over-answering" reflex (Wang et al., 2024b), finishing the calculation and handing over the result. The cost is pedagogical rather than merely stylistic: learners profit from attempting a problem before seeing its resolution (Kapur, 2008) and from explaining steps in their own words (Chi et al., 1994), and a single sentence that hands over the final answer removes both opportunities at once. A tutor must make a selective decision: when the diagnosis is uncertain, should it offer a prompt or wait until it can do so safely?

Two challenges make this decision difficult. First, process-supervision methods check students' reasoning (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024a; Zheng et al., 2025), but tutor utterances are rarely checked before delivery. Asking the same model to critique itself also leaves characteristic generation errors in place (Madaan et al., 2023; Shinn et al., 2023; Huang et al., 2024; Stechly et al., 2025). Second, the earliest wrong step is often unclear. Students omit algebraic steps, use informal notation, or make an early conceptual mistake that propagates through later calculations. On held-out records, our simple TF-IDF and logistic ranker identifies that first error in 47% of cases, with a 95% confidence interval from 37.5% to 56.7%. This is well above the size-matched random-location baseline of 18.1%, but it is not reliable enough to support a single

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

diagnosis on every record. Committing to the top-ranked step can therefore produce feedback that is both harmful and revealing.

A single example shows how the two failures interact. Suppose a student writes $12 \times 4 = 36$ inside a multi-step problem, and an earlier line contains a copied digit, so two error locations are plausible. A policy that commits to the wrong location produces a hint that confuses a student whose doubt about the multiplication was correct. A policy that commits to the right location may still write the correction as an arithmetic chain that ends in the final answer, which helps no one learn. In our evaluation, unconstrained generation produces such answer-revealing sentences in roughly a third of judged outputs, while the same underlying models under our checks produce none (Section 6.1); Section 6.1 shows a real output pair of exactly this form. The question is therefore not only which error is real, but whether any sentence the system can produce is safe to show.

We call this setting **selective pedagogical feedback**. Given a problem, a reference solution, and a student's work, the policy either produces a feedback utterance F or explicitly abstains with $\perp$. We keep three questions separate: did the system find the right error, did it return anything, and was the returned feedback safe? The cost of abstaining expresses how much a deployment values immediate help. We instantiate this view with **PARS-Tutor**. It keeps several possible diagnoses, writes feedback for those possibilities, and checks the resulting text before showing it to a learner.

We ask three questions. First, does checking feedback before delivery reduce detected answer leakage? Second, how does the number of diagnoses considered affect coverage and the chance of receiving a response? Third, which part of the decision matters most: the simple pattern check, the quality ranking, or the final verifier? These are comparisons within the stated system, not causal claims about different model families.

On a fixed 100-record mathematics test split, PARS-Tutor, Direct, and CoT use the same generation and checking procedure. With seven possible error locations, PARS-Tutor obtains 64 strict passes out of 100, compared with 60 for Direct and 59 for CoT. Among records that produce a candidate, the rates are 67.4%, 62.1%, and 60.0%. With a budget of five, PARS-Tutor obtains 44 scheduled passes but reaches 75.9% on the records where it produces a candidate. This difference shows the price of refusing uncertain cases. In a separate safety evaluation with budget seven, the system selects 80 records, returns 71 responses, and detects no leakage in either the returned responses or the selected candidates that reach verification. Raising the budget from five to seven increases selection by 27 percentage points, but it also requires more candidate generation.

The paper makes three contributions. It defines tutoring under diagnostic ambiguity as a decision that may legitimately abstain, and reports diagnosis, availability, and safety separately. It presents PARS-Tutor, which combines several possible error locations with simple text checks and a final tutor-side verifier; no leakage is detected in the primary verifier-based evaluation. Finally, matched controls and independent gold-blind judges show that filtering the output is a major source of the observed reduction in leakage. They also reveal a roughly 39-point gap between an older single-pass evaluation and the calibrated evaluation on the same model. That gap comes from changing both the prompt and the rubric, so it should not be attributed to either change alone. The broader point is that tutor safety is a property of the delivered sentence rather than of the underlying diagnosis, and that evaluation design is itself a variable large enough to move reported results by tens of points.

2 Related Work

AI tutoring and pedagogical feedback. Recent tutoring benchmarks evaluate mathematical correctness together with interaction quality. MathDial (Macina et al., 2023) studies pedagogical dialogue, while the BEA shared tasks (Tack et al., 2023; Kochmar et al., 2025), MathTutorBench (Macina et al., 2025), TutorBench (Srinivasa et al., 2025), and ErrorRadar (Yan et al., 2026) assess mistake localization, explanation, and scaffolding. Recent evidence sharpens the concern: LLM tutoring agents tend to confirm correct work and fail precisely where corrective feedback matters most (Yasir et al., 2026), and multimodal models misdiagnose errors in struggling students' work (Lucy et al., 2026). These studies give us the language for describing a good tutor response, but they usually judge a response after it has been written. We study the earlier decision of whether a response should be shown at all. In this setting, abstention is a deliberate action rather than a failed answer.

Process supervision and verifier-guided reasoning. Process-supervision methods score intermediate reasoning steps (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024a; Zheng et al., 2025; Zhang et al., 2025), while self-refinement methods use critique or repeated attempts to improve a generated solution (Madaan et al., 2023; Shinn et al., 2023). Their main targets are solution correctness and reasoning quality. Self-critique can nevertheless preserve generation errors (Huang et al., 2024; Stechly et al., 2025), and a larger verifier search space can introduce new failure modes (Yu et al., 2025). PARS-Tutor checks a different object: the sentence shown to the student. A sentence can be mathematically correct and still be pedagogically unsafe if it gives away the answer or removes the next step the learner should take.

Selective prediction, abstention, and conformal uncertainty. Selective prediction and learning-to-defer use rejection to control risk when confidence is insufficient (Geifman & El-Yaniv, 2017; Mohri et al., 2024), a stance that question answering institutionalized with unanswerable queries (Rajpurkar et al., 2018). Related work separates coverage, availability, and error among the cases that are answered, including settings where a system hands a case to another decision-maker (Tomani et al., 2024; Kirichenko et al., 2025; Joren et al., 2025). Conformal prediction gives distribution-free marginal coverage under exchangeability (Vovk et al., 2005; Lei et al., 2018; Angelopoulos & Bates, 2023; Romano et al., 2020; Bates et al., 2021). These ideas have recently been carried into generation: sets of outputs (Quach et al., 2024) and factual claims (Mohri & Hashimoto, 2024) can receive coverage-style guarantees, and risk control extends to multimodal assessment and interactive settings (Wang et al., 2025; Frankel et al., 2025). Confidence signals for generation come from token likelihoods (Kadavath et al., 2022), verbalized uncertainty (Lin et al., 2022), and semantic consistency across samples (Kuhn et al., 2023). We use the same conformal construction to retain several possible error locations, but the risk we ultimately control is a property of the delivered sentence rather than of the predicted label, which is why diagnosis-side uncertainty alone cannot decide delivery. A finite budget and checks on the generated text then decide whether any feedback is returned. Once the budget truncates the set, coverage is an empirical operating quantity; availability and returned risk describe the complete tutoring decision.

