AI Assistants Overassist

Verona Teo[1*] **Raghav Jain**[2*] **Tobias Gerstenberg**[1] **Max Kleiman-Weiner**[3]

[1]Stanford University
[2]University of California, San Diego
[3]University of Washington

Abstract

Large language models (LLMs) are increasingly used as tutors and thought partners, helping users reason through problems. While guidance from AI assistants can scaffold thinking and foster learning, such benefits depend on *how* they help—for instance, intervening too early or too frequently may hinder true learning and cognitive engagement. Yet how AI systems navigate intervention decisions during problem-solving remains poorly understood. Here, we introduce INT-BENCH, a simulation-based benchmark for evaluating LLM interventions during learning. INT-BENCH simulates a "student" solving a problem while a "teacher" monitors the student's reasoning and decides whether, when, and how to intervene. Across three domains—code debugging, mathematics, and brain teasers—we evaluate LLM teachers on the frequency and timing of interventions, as well as their impact on both immediate task success and generalization to new problems. We also compare LLMs to humans, finding that LLMs intervene more frequently and earlier than humans. Moreover, in contrast to humans, they tend to provide complete solutions rather than targeted hints. These findings suggest that current LLM assistants often optimize for short-term success rather than supporting the reasoning processes needed for deeper learning and long-term success.

1 Introduction

Large language models (LLMs) are being adopted in a wide range of educational and professional workflows (Chatterji et al., 2025; Peng et al., 2023). In these settings, the value of such assistance depends not only on whether the model can solve the task, but also on *how* it supports the user's reasoning along the way. Effective help requires deciding both when to intervene and how much information to provide, as well as when to stay silent and let a learner reason on their own. Intervening too early or too directly can take over the reasoning process, while waiting too long may leave users stuck (Mclaren et al., 2014; Soderstrom and Bjork, 2015). This reflects a trade-off in learning environments, where teachers and parents balance the benefits of intervention and efficient task completion against the longer-term value of productive struggle and perseverance (Campbell et al., 2025; Shachnai et al., 2025).

Recent work suggests that while AI assistance can improve immediate task performance, it can negatively affect cognitive engagement, motivation, and learning (Faas et al., 2024; He et al., 2025; Shaw and Nave, 2026). However, these studies primarily measure downstream outcomes of assistance. Less is known about the assistance behavior that may give rise to these detrimental effects: when LLMs choose to intervene, how early they step in, how much information they provide, and whether their feedback supports reasoning or simply moves the user toward the answer.

[*]Equal contribution

Preprint.

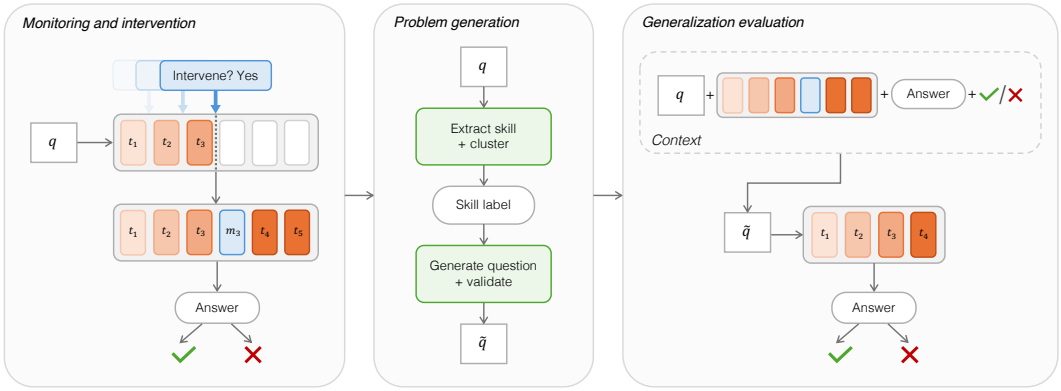

Figure 1: **Overview of INT-BENCH.** For a question q, the student produces a baseline reasoning trace $t = (t_1, t_2, \ldots, t_T)$. If the teacher decides to intervene at step k, an intervention message m_k is injected into the reasoning trace. The student then updates its reasoning and provides an answer, which a judge evaluates for correctness. A structured generation module produces a related new problem $\tilde{q}$ based on the original question q's skills. For this generalization task, the student receives the original problem, reasoning trace, message, answer, and correctness verdict, which serves as the context when attempting the new question $\tilde{q}$.

Motivated by this gap, we study LLM assistance in a controlled, simulated student-teacher setting, allowing us to characterize LLM intervention behavior and compare it with human assistance strategies. We address the following research questions:

- **RQ1:** How frequently do LLMs intervene, and at what point in the problem-solving process do these interventions occur?
- **RQ2:** To what extent do LLM interventions improve (or hurt) immediate task success?
- **RQ3:** What is the effect of LLM interventions on the student's ability to generalize to related, unseen problems?
- **RQ4:** How does the intervention behavior of LLMs differ from that of humans in similar settings?

Our work makes three main contributions. First, we formalize LLM assistance as a sequential intervention game and introduce INT-BENCH (Figure 1), a simulation-based benchmark where a teacher LLM monitors a student's reasoning trace and decides whether, when, and how to intervene. Second, we develop metrics for characterizing assistance behavior, including intervention frequency, timing, immediate helpfulness, and generalization to new, related problems. We use these metrics in a large-scale empirical analysis across multiple models and domains. Third, we conduct a human study where participants act as teachers in similar settings, enabling direct comparisons between LLMs and humans. Overall, we find that LLMs intervene more frequently and earlier than humans, often providing overly informative feedback that mostly gives the solution away. Furthermore, these interventions tend to be highly problem-specific, limiting the student's ability to generalize and apply what was learned to new problems.

2 Related Work

Impacts of AI on learning AI assistance can improve immediate performance while reducing cognitive engagement and autonomy (Chen et al., 2025; Faas et al., 2024; Kosmyna et al., 2025; Stadler et al., 2024). More broadly, delegating mental work to external tools can erode independent analytical capacity over time (Liu et al., 2026; Risko and Gilbert, 2016). Making learning effortful, even at the cost of initial performance, can improve long-term retention and transfer (Bjork, 1994; Gajos and Mamykina, 2022; Kapur, 2014). These studies primarily focus on the downstream *outcomes* of assistance; our work additionally characterizes the assistance behavior that produces them, including when models intervene, how much they reveal, and whether their feedback supports reasoning or substitutes for it.

Timing and boundaries of AI assistance The tension between providing support and withholding assistance to promote learning—often known as the *assistance dilemma*—is a core challenge in educational science (Koedinger and Aleven, 2007; Maniktala et al., 2020; Mclaren et al., 2008, 2014). Recent work has begun operationalizing similar trade-offs in human-AI collaboration. In particular, several systems have focused on learning *when* an agent should speak or stay silent (Manzoor et al., 2025; Patel et al., 2026; Steyvers and Mayer, 2025). However, these approaches typically operate at the level of discrete decision instances or dialogue turns. We instead formalize assistance as sequential monitoring of a reasoning trace, allowing for finer-grained control over intervention timing within a single problem, and evaluate whether interventions promote learning transfer.

Simulated students and teachers Prior work has explored LLMs' abilities to provide feedback and produce appropriate teacher responses (Macina et al., 2023; Matelsky et al., 2023; Tack and Piech, 2022). LLM-simulated students have been used to study learning processes (Ross and Andreas, 2025; Ross et al., 2025) and to support teacher training (Abbasiantaeb et al., 2023; Daheim et al., 2024; Hu et al., 2025; Jin et al., 2025; Liu et al., 2024; Lu and Wang, 2024; Markel et al., 2023). We build on this line of work by simulating both students and teachers using LLMs in an intervention game.

3 The INT-BENCH Framework

3.1 Problem Setup

We formalize the interaction between a student and a teacher as a sequential intervention game (Figure 1). Let $\mathcal{Q} = \{q_1, \ldots, q_N\}$ be a set of N question instances. Each episode of the game involves four components: a problem $q_i \in \mathcal{Q}$, a student model, a teacher model, and a judge. An episode proceeds in two phases.

In the *baseline* phase, the student first solves q without assistance, generating a baseline reasoning trajectory $t = (t_1, t_2, \ldots, t_T)$ and a final answer $\hat{y}$, which is evaluated by the judge against a reference solution y^*. In the *monitoring* phase, the baseline reasoning trace is revealed to the teacher, who decides whether to intervene. At each step $k < T$, the teacher observes the reasoning prefix $t_{1:k} = (t_1, \ldots, t_k)$ and selects an action $a_k \in \{\text{wait}, \text{intervene}\}$. A teacher policy π maps the observed prefix to an action and, if intervening, a message: $\pi(t_{1:k}) = (a_k, m_k)$. If $a_k = \text{intervene}$, the message m_k is immediately injected into the trace after $t_{1:k}$, the monitoring phase ends, and the student updates its reasoning given the intervention.

3.2 Monitoring

We compare teacher behavior during the monitoring phase under two conditions. In the *Standard* condition, the baseline trajectory is revealed to the teacher in cumulative increments of a fixed size s (e.g., characters). At each step k, the teacher observes the reasoning prefix $t_{1:k}$ and chooses between waiting (reveals next increment) or intervening (generates an intervention message m_k and monitoring terminates). The teacher may intervene at most once.

We also evaluate an *Oracle* condition, in which the teacher receives the full baseline trace t, the student's final answer $\hat{y}$, and the correctness verdict $\mathbb{1}[\hat{y} = y^*]$ simultaneously, prior to making any decision. The teacher decides whether to intervene, and if so, selects the optimal point k at which to intervene post hoc.

We define two behavioral metrics:

Intervention Frequency The proportion of episodes where the teacher assists, $\phi = \frac{1}{N}\sum_{i=1}^{N} I_i$, where $I_i \in \{0, 1\}$ indicates an intervention on problem q_i. We additionally compute conditional frequencies $\phi_{\text{correct}} = \mathbb{P}(I_i = 1 \mid \mathbb{1}[\hat{y}_i = y_i^*] = 1)$ and $\phi_{\text{incorrect}} = \mathbb{P}(I_i = 1 \mid \mathbb{1}[\hat{y}_i = y_i^*] = 0)$, which capture how often teachers intervene when the student would have succeeded or failed without assistance.

Intervention Timing We define the *absolute timing* τ_{abs} as the length of the reasoning trace (e.g., number of characters) revealed before the intervention. The *relative timing* τ_{rel} is the normalized point of intervention, defined as $\tau_{\text{rel}} = \tau_{\text{abs}}/L \in [0, 1]$, where L is the total length of the baseline trajectory. Values near 0 indicate early intervention.