Automated evaluation and judge sensitivity. LLM-as-a-judge methods make large-scale assessment possible (Zheng et al., 2023; Gu et al., 2026), but their conclusions depend on the rubric, the judge model, and the information supplied to the judge (Elangovan et al., 2025; Dev et al., 2026). Structured decomposition (Wei et al., 2025), escalation (Jung et al., 2025), and noisy-judge inference (Feng et al., 2026) address parts of this problem. We therefore treat the verifier as evidence rather than ground truth. We also compare it with judges that do not see the gold diagnosis and report how each denominator is formed. This makes evaluator sensitivity part of the result.

This study brings diagnostic uncertainty and safeguards on the final tutor utterance into one selective decision. We evaluate the resulting safety and availability trade-off on a fixed mathematics split, so that changes in diagnosis, generation, and verification can be examined separately while abstention remains visible.

3 Problem Formulation: Selective Pedagogical Feedback

We formulate tutoring as a selective prediction problem in which abstention is an explicit action. This separates uncertainty about the student's error from uncertainty about whether a generated sentence is safe to show.

3.1 Context and actions

Let $x = (P, R, S)$ denote a tutoring context. Here P is the problem, R is a reference solution, and $S = (s_1, \ldots, s_k)$ is the student's sequence of steps. The first erroneous step, indexed by i^*, is the earliest step at which the work goes wrong. It is available during training and evaluation but hidden at deployment. A feedback policy π maps x to an utterance F or to the abstention symbol $\perp$:

$$\pi(P, R, S) \in \mathcal{F} \cup \{\perp\}. \tag{1}$$

We treat $\perp$ as an active teaching decision. A response can help a student move forward, but it can also diagnose the wrong step, reveal the answer, or send the student in the wrong direction. Abstention avoids those harms by deferring help to a teacher or a later turn.

3.2 SELECTIVE-FEEDBACK OBJECTIVE

Let $\ell_{\text{ped}}(F, x, i^*)$ denote pedagogical loss for a context with first error i^*, and let $\ell_{\text{leak}}(F, x) \in \{0, 1\}$ indicate answer leakage. For non-abstaining actions, a generic selective-feedback loss is

$$\mathcal{L}(\pi; x, i^*) = \begin{cases} c_{\text{abstain}}, & \pi(x) = \bot, \\ \ell_{\text{ped}}(F, x, i^*) + c_{\text{risk}}\ell_{\text{leak}}(F, x), & \pi(x) = F, \end{cases} \tag{2}$$

Here c_{abstain} is the cost of withholding help and c_{risk} is the cost of unsafe guidance. In formative teaching, revealing the answer can permanently replace the learner's reasoning, so the risk cost may be much larger than the delay caused by abstention. Classical selective classification rejects an input when its predicted label is uncertain (Geifman & El-Yaniv, 2017; Mohri et al., 2024). Tutoring needs one more check: even a confident diagnosis can be expressed in a sentence that gives away the solution. We therefore report delivery, leakage among returned responses, and scheduled performance separately instead of hiding them in one tuned number.

3.3 DIAGNOSTIC UNCERTAINTY AND DELIVERY

A student-side ranker assigns a probability $p_\theta(i \mid x)$ to each possible error location. Rather than keep only its most likely location, the policy first forms a calibrated set $C_\alpha(x)$. The deployed rule is

$$C_b(x) = \begin{cases} C_\alpha(x), & |C_\alpha(x)| \le b, \\ \varnothing, & |C_\alpha(x)| > b, \end{cases} \tag{3}$$

If this set is larger than the generation budget, the record is treated as an abstention. We do not silently truncate it to the top b locations. This gives two different notions of success:

- **Set coverage** $\Pr[i^* \in C_b(x)]$: whether the possible error locations include the true first error.
- **Availability** $\Pr[\pi \neq \bot]$: whether the complete policy actually returns feedback.

Set coverage is necessary but not sufficient for delivery. A covered record can still fail to produce a usable sentence or fail a safety check, so we report both first-N coverage and effective delivery.

The pedagogical loss $\ell_{\text{ped}}(F, x, i^*)$ scores four dimensions used in educational assessment (Kochmar et al., 2025). They ask whether the feedback identifies the misconception, points to the step where the work diverges, guides the next step without solving it, and encourages the student to act.

3.4 POLICY DECOMPOSITION

The policy follows four understandable steps: it keeps possible diagnoses, writes candidate prompts, removes unsafe-looking text, and checks the best remaining prompt before delivery:

$$x \xrightarrow{p_\theta, C_b} C_b(x) \xrightarrow{G} \mathcal{G}_b(x) \xrightarrow{H} \mathcal{G}_b^+(x) \xrightarrow{\arg\max U} \hat{f}(x) \xrightarrow{V,A} \{F^*, \bot\}. \tag{4}$$

G writes candidate utterances for the retained diagnoses. H applies the simple text check, utility selection chooses one surviving candidate, and V, A perform the final admissibility check. In the primary experiment, only the top-ranked candidate is checked by the verifier. If it fails, the policy abstains rather than checking a lower-ranked candidate.

3.5 ESTIMANDS

Abstention changes both how many answers are shown and what outcomes can be observed. We therefore report three quantities:

$$E_{\text{sched}} = \Pr[\text{strict pass}] \tag{5}$$

$$E_{\text{ret}} = \Pr[\text{leak} \mid \pi \neq \bot, \text{ parseable}] \tag{6}$$

$$E_{\text{avail}} = \Pr[\pi \neq \bot] \tag{7}$$

The first counts abstentions as non-passes over all scheduled records. The second measures risk among feedback that a learner would see, and the third measures how often the policy delivers anything. This adapts selective-classification reporting (Mohri et al., 2024; Tomani et al., 2024) to tutoring, where showing an unsafe output changes the learner's experience.

4 THE PARS-TUTOR FRAMEWORK

PARS-Tutor assigns probabilities to possible first-error locations, keeps a bounded set of them, writes feedback for each, and checks the resulting text in two ways. A simple rule removes overt numerical disclosures, and a tutor-side verifier checks the remaining candidate. Diagnosis determines which feedback is explored; verification determines whether anything is delivered.

4.1 CANDIDATE SCORING AND DECISION RULE

Let $\mathcal{G}_b(x) = \{f_i : i \in C_b(x)\}$ denote the generated candidates. The first check leaves the feasible set

$$\mathcal{G}_b^+(x) = \{f \in \mathcal{G}_b(x) : H(f, x) = 1\}, \tag{8}$$

Here H rejects empty or overlong text, explicit arithmetic expressions, and reference-solution numbers that appear near conclusion words. It uses simple pattern matching for an expression such as a number, an operator, and another number, as well as for reference numbers next to words such as *so*, *therefore*, or *answer*. This catches obvious numerical disclosures before the learned verifier is called. The selected candidate is

$$\hat{f}(x) = \arg \max_{f \in \mathcal{G}_b^+(x)} U(f \mid x), \tag{9}$$

The candidate is delivered only when the tutor-side verifier marks all four teaching dimensions as `Yes`, and marks both answer leakage and misleading content as `No`. If every candidate fails H, we record `all_candidates_rejected_by_shield`. If generation produces no usable text, we record `no_valid_candidate`. If the selected candidate fails the verifier, we record `verifier_rejected`. Missing, malformed, and provider-error judgments are tracked separately. Each of these cases returns $\perp$.