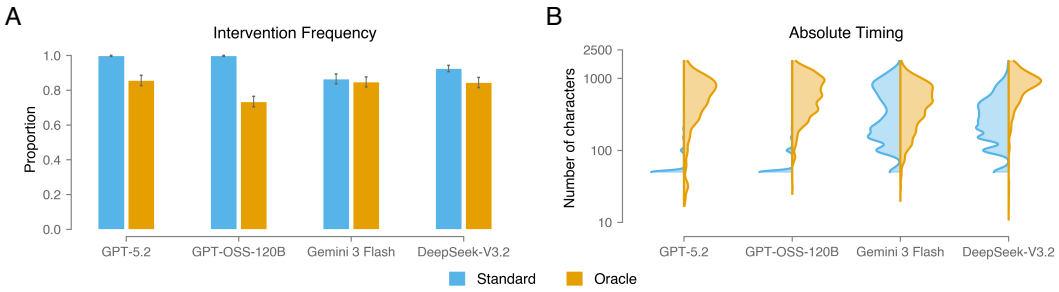

Figure 2: **Intervention frequency and absolute timing for each model across all three domains.** **(A)** Across all models, teachers in the *Standard* condition intervened more often than *Oracle* teachers. **(B)** Number of characters that teachers chose to reveal from the reasoning trace prior to intervening. *Standard* teachers received 50-character increments at a time. Error bars are 95% bootstrapped CIs.

3.3 Post-Intervention Reasoning

Following an intervention at step k, we construct an updated context by truncating the baseline reasoning trace to $t_{1:k}$ and appending the teacher's intervention message m_k. The student then attempts to solve the problem using one of three update strategies:

1. *Standard-Continue*: The student receives feedback from the *Standard* teacher and generates a revised reasoning trajectory t' and a final answer $\hat{y}'$ conditioned on the updated context.
2. *Oracle-Continue*: The student receives feedback from the *Oracle* teacher and similarly generates a revised trajectory t' and a final answer $\hat{y}'$.
3. *Stop-and-Answer*: The student receives feedback from the *Standard* teacher, but is prevented from generating any further reasoning steps, and must output a final answer $\hat{y}'$ immediately.

To quantify the impact of these strategies on task performance, we measure the average signed change in answer correctness following an intervention across all intervened episodes.

Immediate Helpfulness Let $\mathcal{I}$ denote the set of episodes where the teacher intervenes. For each $i \in \mathcal{I}$, we define $H_i = \mathbb{1}[\hat{y}'_i = y^*_i] - \mathbb{1}[\hat{y}_i = y^*_i] \in \{-1, 0, 1\}$, where $\hat{y}_i$ and $\hat{y}'_i$ are the baseline and post-intervention answers, respectively. The overall immediate helpfulness score $H = \frac{1}{|\mathcal{I}|} \sum_{i \in \mathcal{I}} H_i$ reflects the net accuracy gain across intervened episodes.

3.4 Problem Generation

To understand whether interventions promote transfer to new problems, we evaluate students on related but distinct problems that require the same underlying skill as the original questions. We use a structured four-step pipeline, adapted from Didolkar et al. (2024), to generate variants $\tilde{q}$ for each reference problem q. First, an LLM extracts a fine-grained skill label from q (e.g., "modular arithmetic"). Second, these labels are clustered into broader skill categories. Third, a generator LLM produces candidate variant problems conditioned on q and its skill category, ensuring they require the same underlying skill to solve, but differ in surface form, context, and parameters. Finally, a validation step filters out candidates that are inconsistent with the target skill category or contain incorrect reference solutions. A single valid variant $\tilde{q}$ is sampled for the generalization evaluation.

3.5 Generalization Evaluation

We evaluate the student's performance on the variant problem $\tilde{q}$ under three context conditions to isolate what aspects of prior experience drive generalization ability:

1. *No-Context*: The student attempts to solve $\tilde{q}$ from scratch without any information about the original problem q. This serves as a generalization baseline.
2. *Problem-Context*: The student receives the original problem q, its unassisted baseline reasoning trace t, the final answer $\hat{y}$, and the judge's correctness verdict. This measures whether exposure to a related question and its solution alone helps the student solve the new problem.

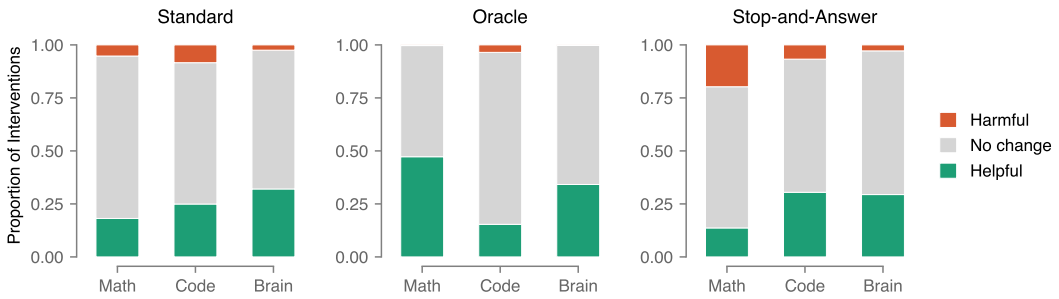

Figure 3: **Proportion of interventions that were helpful, harmful, or made no difference to the correctness of answers**. Bars are averaged over all teacher models. Helpful interventions are those in which a student's originally incorrect answer became correct after intervention, while harmful interventions are those in which an initially correct answer became incorrect.