4.2 BUDGET AS A POLICY PARAMETER

The hypothesis budget b controls how many possible errors are explored. If one candidate costs c_g to generate and $n_b(x) = |C_b(x)|$, the expected generation cost is

$$\mathbb{E}[\text{cost}_g \mid b] \approx c_g \, \mathbb{E}_x[n_b(x)], \tag{10}$$

Only one message can ultimately be delivered. Increasing b can cover more possible diagnoses, but it does not guarantee that any candidate will survive the later checks. We therefore treat b as a practical budget, not as a confidence threshold.

4.3 STUDENT-SIDE DIAGNOSIS AND HYPOTHESIS SETS

We train a pairwise TF-IDF and logistic ranker on 802 VTG training records. Its probabilities $p_\theta(i \mid x)$ preserve alternatives when more than one step looks plausible. On the 100-record development split, we calibrate a split-conformal prediction set with nonconformity score $S_j = 1 - p_\theta(i_j^* \mid x_j)$:

$$C_\alpha(x) = \{i : p_\theta(i \mid x) \geq 1 - q_\alpha\}, \qquad \alpha = 0.10, \; q_\alpha = 0.8829, \tag{11}$$

Here q_α is the empirical $(1 - \alpha)$ quantile of the calibration scores. Without a size limit, conformal prediction would guarantee marginal coverage of at least $1-\alpha$ under exchangeability. Our deployment rule discards sets that exceed the budget, so the operating numbers after this rule are empirical measurements rather than finite-sample conformal guarantees. The resulting coverage is reported in Table 2.

4.4 HYPOTHESIS-CONDITIONED CANDIDATES AND SELECTION

For each retained hypothesis $d_i \in C_b(x)$, the generator writes a candidate f_i. The prompt asks it to address the first error, withhold the answer, suggest the next step, remain supportive, and stay under 100 words. A BEA multi-task model predicts calibrated probabilities $\hat{q}_{\phi,m}(f, d_i, x)$ for the four teaching dimensions. Candidates are ranked by the probability-weighted utility

$$U(f \mid x) = \sum_{i \in C_b(x)} p_\theta(i \mid x) \cdot \frac{1}{4} \sum_{m=1}^{4} \hat{q}_{\phi,m}(f, d_i, x) - \lambda \, \text{Cost}(f) - \mu \, \text{dist}(f, f_0). \tag{12}$$

In the primary run, $\mathrm{Cost}(f) = 1.0$ and $\mathrm{dist}(f, f_0) = 0$. The expression therefore ranks candidates within a record; it is not itself a safety guarantee.

4.5 Output-side verification and the return-or-abstain gate

The tutor-side verifier records six judgments for each candidate:

$$V(F, d_i) = (\rho_{\mathrm{id}}, \rho_{\mathrm{loc}}, \rho_{\mathrm{guide}}, \rho_{\mathrm{act}}, \rho_{\mathrm{leak}}, \rho_{\mathrm{mislead}}), \tag{13}$$

The first four are teaching dimensions and the last two are binary safety flags. Other risk scores may be logged for analysis, but they do not decide delivery. A candidate is *admissible* if:

$$\mathrm{admissible}(F) \;\equiv\; \rho_{\mathrm{id}} = \rho_{\mathrm{loc}} = \rho_{\mathrm{guide}} = \rho_{\mathrm{act}} = \texttt{Yes} \;\wedge\; \rho_{\mathrm{leak}} = \rho_{\mathrm{mislead}} = \texttt{No}. \tag{14}$$

Component scope and evidential roles. The primary VTG run exercises the hypothesis sets, multiple candidates, the text veto, and the admissibility check. Posterior weighting in equation 12 and the cost term λ did not change within-record rankings, and feedback repair or rollback never ran. We do not use those inactive mechanisms to support the main claim.

Abstention has five possible causes: the diagnosis set is too large for the budget, generation is empty, every candidate fails the text veto, the verifier rejects the selected candidate, or a parser or provider fails. In this single-turn setting, $\perp$ is a safety fallback. It defers help to a later interaction, a reflection prompt, or a human educator, and it is reported explicitly through E_{avail}.

5 Experimental Setup

Data. VTG contains 1,002 mathematics word problems with gold first-error annotations. We split it into 802 training, 100 development, and 100 test records. The test records contain an average of 5.72 student steps and cover multi-step arithmetic, fractions, linear equations, and elementary geometry. The traces include calculation slips, sign inversions, and misunderstandings of operation order. We use BEA 2025 (Kochmar et al., 2025) only to supervise pedagogical quality. It contributes 1,982 training and 247 development responses; VTG remains the evaluation corpus. Details appear in Section A.8.

Conditions. The main comparison uses the same `glm-5.2` model and the same checking procedure for PARS-Tutor, Direct, and CoT. Each condition searches over $N = 5$ or $N = 7$ candidates, uses the text veto and quality ranking, and then calls `feedback_verifier_calibrated`. We draw seven independent samples at temperature 0.7. A separate safety evaluation uses budget 7 and a different routing rule. It makes 505 candidate calls over 95 records whose diagnosis sets fit the budget, and one unconstrained call per record for each always-answer control. We report the two evaluations separately because they do not pass through the same stages or use the same denominators. Standard CoT prompts encourage step-by-step internal reasoning, which can expose a computed answer in the student-facing text. We also include an offline control that reuses the same candidate pool; see Section A.4.

Verification and evaluator variation. The primary verifier receives the gold error diagnosis, as described in Section A.3. PARS and the verifier-scored controls share this input, while baseline generation does not. The resulting leakage rates are therefore conditional on a verifier that knows the diagnosis; they are not human-validated safety guarantees. To test sensitivity to this choice, three gold-blind judges evaluate the same responses without diagnoses or verifier traces. They answer only whether the response gives away the answer, using the problem, reference solution, and student steps. The judges are hosted `gpt-5.6-sol`, local `Qwen3.5-9B`, and `claude-fable-5`.

Statistics. We report 95% Wilson score intervals for proportions, Newcombe hybrid-score intervals for differences, and paired exact McNemar tests for scheduled test-set comparisons and gold-blind subsets. Abstentions and unparseable outputs count as non-passes. The counting conventions are given in Table 4 and the McNemar tests in Table 5.

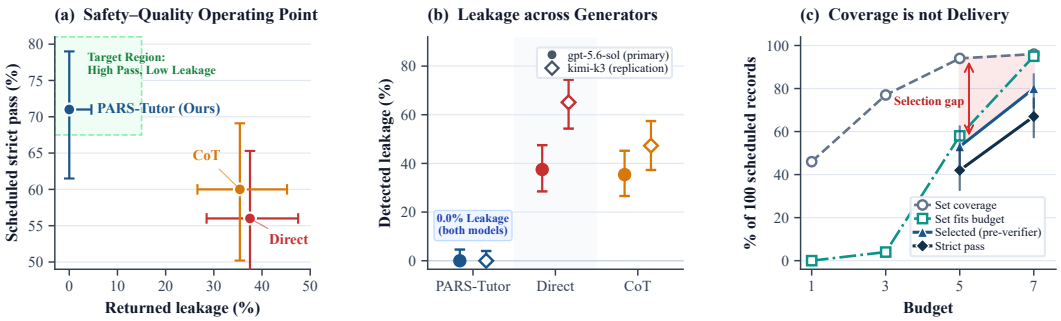

Figure 1: Results overview. **(a)** Safety–quality operating point on the secondary verifier-based evaluation (Table 1): PARS-Tutor (returned, budget 7) sits in the target region of high strict-pass and low leakage, while Direct and CoT (always-answer) trade higher leakage for similar pass rates. **(b)** Detected leakage across generators, comparing the primary `gpt-5.6-sol` run against the `kimi-k3` replication on the same 100 records and the same verifier; PARS-Tutor stays at zero under both generators. **(c)** Coverage is not delivery: the gap between set coverage and strict passes (% of scheduled records) as the budget grows from 1 to 7, illustrating that first-N coverage can overstate delivery by up to 40 points at small budgets.

Table 1: Secondary verifier-based safety evaluation on the frozen VTG test split; reported outcomes include selection, return, strict pass, and detected leakage.

Condition	Sel.	Ret.	Strict	Leak	Misld.	Other	Unjdg.	Strict / sched.
PARS (budget 7), returned	80	**71**	**71**	0/71	0	0	29	**71.0 [61.5, 79.0]**
PARS, pre-gate selected	80	–	–	0/80	7	2	–	–
PARS, `verifier_rejected`	–	–	0	0/9	7	2	–	–
Direct, always answer	n/a	n/a	56	36/96	3	1	4	56.0 [46.2, 65.3]
CoT, always answer	n/a	n/a	60	35/99	3	1	1	60.0 [50.2, 69.1]

6 Experimental Results

6.1 RQ1: Does the gating pipeline reduce detected leakage?

The main RQ1 comparison gives all three conditions the same `glm-5.2` model, search procedure, text veto, and verifier. Its results appear in Table 3. The verifier-based safety evaluation in Table 1 uses a different routing rule and denominator, so we treat it as a secondary view of the safety and availability trade-off. The equal-budget results support the main performance claim.

Table 3 reports the main same-backbone result. At budget 7, PARS-Tutor reaches 64 strict passes out of 100, compared with 60 for Direct and 59 for CoT. Among records that produce a candidate, the rates are 67.4%, 62.1%, and 60.0%. At budget 5, PARS-Tutor reaches 44 scheduled passes but 75.9% on the comparable records. The higher conditional rate comes from abstaining on more uncertain cases. In the secondary evaluation, Table 1 finds no detected leakage among 71 returned PARS responses or the 80 selected candidates that reach verification. Unconstrained Direct and CoT leak in 36 of 96 and 35 of 99 judged outputs. These safety numbers use the routing and counting rules defined for that evaluation.

Qualitative inspection shows why unconstrained generation is risky. For a student who wrote $12 \times 4 = 36$, CoT produced: *"In step 2, 12×4 is 48, not 36. Adding 15 gives 63, so the answer is 63."* This response both corrects the arithmetic and supplies the final answer. PARS-Tutor instead selected: *"Check the multiplication in step 2 and recompute that operation."* The text veto would reject the CoT response because it contains direct numerical operations, and its final sentence can also trigger the reference-number rule. When every candidate contains such numbers, PARS-Tutor rejects the whole set and returns $\perp$ rather than expose a revealing hint.

In the secondary evaluation, paired McNemar tests on scheduled records show 29 strict-pass pairs won by PARS-Tutor against 14 won by Direct, with $p = 0.0315$. Against CoT, PARS-Tutor wins 24 pairs and CoT wins 13, with $p = 0.0989$. For detected leakage, the discordant pairs are 0 for PARS-Tutor versus 36 for Direct, and 0 versus 35 for CoT. Both tests give $p < 10^{-10}$; see Table 5.

Table 2: Budget sweep on the VTG test split: coverage, selection, return, and strict-pass rates.

Budget	Set cov.	Eff. set cov.	Avg. size	Set-fit	Selected	Returned	Strict
1	46%	0%	0.96	0%	n/a	n/a	n/a
3	77%	4%	2.87	4%	n/a	n/a	n/a
5	94%	58%	4.62	58%	53 [43.3, 62.5]	42 [32.8, 51.8]	42 [32.8, 51.8]
7	96%	95%	5.12	95%	80 [71.1, 86.7]	67 [57.3, 75.4]	67 [57.3, 75.4]

Note: This is the equal-budget sweep; its budget-7 row has 67 returned/strict records and is distinct from the secondary safety evaluation in Table 1, which reports 71 scheduled strict passes. Budget-to-budget differences are reported in Section 6.2.

These paired tests summarize outcomes and do not identify a single cause for the equal-budget performance difference.

6.2 RQ2: Availability under different budgets

Table 2 answers RQ2. At budget 5, the retained sets contain 94% of the first errors, but only 53 records produce a candidate and 42 pass all strict checks. Increasing the budget to 7 raises selection by 27.0 percentage points, with an interval from 14.0 to 38.8, and raises strict passes by 25.0 points, with an interval from 11.2 to 37.5. The distinction matters at small budgets. A record can count as covered under first-N accounting even when its full calibrated set is too large to search. Effective coverage treats that record as an abstention, so the 46% set coverage at budget 1 becomes 0% delivered feedback. Coverage alone therefore overstates delivery by about 40 points; Figure 1(c) traces this gap as the budget grows.

6.3 RQ3: What actually drives the decision? Mechanism attribution

The student-side ranker is informative but weakly calibrated. It identifies the first error at top-1 accuracy of 47%, with a 95% interval from 37.5% to 56.7%. The size-matched random-location baseline is 18.1%, a descriptive difference of 28.9 points. The ranker's normalized entropy is 0.948 and its maximum probability is 0.195, close to the uniform value of 0.200. Posterior weighting therefore adds little separation over uniform ranking. The compute-matched control instead points to the text veto. Among 271 candidates, it rejects 87, or 32.1%. Randomly choosing one candidate flags 31.9% of records, and choosing the highest-quality candidate without the veto flags 31.0%. Adding the veto produces no flagged output among the 51 parseable verifier judgments from 53 shield-passing records. Of those 53 records, 42 survive the final admissibility check and become returned responses; five have no valid candidate. The detailed ablation is in Table 6.