3. *Intervention-Context*: The student receives the full assisted episode for q (from the *Standard* monitoring condition), including the reasoning prefix $t_{1:k}$, the teacher's intervention message m_k, the post-intervention trace t', the updated answer $\hat{y}'$, and the judge's correctness verdict. This measures the value added by the intervention over simple problem exposure.

In all context conditions, the judge evaluates the student's answer to $\tilde{q}$ against its reference solution. To isolate the transfer value of these contexts, we define a generalization metric.

Generalization Helpfulness The change in correctness on the variant $\tilde{q}_i$ relative to the *No-Context* baseline. For each reference-variant pair, $G_i = \mathbb{1}[\hat{\tilde{y}}_i^{\mathrm{C}} = \tilde{y}_i^*] - \mathbb{1}[\hat{\tilde{y}}_i^{\mathrm{NC}} = \tilde{y}_i^*]$, where $\hat{\tilde{y}}_i^{\mathrm{C}}$ and $\hat{\tilde{y}}_i^{\mathrm{NC}}$ are the student's answers with (*Problem-Context* or *Intervention-Context*) and without context, respectively. The overall generalization helpfulness score $G = \frac{1}{M}\sum_{i=1}^{M} G_i$ is the average over all M reference-variant pairs.

4 Experiments

4.1 Simulation Setup

Datasets We evaluate our framework on 1,500 problems across three domains:

- **Code Debugging**: We sampled 500 code debugging problems from the DebugEval dataset (Yang et al., 2025), which includes buggy code snippets and corresponding solutions.
- **Mathematics**: We used the MATH-500 dataset (Hendrycks et al., 2021), a subset of the MATH benchmark designed to evaluate mathematical reasoning.
- **Brain Teasers**: We scraped problems from the Braingle website,[1] an online platform of lateral thinking puzzles. We included problems from four different categories ("Riddle," "Language," "Rebus," and "Group") and sampled the 500 most popular and easiest problems (Appendix A.1).

For the generalization evaluation, we first sampled 150 problems per domain to generate variant questions via the pipeline described in §3.4. We then subsampled 100 successfully validated reference-variant pairs per domain (300 total).

Models We used Qwen2.5-7B-Instruct as the student model, which was prompted to show its reasoning step-by-step. We selected this model because it exhibits intermediate baseline performance on each of our selected domains, providing sufficient headroom to observe learning from teacher interventions. For teacher models, we evaluated two closed-source models (GPT-5.2, Gemini 3 Flash) and two open-source models (GPT-OSS-120B, DeepSeek-V3.2). We ran each model three times per question. We used GPT-5.2 as the judge model for evaluating student answers. All student and teacher models were run with temperature 0.7, and the judge model was run with temperature 0.

[1] https://www.braingle.com

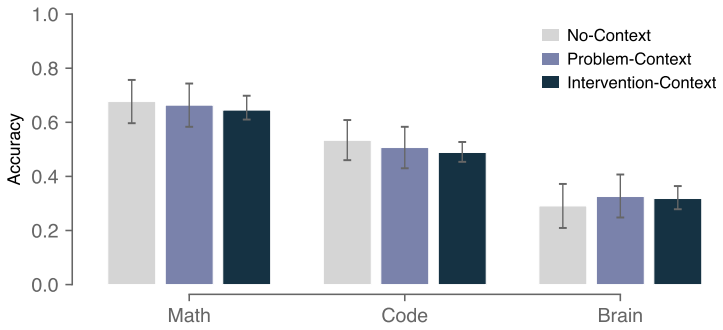

Figure 4: **Proportion of problems answered correctly on the variant questions.** Bars show the student accuracy under the *No-Context*, *Problem-Context*, and *Intervention-Context* conditions across the math, code debugging, and brain teaser domains. *Intervention-Context* is averaged over the four teacher models. Error bars represent 95% bootstrapped CIs.

For each experiment and the results that follow, we set a fixed increment size of $s = 50$ characters (i.e., teachers in the *Standard* monitoring condition received 50-character increments). [2]

4.2 Human Studies

To compare LLM-based interventions with human behavior, we conducted a study with two conditions—*Standard* and *Oracle*—where human participants acted as teachers monitoring simulated student reasoning. We manually selected 30 problems from the brain teaser dataset and used the same reasoning traces generated by our student model, allowing us to make direct comparisons between human and LLM teachers.

We chose brain teasers because they require minimal domain expertise. Unlike math and code debugging, in which teachers need specialized conceptual or technical knowledge to evaluate student reasoning, brain teasers primarily rely on domain-general skills, such as ordinary language understanding and cognitive reflection. This ensures that variation in participant behavior reflects differences in intervention strategy rather than domain expertise.

We recruited 25 participants for each condition via Prolific, for a total of $N = 50$ participants. Each participant completed 6 trials (episodes), which were randomly ordered. In the *Standard* condition, participants could reveal the reasoning trace in 50-character increments and could intervene at any point by providing a written message. If they reached the end of the trace without intervening, they could either provide feedback or proceed to the next trial. In the *Oracle* condition, participants were asked whether and when they would intervene only once they had read the entire reasoning trace.