Expected quality $\hat{q}$ and shield rejection $1 - H(f, x)$ have almost no correlation: $r = -0.058$ across 271 candidates. In other words, a fluent and pedagogically promising sentence can still reveal a number. Quality ranking alone cannot replace a check on the text that will be shown to the learner.

7 Evaluation Sensitivity

Evaluation choices materially affect the measured outcome. The archived legacy setting reports 95.0% strict passes, or 95 out of 100, for single-pass generation on the same `gpt-5.6-sol` model. The calibrated setting reports 56.0% for Direct, or 56 out of 100, and detects leakage in 36 of 96 judged outputs. The difference is 39.0 percentage points, with an interval from 27.8 to 49.2. Both the prompt and the rubric changed, while the generator (`gpt-5.6-sol`) stayed fixed. The older queries rewarded polite responses but often missed subtle disclosures; the calibrated rubric explicitly asks whether the answer was withheld.

To assess judge sensitivity, three automated gold-blind judges evaluated 60 shared records without gold diagnoses. They were hosted `gpt-5.6-sol`, local `Qwen3.5-9B`, and `claude-fable-5`. Detected leakage for PARS-Tutor was 0/60, 0/60, and 1/60. For Direct it was 31/60, 8/60, and 43/60; for CoT it was 22/60, 9/60, and 28/60. Self-Refine was judged only by the first two models, with 6/60 and 7/60 leaks. The control rates vary widely, from 13.3% to 71.7% for Direct, but all judges keep the PARS rate at zero or near zero. Exact McNemar tests using Qwen give $p = .0039$ to $.0156$,

Table 3: Equal-budget comparison of PARS-Tutor, Direct, and CoT on the frozen VTG test split.

Budget	Condition	Covered	Strict/sched.	Comparable rate	Leak
5	PARS-Tutor	58	44 [34.7, 53.8]	75.9 [63.5, 85.0]	0/45
5	Direct	100	**56 [46.2, 65.3]**	60.3 [47.5, 71.9]	1/57
5	CoT	100	55 [45.2, 64.4]	56.9 [44.1, 68.8]	0/55
7	PARS-Tutor	95	**64 [54.2, 72.7]**	67.4 [57.4, 76.0]	0/64
7	Direct	100	60 [50.2, 69.1]	62.1 [52.1, 71.2]	0/60
7	CoT	100	59 [49.2, 68.1]	60.0 [49.9, 69.3]	0/59

and the hosted judges agree with Cohen's $\kappa = 0.842$; see Section A.6. The shared qualitative pattern is encouraging, but it does not show that zero leakage is intrinsic to PARS-Tutor or rule out evaluator bias. The generator-family replication on `kimi-k3` is reported in Section A.5; the equal-budget analysis of Table 3 in Section A.5.

8 Discussion & Limitations

What the evidence supports. The verifier-based evaluation detects no leakage in 71 returned responses or in the 80 selected candidates that reach verification. Two of the three gold-blind judges also detect none; the Claude Fable pilot flags one of 60 PARS-Tutor outputs. Compute-matched controls indicate that the text veto is a major contributor, while quality ranking alone is insufficient; see Section A.4. On scheduled records, the quality gain is statistically supported against Direct but not CoT.

Implications for deployment. Reasoning-heavy models have been shown to abstain less on unanswerable or ill-posed inputs (Kirichenko et al., 2025), so capability gains can quietly erode a safety behavior that must then be recovered by external structure. Making abstention an explicit, budgeted action turns that structure into a dial: a deployment chooses how much diagnostic diversity to explore and sees the availability cost of that choice, rather than discovering it in user feedback. Because abstention defers rather than resolves, it is most useful when routed somewhere—a reflective prompt back to the learner, or a queue for a human educator who can see why help was withheld (Wang et al., 2024c).

Trade-off and limitations. The budget results are best understood as a policy curve rather than a single ranking: at budget 5, PARS-Tutor returns feedback less often but reaches a higher rate among selected records; at budget 7, it recovers many records that would otherwise be deferred, but each additional candidate still costs a generation call and may end in abstention. A deployment that values immediate assistance may accept this cost; a classroom tool with limited inference capacity may prefer the smaller budget and make the delay explicit to the teacher. The primary verifier sees gold diagnoses, so its leakage rates are conditional on information that would not be available in deployment; the gold-blind evaluations judge only answer leakage. The 39-point gap between the archived and calibrated settings reflects simultaneous changes to the prompt and rubric. The abstention cost should be tuned to support human educators, not to replace them.

9 Conclusion

In this work, we formulate tutoring under diagnostic ambiguity as a selective feedback mechanism with explicit abstention. On the frozen 100-record VTG test split, PARS-Tutor achieves 64 strict passes under a budget of 7, compared to 60 for Direct prompting and 59 for CoT; notably, its margin over CoT is not statistically significant. In the verifier-based safety evaluation on the same split, zero leakage occurs among the 71 returned responses and 80 candidate outputs evaluated by the verifier, a finding corroborated by three gold-blind human judges reporting at most one leak across 60 records. Collectively, these findings demonstrate that output-side verification provides an explicit mechanism to navigate the safety–availability trade-off, rather than evidence of direct pedagogical gains. Consequently, assessing learner repair and knowledge transfer in interactive, multi-turn environments remains the critical next step.

AI USE STATEMENT

Generative AI tools assisted with language editing, code inspection, and formatting during manuscript preparation. The authors independently verified the results, data provenance, and citations and remain responsible for the content.

ETHICS STATEMENT

This study uses existing VTG and BEA data and automated model-based evaluation. It does not recruit human participants. The primary verifier sees gold diagnoses, while separate gold-blind evaluations assess leakage without them. The system is intended to assist teachers rather than replace them.

REPRODUCIBILITY STATEMENT

The Online Methods appendix records the data provenance, splits, prompts, conformal budgets, acceptance thresholds, veto rules, evaluator checks, and counting conventions. The repository contains analysis scripts and frozen standard-output captures. To reproduce every figure, the submission artifact must also include the raw generation and judgment bundles listed in the provenance manifest.

REFERENCES

Anastasios N. Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Stephen Bates, Anastasios Angelopoulos, Lihua Lei, Jitendra Malik, and Michael I. Jordan. Distribution-free, risk-controlling prediction sets. *Journal of the ACM*, 68(6):1–34, 2021.

Michelene T. H. Chi, Nicholas de Leeuw, Mei-Hung Chiu, and Christian LaVancher. Eliciting self-explanations improves understanding. *Cognitive Science*, 18(3):439–477, 1994. doi: 10.1207/s15516709cog1803_3.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Sunishchal Dev, Andrew Sloan, Joshua Kavner, Nicholas Kong, and Morgan Sandler. Judge reliability harness: Stress testing the reliability of LLM judges. In *ICLR 2026 Workshop on Agents in the Wild: Safety, Security, and Beyond*, Rio de Janeiro, Brazil, 4 2026. Workshop paper, not ICLR 2026 main conference.

Aparna Elangovan, Lei Xu, Jongwoo Ko, Mahsa Elyasi, Ling Liu, Sravan Babu Bodapati, and Dan Roth. Beyond correlation: The impact of human uncertainty in measuring the effectiveness of automatic evaluation and LLM-as-a-judge. In *International Conference on Learning Representations (ICLR)*, 2025.

Chen Feng, Minghe Shen, Ananth Balashankar, Carsten Gerner-Beuerle, and Miguel R. D. Rodrigues. Noisy but valid: Robust statistical evaluation of LLMs with imperfect judges. In *International Conference on Learning Representations (ICLR)*, 2026.

Eric Frankel, Shuyue Stella Li, Lillian J. Ratliff, Yulia Tsvetkov, Sewoong Oh, and Pang Wei Koh. Conformal reasoning: Uncertainty estimation in interactive environments. `https://openreview.net/forum?id=Vf5ZUalFk8`, 2025. Preprint.

Yonatan Geifman and Ran El-Yaniv. Selective classification for deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. A survey on LLM-as-a-Judge. *The Innovation*, 7(6): 101253, 2026. doi: 10.1016/j.xinn.2025.101253.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

Hailey Joren, Jianyi Zhang, Chun-Sung Ferng, Da-Cheng Juan, Ankur Taly, and Cyrus Rashtchian. Sufficient context: A new lens on retrieval augmented generation systems. In *International Conference on Learning Representations (ICLR)*, 2025.

Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or escalate: LLM judges with provable guarantees for human agreement. In *International Conference on Learning Representations (ICLR)*, 2025. Oral presentation.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Manu Kapur. Productive failure. *Cognition and Instruction*, 26(3):379–424, 2008. doi: 10.1080/07370000802212669.

Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J. Bell. AbstentionBench: Reasoning LLMs fail on unanswerable questions. *arXiv preprint arXiv:2506.09038*, 2025.

Paul A. Kirschner, John Sweller, and Richard E. Clark. Why minimal guidance during instruction does not work: An analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. *Educational Psychologist*, 41(2):75–86, 2006.

Ekaterina Kochmar, Kaushal Maurya, Kseniia Petukhova, KV Aditya Srivatsa, Anaïs Tack, and Justin Vasselli. Findings of the BEA 2025 shared task on pedagogical ability assessment of AI-powered tutors. In *Proceedings of the 20th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2025)*, pp. 1011–1033, Vienna, Austria, 2025. doi: 10.18653/v1/2025.bea-1.77.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-free predictive inference for regression. *Journal of the American Statistical Association*, 113(523): 1094–1111, 2018.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research (TMLR)*, 2022.

Li Lucy, Albert Zhang, Nathan Anderson, Ryan Knight, and Kyle Lo. The aftermath of DrawEduMath: Vision language models underperform with struggling students and misdiagnose errors. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, pp. 48–62, San Diego, California, USA, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.bea-1.5.

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

Jakub Macina, Nico Daheim, Ido Hakimi, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 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 (EMNLP)*, 2025. Oral presentation.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Christopher Mohri and Tatsunori Hashimoto. Language models with Conformal Factuality guarantees. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Christopher Mohri, Daniel Andor, Eunsol Choi, Michael Collins, Anqi Mao, and Yutao Zhong. Learning to reject with a fixed predictor: Application to decontextualization. In *International Conference on Learning Representations (ICLR)*, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Victor Quach, Adam Fisch, Tal Schuster, Adam Yala, Jae Ho Sohn, Tommi S. Jaakkola, and Regina Barzilay. Conformal language modeling. In *International Conference on Learning Representations (ICLR)*, 2024.

Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. Classification with valid and adaptive coverage. *Electronic Journal of Statistics*, 14(2):4306–4336, 2020.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R. Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R. Johnston, et al. Towards understanding sycophancy in language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Rakshith S Srinivasa, Zora Che, Chen Bo Calvin Zhang, Diego Mares, Ernesto Hernandez, Jayeon Park, Dean Lee, Guillermo Mangialardi, Charmaine Ng, Ed-Yeremai Hernandez-Cardona, Anisha Gunjal, Yunzhong He, Bing Liu, and Chen Xing. TutorBench: A benchmark to assess tutoring capabilities of large language models. *arXiv preprint arXiv:2510.02663*, 2025. An earlier version was submitted to ICLR 2026 (withdrawn); also presented at the ICLR 2026 DATA-FM workshop.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. On the self-verification limitations of large language models on reasoning and planning tasks. In *International Conference on Learning Representations (ICLR)*, 2025.

Anaïs Tack, Ekaterina Kochmar, Zheng Yuan, Serge Bibauw, and Chris Piech. The BEA 2023 shared task on generating AI teacher responses in educational dialogues. In *Proceedings of the 18th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2023)*, 2023. doi: 10.18653/v1/2023.bea-1.64.

Christian Tomani, Kamalika Chaudhuri, Ivan Evtimov, Daniel Cremers, and Mark Ibrahim. Uncertainty-based abstention in LLMs improves safety and reduces hallucinations. *arXiv preprint arXiv:2404.10960*, 2024.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, 2005.

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

Peiyi Wang, Lei Li, Zhihong Shao, R. X. Xu, Damai Dai, Yifei Li, Deli Chen, Y. Wu, and Zhifang Sui. Math-Shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024a.

Qingni Wang, Tiantian Geng, Zhiyuan Wang, Teng Wang, Bo Fu, and Feng Zheng. Sample then identify: A general framework for risk control and assessment in multimodal large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Rose E. Wang, Ana T. Ribeiro, Carly D. Robinson, Susanna Loeb, and Dora Demszky. Tutor CoPilot: A Human-AI approach for scaling real-time expertise. *arXiv preprint arXiv:2410.03017*, 2024b.

Rose E. Wang, Qingcheng Zeng, Lucas Garber, Zhaoyang Wang, Mingyu Yang, Rebecca D. Silverman, Dorottya Demszky, and Noah D. Goodman. 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)*. Association for Computational Linguistics, 2024c.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems (NeurIPS)*, 35, 2022.

Tianjun Wei, Wei Wen, Ruizhi Qiao, Xing Sun, and Jianghong Ma. RocketEval: Efficient automated LLM evaluation via grading checklist. In *International Conference on Learning Representations (ICLR)*, 2025.

David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100, 1976.

Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, YiFan Zhang, Tianlong Xu, Zhendong Chu, Aoxiao Zhong, Kun Wang, Hui Xiong, Philip Yu, Xuming Hu, and Qingsong Wen. ErrorRadar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24297–24334, San Diego, California, USA, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1217.

Tahreem Yasir, Wenbo Li, Sam Gilson, Sutapa Tithi, Xiaoyi Tian, and Tiffany Barnes. Confirming correct, missing the rest: LLM tutoring agents struggle where feedback matters most. In *Proceedings of the 21st Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2026)*, pp. 819–840, San Diego, California, USA, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.bea-1.56.

Fei Yu, Yingru Li, and Benyou Wang. Scaling flaws of verifier-guided search in mathematical reasoning. In *ICLR Workshop on Reasoning and Planning for Large Language Models*, 2025.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Seyed Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations (ICLR)*, 2025.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. PROCESSBENCH: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1009–1024, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.50.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS): Datasets and Benchmarks Track*, 2023.