5 Results

We organize our results around our four research questions. We find that LLMs intervene frequently and early (§5.1), interventions are moderately helpful for immediate task success (§5.2), interventions do not reliably improve generalization (§5.3), and LLM intervention behavior differs both quantitatively and qualitatively from that of humans (§5.4).

Student baseline accuracy Across all three domains, the student achieved a baseline accuracy of 43% prior to interventions: 70.4% on math, 45.2% on code debugging, and 14.4% on brain teasers.

5.1 How often and when do LLM teachers intervene? (RQ1)

LLMs intervene frequently and early, even when the student would have gotten the correct answer. Figure 2 shows the overall intervention frequency and absolute timing for each model

[2] We also ran experiments with different increment sizes (e.g., fixed 300-character increments, increments at the sentence level), student models (e.g., Qwen3-32B, LLaMA-3.1-8B), and prompt variations (e.g., "only intervene if truly necessary"). We found that these variations resulted in qualitatively similar results; full details are provided in Appendix E.

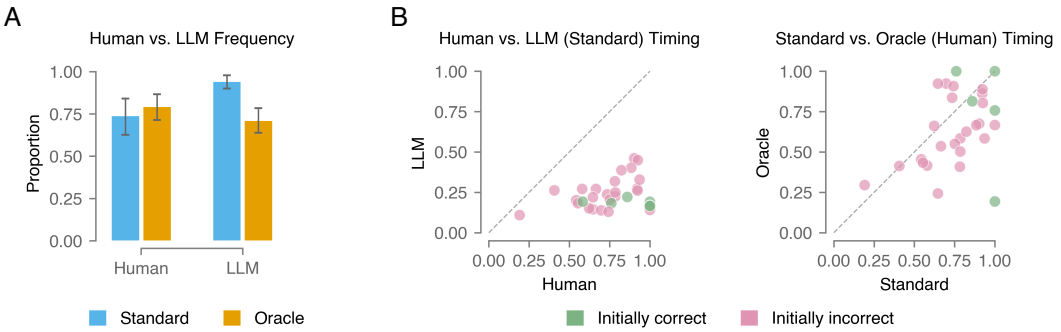

Figure 5: **Human vs. LLM intervention behavior on brain teasers. (A)** Intervention frequency for human and LLM teachers under the *Standard* and *Oracle* conditions. Error bars are bootstrapped 95% CIs. **(B)** Per-question relative timing for humans vs. LLMs in the *Standard* condition (left) and human relative timing under the *Standard* vs. *Oracle* conditions (right). Humans intervene later than LLMs, including for questions the student would have gotten right without intervention (green dots).

and domain. We found that in the *Standard* monitoring condition, most models intervened very frequently and early in the reasoning process ($\phi = 0.90$, $\tau_{\text{rel}} = 0.18$). GPT-5.2, GPT-OSS-120B, and DeepSeek-V3.2 intervened in over 90% of trials across the three domains, with GPT-5.2 and GPT-OSS-120B teachers intervening even before the second increment (100 characters) had been shown. Furthermore, with the exception of Gemini 3 Flash, *Standard* teachers often intervened even when the student would have gotten the answer correct. For instance, GPT-5.2 and GPT-OSS-120B intervened on questions initially correctly solved 98% and 100% of the time, respectively.

Teachers intervene less often when they have full information. In the *Oracle* condition, models intervened significantly less frequently compared to *Standard* teachers (*Standard*: $\phi = 0.90$, 95% confidence interval (CI) $[0.89, 0.90]$; *Oracle*: $\phi = 0.54$, 95% CI $[0.52, 0.56]$; $p < .001$). They also intervened significantly later in the reasoning trace (*Standard*: $\tau_{\text{rel}} = 0.18$, 95% CI $[0.17, 0.18]$; *Oracle*: $\tau_{\text{rel}} = 0.56$, 95% CI $[0.55, 0.57]$; $p < .001$). Here, the models almost never intervened when the student would have arrived at the correct answer. In cases where the student would have succeeded on their own, models intervened less than 3% of the time, indicating that student correctness is a primary factor for their intervention decisions. Compared to other models, Gemini 3 Flash had the smallest difference in the number of characters shown prior to intervening between the *Standard* and *Oracle* conditions.

5.2 How helpful are the interventions? (RQ2)

Oracle interventions are most helpful (and least harmful). Across the domains and models, *Standard* teacher interventions were moderately helpful, resulting in a net accuracy gain of $H = 0.20$, with 25.5% initially incorrect answers becoming correct and 5.4% becoming incorrect post-intervention (Figure 3). The interventions from the *Oracle* condition were the most helpful overall ($H = 0.30$), particularly in math, and also almost never hurt student performance—the rate at which they caused a correct student to become incorrect was only 1.2% across all domains. So, when models have access to the full reasoning and correctness verdict, they can intervene more effectively.

Interventions are overly informative. To understand the extent to which interventions improved student reasoning, we compared against a *Stop-and-Answer* ablated baseline. In the math domain, we observed that forcing the student to answer without further reasoning after the intervention reduced immediate helpfulness overall compared to the *Standard-Continue* and *Oracle-Continue* conditions. This suggests that, at least for math, the value of the intervention lies partly in guiding subsequent reasoning; simply stopping the student is not as effective as allowing them to incorporate the feedback.