A Online Methods

This appendix gives the details needed to reproduce the evaluation and then explains the checks that qualify the reported numbers. It starts with how records are counted, describes the text veto and verifier inputs, and ends with controls, blind judgments, and ablations.

A.1 Counting conventions

The primary verifier-based safety evaluation, the equal-budget sweep, and the budget-5 ablation use different routes through the policy. Their counts therefore answer different questions and should not be combined.

Table 4: Evaluation counting conventions.

Quantity	**Size**	**Definition**
Scheduled	100	Every record submitted to the system. Abstentions, provider errors and unparseable responses count as non-passes for scheduled strict performance.
Set-fit / searchable	58 at $b = 5$; 95 at $b = 7$	Equal-budget sweep records whose retained diagnosis set fit the generation budget and for which candidate generation was attempted.
Selected	53 at $b = 5$; 80 at $b = 7$	Sweep records for which a candidate was selected before verifier admissibility; this is a pre-verifier routing outcome.
Budget-sweep returned	42 at $b = 5$; 67 at $b = 7$	Sweep records whose selected candidate had a parseable judgment satisfying the admissibility criterion; provider-error records are not returned.
Verifier-evaluation returned	71 at $b = 7$	Primary safety-evaluation records whose selected candidate passed the verifier admissibility test; this is the reported output set.
Verifier_rejected	11 in the budget-5 ablation; 9 in the budget-7 primary run	Selected records with a parsed candidate that failed the verifier admissibility test; the policy returns $\perp$ for these.
Judged	run-dependent	Selected records with a parseable verifier response; provider and parser failures are reported separately where available.

In the budget-7 primary run, 80 of the 100 scheduled records have a selected candidate and 71 are returned. Nine selected candidates fail the verifier. The other 20 records never reach selection: 15 lose all candidates to the text veto and 5 produce no usable candidate. The separate budget sweep selects 53 records at budget 5 and 80 at budget 7, with 42 and 67 returned or strict passes. It uses a separate `glm-5.2` run to describe the effect of budget, not to tune the test set.

Table 5: Paired McNemar tests for strict pass and detected leakage.

Outcome	**Control**	**Both**	**PARS only**	**Control only**	**Neither**	p
Strict pass	Direct	42	29	14	15	0.0315
Strict pass	CoT	47	24	13	16	0.0989
Detected leakage	Direct	0	0	36	64	2.91×10^{-11}
Detected leakage	CoT	0	0	35	65	5.82×10^{-11}

A.2 The deterministic feedback veto, verbatim

The text veto applies four deterministic rules. It is a practical filter, not a semantic safety evaluator. Feedback is rejected when it is empty, exceeds 100 words, contains an arithmetic expression of the form *number operator number*, where the operator class is the ASCII characters +, −, ⋆, /, x together with the Unicode multiplication and division signs; or if it contains a number that appears in the reference solution but not in the problem statement, in text that also matches one of the conclusion markers *so*, *therefore*, *answer*, *equals* or *is*, matched case-insensitively. The last rule distinguishes a likely leaked conclusion from a number quoted from the problem statement. The arithmetic rule therefore vetoes strings such as "12 × 4" and "48 + 15" even when they are intended as hints.

Two consequences are easy to miss. The arithmetic rule is stricter than the criterion *reveals the answer*. It rejects feedback that quotes the student's own erroneous calculation, such as $105 \div 3 = 35$, even when the purpose is to ask the student to recheck it. The filter therefore favors safer but vaguer feedback and can reduce measured pedagogical quality. It also has no semantic understanding of leakage: prose that describes the solution path without numerals can pass.

On a synthetic set designed to leak the reference answer, the filter recovers 92% of cases with the arithmetic rule and 95% with the combined rules. It has a 0% false-positive rate on feedback that quotes only numbers from the problem statement. These are run-specific diagnostics on a constructed probe set; they do not estimate semantic leakage detection in naturally generated prose.

The budget-5 exploratory run made 272 candidate calls. Of these, 271 returned non-empty feedback across 58 records whose diagnosis sets fit the budget. The veto rejected 87 of the 271 texts, or 32.1%, with interval $[26.8, 37.9]$; 184 passed. The rule breakdown and synthetic recall are diagnostics for this run, not estimates of semantic leakage detection.

A.3 Gold exposure of the primary verifier

The primary verifier prompt includes the record field *Known student error diagnosis*, which contains the gold error annotation. In the budget-5 candidate-pool evaluation, this field appears in 53 of 53 PARS candidate-pool jobs and in 58 of 58 budget-5 single-pass jobs. None of the 330 generator jobs contains it. The exposure is specific to the primary verifier and limits how its absolute rates should be interpreted. Gold-blind evaluations remove the field when judging leakage, but they do not establish autonomous diagnosis or the full teaching criterion. The blind re-judging is described in Section A.6.

A.4 Compute-matched pool control

The primary verifier comparison is not compute-matched. We therefore run an offline control on the frozen budget-5 candidate pool. It contains 272 calls, 271 non-empty texts, and 58 records whose sets fit the budget. Every arm uses the same generated texts and candidate count; only selection and the text veto differ. Random-1 samples one candidate per record without a veto. Quality-argmax chooses the highest-quality candidate without a veto. Quality-argmax+shield chooses the highest-quality candidate among texts that pass the veto. The budget-5 PARS result is included as the verifier-plus-veto reference with 53 selected records.

The veto rejects 87 of 271 candidate texts, or 32.1%, with 95% Wilson interval $[26.8, 37.9]$. Quality and rejection have point-biserial correlation $r = -0.058$. At the record level, Random-1 flags 31.9% with interval $[22.4, 41.4]$ in a 2,000-draw simulation. Quality-argmax without a veto flags 18 of 58 records, or 31.0%. Adding the veto produces no flagged output among the 53 records with a passing candidate. Five records have no passing candidate. These results identify the veto as the source of the reduction on this deterministic proxy, rather than quality selection alone. They do not estimate semantic leakage: the proxy focuses on numbers and can both over-reject and miss prose leakage.

A.5 Same-pipeline equal-budget baseline control

For reproducibility, both baselines are generated with `glm-5.2` at temperature 0.7, using seven independent samples per record. Each arm then uses the same text veto, probability-weighted quality score, and `feedback_verifier_calibrated` test as PARS-Tutor. All three conditions are re-verified in one batch, so their verdicts share a verifier run. Candidate preparation is in `prepare_equal_budget_candidates.py`; aggregation is in `summarize_equal_budget.py`; and the live runner is `run_baseline_equal_budget_live.sh`. The older dry-run scaffold remains only as an implementation record. Results appear in Table 3.

Analysis of the equal-budget results. The main comparison keeps the model and output checks fixed. PARS-Tutor, Direct, and CoT use `glm-5.2`, best-of-N search, the text veto, and `feedback_verifier_calibrated` with seven samples per record at temperature 0.7. On the 100-record schedule, PARS-Tutor trails at budget 5, with 44 passes versus 56 for Direct and 55 for CoT, because it abstains more often. At budget 7 it leads in absolute terms, with 64 passes versus 60 and 59, but the paired McNemar test gives $p = 0.0315$ against Direct and $p = 0.0989$ against CoT,

so the lead over CoT is not statistically supported. Among candidate-producing records, its budget-5 strict-pass rate is 75.9%, versus 60.3% for Direct and 56.9% for CoT. The difference from CoT at budget 5 is 19.0 points, with a Newcombe interval from 1.8 to 34.7. At budget 7 the comparable rates are 67.4%, 62.1%, and 60.0%, with Newcombe intervals of 5.3 points [−8.2, 18.5] against Direct and 7.4 points [−6.2, 20.6] against CoT; both intervals straddle zero and are consistent with the McNemar result that the budget-7 lead is not significant against CoT. These are the backbone-matched performance results; Table 1 gives the secondary safety view under a different routing rule.

Generator-family replication. To test whether the safety result holds across generator families, we re-ran the secondary verifier-based evaluation on the same 100-record test split with `kimi-k3` as the generator, holding the `gpt-5.6-sol` verifier fixed. On 92 judged outputs per condition, detected leakage for PARS-Tutor remains 0/92 [0.0, 4.0]; for Direct it rises to 54/83 (65.1% [54.3, 74.4]), a Newcombe interval of +27.6pp [6.9, 46.0] over the `gpt-5.6-sol` primary; for CoT it rises to 43/92 (46.7% [36.9, 56.9]), an interval of +11.4pp [−8.3, 30.2]. The PARS-Tutor rate stays at the floor under a different generator, so the safety gap is not an artifact of the specific `gpt-5.6-sol` generation pipeline; see Figure 1(b).

Where future improvements are most likely to matter. The diagnosis ranker is useful for keeping alternatives alive, but its near-uniform probabilities mean that better ranking alone may have a limited effect. The largest observed reduction in the numerical leakage proxy comes from checking the candidate text before delivery. That check is deliberately conservative: it blocks some useful prompts because it cannot tell whether a number is being quoted for reflection or supplied as a solution. A more precise safeguard could inspect the relation between the number and the student's work, or ask a second model to identify whether the learner still has a meaningful next step. Such a safeguard should be evaluated on the same scheduled denominators, since reducing false positives is valuable only if the resulting feedback remains safe.

Connecting abstention to a recovery action. In a deployed tutor, abstention should also be connected to a recovery action. The system could ask the learner to restate the disputed step, offer a non-numerical request to explain the reasoning, or notify a teacher when uncertainty persists. These actions would turn a silent refusal into another teaching opportunity, but they require a multi-turn evaluation that is outside the present study. Our single-turn results therefore support a narrow claim: explicit withholding and output checking make the safety–availability choice measurable. They do not establish that the policy improves learning, that the current veto is optimal, or that the same trade-off will hold for subjects beyond mathematics.

What a safe response still needs to accomplish. Avoiding the answer is only the first requirement for useful tutoring. The response must also leave the learner with a concrete next action without completing the reasoning on the learner's behalf. Our four teaching checks aim to enforce this balance, but they remain model-based judgments. Future studies should test whether learners can repair the original step, explain the repair, and transfer it to a new problem. These outcomes measure educational value rather than safety at delivery alone.

A.6 GOLD-BLIND EVALUATION

We prepared 240 gold-blind items from the 60 records on which PARS-Tutor, Direct, CoT, and Self-Refine all produce feedback. Each item contains the problem, reference solution, student steps, and one candidate. It omits the gold diagnosis and the primary verifier's output. Two judges answer the same binary question in JSON: does the feedback give away the answer? The judges are hosted gpt-5.6-sol and local Qwen3.5-9B. The reference solution is shown because leakage cannot be judged without knowing the answer.

The comparison is paired across conditions. We report Wilson intervals and exact McNemar tests on shared records. Under gpt-5.6-sol, detected leakage is 0/60 for PARS-Tutor, 31/60 for Direct, 22/60 for CoT, and 6/60 for Self-Refine. Under Qwen3.5-9B, the counts are 0/60, 8/60, 9/60, and 7/60. Against PARS-Tutor, Qwen gives exact McNemar p-values of 0.0078, 0.0039, and 0.0156 for Direct, CoT, and Self-Refine. The zero PARS count and the direction of the Direct and CoT contrasts agree across all three judges. Self-Refine is evaluated only by the first two judges and is absent from the Claude Fable pilot. The input bundle, raw judgments, and scripts are retained as reproducibility

artifacts. An independent Claude Fable 5 pilot on the same 60 records flags 1/60 PARS-Tutor, 43/60 Direct, and 28/60 CoT outputs. The 17 GPT labels recovered by retry are excluded from the agreement comparison so that Claude and GPT see the same inputs; on that pairing, agreement is 59/60 for PARS-Tutor, 35/43 for Direct, and 42/43 for CoT, with pooled Cohen's $\kappa = 0.842$ over 146 paired judgments.

A.7 Component ablations

Table 6 removes one component at a time on the held-out test split. We do not ablate conformal calibration or rollback. Removing calibration changes the diagnosis sets and would require a new generation run; rollback never runs on this split and cannot be identified. The remaining rows rerun selection on the existing candidate pool with one named component disabled. Every variant uses the same `feedback_verifier_calibrated` verifier, gpt-5.6-sol, so the comparison isolates the selection differences.

Table 6: PARS-Tutor component ablations at budget 5.

Variant	Sel	Valid	Returned	Strict / returned	Leak / valid	Mlead / valid
Full PARS-Tutor	53	51	42	42/42	0/51 [0.0, 7.0]	8/51 [8.2, 28.0]
w/o Deterministic shield	58	57	51	51/51	1/57 [0.3, 9.3]	4/57 [2.8, 16.7]
w/o Posterior marginalization	53	53	46	46/46	0/53 [0.0, 6.8]	6/53 [5.3, 22.6]
w/o Quality model	53	53	45	45/45	0/53 [0.0, 6.8]	8/53 [7.9, 27.1]

verifier_rejected bucket: 11 for full, 7 for no-shield because more candidates survive the shield, 7 for no-posterior, 8 for no-quality.

The no-shield variant selects 58 records, five more than the full system. Those five are exactly the records blocked by the veto before selection. The full system then returns 42 of its 53 selected records because 11 fail the verifier; this is the verifier's additional contribution after the text check. The Valid column separates parseable provider outcomes from returned responses. Removing the quality model or posterior weighting does not change the leak count, but it changes strict passes by changing which candidate is selected. These are conditional ablations, not delivery rates over all scheduled records.

A.8 Data provenance

VTG contains 1,002 mathematics word problems with gold first-error annotations and is split 802/100/100 into train, development, and test sets. It is the evaluation corpus. BEA 2025 supplies 1,982 training and 247 development tutor responses for the four pedagogical dimensions. BEA is used for supervision and not as a test corpus.