In contrast, for code debugging problems, the *Stop-and-Answer* condition actually *improved* immediate helpfulness over the *Standard-Continue* condition ($H = 0.24$ vs. 0.17), and for brain teasers, the two conditions resulted in similar performance ($H = 0.27$ vs. 0.30). One possible explanation is that teachers tend to provide long, detailed interventions that often either directly reveal the solution or make the solution easy to infer. Such interventions mean that students do not need to continue

Table 1: **Distribution of intervention categories on the 30 brain teaser questions.** Values are the percentage of interventions assigned to each category (each intervention has a single category label) for humans and LLMs under both monitoring conditions. Subtable (a) characterizes the functional role of the intervention, and (b) measures the extent to which the intervention leaks the solution. Column values sum to 100% within each subtable.

(a) Functional Role

	Human		LLM	
Category	Standard	Oracle	Standard	Oracle
Redirecting away from an unproductive path	29.7	25.2	20.9	34.8
Reframing the problem representation	15.3	16.0	24.8	23.0
Correcting local errors	16.2	10.1	2.9	10.5
Refocusing attention on salient evidence	14.4	10.9	15.9	10.2
Connecting partial insights into a coherent whole	2.7	3.4	8.3	3.5
Narrowing the search space	4.5	1.7	7.1	3.1
Prompting pattern recognition	5.4	6.7	7.1	5.1
Validating productive reasoning	4.5	6.7	0.0	0.0
Encouraging verification and self-checking	1.8	5.9	0.0	2.0
Extending a correct partial solution	0.9	5.9	2.7	0.4
Clarifying rules and constraints	1.8	0.8	3.5	4.3
Modeling a systematic strategy	0.0	1.7	3.8	0.8
Providing targeted hints or candidate answers	0.9	3.4	1.8	2.0
Providing closure or final confirmation	0.9	1.7	0.0	0.0
Explaining why an answer fits	0.0	0.0	1.2	0.4
Supporting persistence and reducing frustration	0.9	0.0	0.0	0.0

(b) Solution Leakage

	Human		LLM	
Category	Standard	Oracle	Standard	Oracle
Full solution revealed	5.4	6.7	14.2	10.2
Strong narrowing or near-solution scaffold	29.7	14.3	45.1	51.2
Key mechanism disclosure	27.0	18.5	31.0	29.3
Minor local correction	9.0	6.7	1.5	2.3
Answer revealed without full explanation	5.4	5.0	5.3	5.1
Partial worked solution	0.0	0.0	1.5	1.2
General reframing hint	5.4	12.6	0.9	0.0
Wrong-path rejection	7.2	11.8	0.0	0.0
Validation only	5.4	10.9	0.0	0.0
Broad process hint	1.8	5.9	0.3	0.0
No substantive reveal	1.8	5.0	0.3	0.4
Method-level hint	1.8	2.5	0.0	0.4

reasoning before arriving at the answer—they can simply read and extract the solution. We explore this hypothesis further in §5.4.

5.3 Do interventions help students generalize to new problems (RQ3)?

Interventions do not consistently improve student generalization ability. Across all three domains, neither with-context condition yielded a significant accuracy gain (Figure 4). The *Problem-Context* alone produced small domain-level shifts (math: $G = -0.01$; code: $G = -0.03$; brain: $G = +0.03$). Similarly, the *Intervention-Context* condition also did not improve generalization across domains (math: $G = -0.02$; code: $G = -0.04$; brain: $G = +0.04$), where each score G is averaged over the four teacher models. This indicates that access to teacher interventions and revised reasoning does not reliably translate into transferable problem-solving ability on new instances. One possible explanation is that interventions are overly specific to the original problem and emphasize

instance-level corrections rather than more general, transferable strategies. As a result, the added context may fail to provide reusable insights, and in some cases, can even reduce performance when irrelevant information distracts the model (Shi et al., 2023; Wu et al., 2024).

5.4 How do LLMs compare to human behavior? (RQ4)

Humans intervene much less and much later than LLMs. In the *Standard* monitoring condition, humans intervened less often and later than LLMs (Human: $\phi = 0.74$, $\tau_{\text{rel}} = 0.74$; LLM: $\phi = 0.94$, $\tau_{\text{rel}} = 0.24$; see Figure 5). Human intervention behavior changed little across conditions ($\Delta\phi = +0.05$, $\Delta\tau_{\text{rel}} = -0.11$), while LLMs shifted their strategy substantially ($\Delta\phi = -0.23$, $\Delta\tau_{\text{rel}} = +0.33$). To quantify how much human and LLM strategies diverged across conditions, we fit two Bayesian mixed models.[3] We found that both outcomes showed a large interaction: frequency ($\beta = -2.60$, 95% credible interval (CrI) $[-3.40, -1.80]$) and relative timing ($\beta = 0.45$, 95% CrI $[0.36, 0.53]$). This interaction indicates that LLMs and humans respond to the kind of information available in different ways, such that while LLMs became significantly more selective and waited longer to intervene when they had full context, humans did not exhibit these changes.

Furthermore, on questions that the student would have solved correctly without intervention (6 out of 30 brain teasers), LLMs reduced their intervention rate when given full information (*Standard*: $\phi_{\text{correct}} = 0.72$; *Oracle*: $\phi_{\text{correct}} = 0.00$), whereas humans intervened at a similar rate across conditions (*Standard*: $\phi_{\text{correct}} = 0.37$; *Oracle*: $\phi_{\text{correct}} = 0.40$). In these cases, rather than remaining silent, people sometimes praised or confirmed the student's answer (4 of 12 interventions), or they intervened to flag a flawed intermediate step in the reasoning, despite the student eventually arriving at the correct answer. This suggests that people draw on richer pedagogical cues beyond answer correctness when deciding how to help.

To analyze the qualitative differences between human and LLM interventions, we categorized all messages (for the 30 selected brain teaser questions) along two dimensions: (i) *functional role*, how the intervention supports the student, and (ii) *solution leakage*, the extent to which the message directly discloses the solution (Table 1; see Appendix C.3 for details).

Both LLMs and humans primarily intervene to redirect unproductive reasoning. Across both monitoring conditions, human and LLM-based interventions most frequently redirected the student away from unproductive reasoning paths (Table 1a). While LLMs also often reframed the problem representation (∼24%), humans more often relied on other strategies, such as correcting local errors (∼13%) and refocusing student attention on salient evidence (∼13%). Finally, humans sometimes validated the student's reasoning process (up to 6.7%), whereas the models never did.

LLMs often reveal substantial solution content, while humans offer more indirect guidance. Overall, LLMs disclosed substantially more solution information than humans (Table 1b). LLMs revealed the full solution roughly twice as often as humans (14.2% vs. 5.4% in the *Standard* condition, 10.2% vs. 6.7% in the *Oracle* condition). Models also heavily preferred providing near-complete solution scaffolding (∼48%), more so than humans did (∼22%). Compared to models, humans instead relied more on wrong-path rejections and general reframing hints.

6 Discussion

In this work, we formalize helping as a sequential intervention game and introduce INT-BENCH, a simulation-based benchmark for studying when, how, and how often LLMs intervene during problem solving. Using formal metrics and a complementary human study, we find that LLMs intervene more frequently and earlier than humans, often providing overly informative, instance-specific feedback that includes substantial solution content. In contrast, in both partial and full information settings, humans tend to rely more on hints, scaffolding, and trajectory-aware guidance that leaves room for continued reasoning. These results highlight a tension between short-term task success and longer-term learning. Though current AI assistants may be effective at correcting individual instances, they may be less reliable at preserving reasoning opportunities or supporting transfer to new problems.

[3] We fit a logistic mixed-effect model for intervention frequency and a Gaussian linear mixed-effect model for relative timing. Both include fixed effects for agent (human/LLM), condition (Standard/Oracle), and their interaction, with a random intercept for question: $Y_{ij} \sim \beta_0 + \beta_1\,\text{agent}_i + \beta_2\,\text{condition}_j + \beta_3\,(\text{agent} \times \text{condition})_{ij} + u_j,\ u_j \sim \mathcal{N}(0, \sigma_q^2)$.

AI assistants should ideally do more than correct an answer for a specific problem; they should help users remain engaged, build the persistence needed to tackle future tasks independently, and become better general problem solvers. In learning settings, effort and productive struggle are often key ingredients to forming good understanding (Inzlicht et al., 2018; Marsh et al., 2022; Norton et al., 2012). More broadly, repeated exposure to overly direct assistance may gradually shift reasoning processes from humans to AI systems—a form of "gradual disempowerment" in which users become increasingly dependent on AI for how to reason through problems (Kulveit et al., 2025; Sharma et al., 2026). Recent work exploring assistants that are trained to maximize user empowerment (Ellis et al., 2025) indicates one potential direction, though designing assistants that balance short-term help with long-term success remains an open challenge.

Our experiments currently rely on LLM-simulated students. Though this allows for more controlled, large-scale evaluation of intervention behavior, it is unclear how closely simulated student responses reflect the cognitive and motivational processes at play in human learning (e.g., misconceptions, cognitive load, and affective responses to feedback). Additionally, our measure of generalization is limited to immediate transfer on a single related problem; however, interventions may influence learning over longer timescales, such as through repeated exposure and practice, which the current design does not capture. Finally, extending INT-BENCH to more student models, longer multi-episode interaction sequences, and real human learners would enable a more comprehensive account of AI interventions and their effects on users.

Acknowledgments

We would like to thank the Supervised Program for Alignment Research (SPAR) for facilitating collaboration. MKW was supported by Toyota Research Institute (TRI), Cooperative AI Foundation, the Foresight Institute, the Sony Research Award Program, UW-Tsukuba Amazon NVIDIA Cross Pacific AI Initiative, Jacobs CIFAR Research Fellowship, Templeton World Charity Foundation (https://doi.org/10.54224/34843). TG was supported by grants from the Stanford Institute for Human-Centered Artificial Intelligence (HAI), Toyota Research Institute (TRI), and the Cooperative AI Foundation.

References

Zahra Abbasiantaeb, Yifei Yuan, Evangelos Kanoulas, and Mohammad Aliannejadi. Let the llms talk: Simulating human-to-human conversational qa via zero-shot llm-to-llm interactions, 2023. URL https://arxiv.org/abs/2312.02913.

Robert A Bjork. Memory and metamemory considerations in the training of human beings. In Janet Metcalfe and Arthur P Shimamura, editors, *Metacognition: Knowing about knowing*, pages 185–205. MIT Press, 1994.

Aidan V. Campbell, Yiyi Wang, and Michael Inzlicht. Experimental evidence that exerting effort increases meaning. *Cognition*, 257:106065, 2025. ISSN 0010-0277. doi: https://doi.org/10.1016/j.cognition.2025.106065. URL https://www.sciencedirect.com/science/article/pii/S0010027725000058.

Aaron Chatterji, Thomas Cunningham, David J Deming, Zoe Hitzig, Christopher Ong, Carl Yan Shan, and Kevin Wadman. How people use chatgpt. Technical report, National Bureau of Economic Research, 2025.

Xinyue Chen, Kunlin Ruan, Kexin Phyllis Ju, Nathan Yap, and Xu Wang. More ai assistance reduces cognitive engagement: Examining the ai assistance dilemma in ai-supported note-taking. *Proceedings of the ACM on Human-Computer Interaction*, 9(7):1–29, October 2025. ISSN 2573-0142. doi: 10.1145/3757632. URL http://dx.doi.org/10.1145/3757632.

Nico Daheim, Jakub Macina, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. Stepwise verification and remediation of student reasoning errors with large language model tutors. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8386–8411, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.478. URL https://aclanthology.org/2024.emnlp-main.478/.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. Metacognitive capabilities of llms: An exploration in mathematical problem solving. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 19783–19812. Curran Associates, Inc., 2024. doi: 10.52202/079017-0623. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/2318d75a06437eaa257737a5cf3ab83c-Paper-Conference.pdf.

Evan Ellis, Vivek Myers, Jens Tuyls, Sergey Levine, Anca Dragan, and Benjamin Eysenbach. Training llm agents to empower humans, 2025. URL https://arxiv.org/abs/2510.13709.

Cedric Faas, Richard Bergs, Sarah Sterz, Markus Langer, and Anna Maria Feit. Give me a choice: The consequences of restricting choices through ai-support for perceived autonomy, motivational variables, and decision performance, 2024. URL https://arxiv.org/abs/2410.07728.

Krzysztof Z. Gajos and Lena Mamykina. Do people engage cognitively with ai? impact of ai assistance on incidental learning. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, IUI '22, page 794–806, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391443. doi: 10.1145/3490099.3511138. URL https://doi.org/10.1145/3490099.3511138.

Gaole He, Gianluca Demartini, and Ujwal Gadiraju. Plan-then-execute: An empirical study of user trust and team performance when using llm agents as a daily assistant. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400713941. doi: 10.1145/3706598.3713218. URL https://doi.org/10.1145/3706598.3713218.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL https://arxiv.org/abs/2103.03874.

Bihao Hu, Jiayi Zhu, Yiying Pei, and Xiaoqing Gu. Exploring the potential of llm to enhance teaching plans through teaching simulation. *npj Science of Learning*, 10(1), December 2025. ISSN 2056-7936. doi: 10.1038/s41539-025-00300-x. Publisher Copyright: © The Author(s) 2025.

Michael Inzlicht, Amitai Shenhav, and Christopher Y. Olivola. The effort paradox: Effort is both costly and valued. *Trends in Cognitive Sciences*, 22(4):337–349, 2018. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2018.01.007. URL https://www.sciencedirect.com/science/article/pii/S1364661318300202.

Hyoungwook Jin, Minju Yoo, Jeongeon Park, Yokyung Lee, Xu Wang, and Juho Kim. Teachtune: Reviewing pedagogical agents against diverse student profiles with simulated students, 2025. URL https://arxiv.org/abs/2410.04078.

Manu Kapur. Productive failure in learning math. *Cognitive Science*, 38(5):1008–1022, 2014.

Kenneth R Koedinger and Vincent Aleven. Exploring the assistance dilemma in experiments with cognitive tutors. *Educational Psychology Review*, 19(3):239–264, 2007.

Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. Your brain on chatgpt: Accumulation of cognitive debt when using an ai assistant for essay writing task, 2025. URL https://arxiv.org/abs/2506.08872.

Jan Kulveit, Raymond Douglas, Nora Ammann, Deger Turan, David Krueger, and David Duvenaud. Position: Humanity faces existential risk from gradual disempowerment. In *Forty-second International Conference on Machine Learning Position Paper Track*, 2025.

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

Grace Liu, Brian Christian, Tsvetomira Dumbalska, Michiel A. Bakker, and Rachit Dubey. Ai assistance reduces persistence and hurts independent performance, 2026. URL https://arxiv.org/abs/2604.04721.

Jiayu Liu, Zhenya Huang, Tong Xiao, Jing Sha, Jinze Wu, Qi Liu, Shijin Wang, and Enhong Chen. Socraticlm: Exploring socratic personalized teaching with large language models. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 85693–85721. Curran Associates, Inc., 2024. doi: 10.52202/079017-2721. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9bae399d1f34b8650351c1bd3692aeae-Paper-Conference.pdf.

Xinyi Lu and Xu Wang. Generative students: Using llm-simulated student profiles to support question item evaluation. In *Proceedings of the Eleventh ACM Conference on Learning @ Scale*, L@S '24, page 16–27. ACM, 2024. doi: 10.1145/3657604.3662031. URL http://dx.doi.org/10.1145/3657604.3662031.

Jakub Macina, Nico Daheim, Sankalan 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 Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.372. URL https://aclanthology.org/2023.findings-emnlp.372/.

Mehak Maniktala, Christa Cody, Amy Isvik, Nicholas Lytle, Min Chi, and Tiffany Barnes. Extending the hint factory for the assistance dilemma: A novel, data-driven helpneed predictor for proactive problem-solving help. *Journal of Educational Data Mining*, 12(4):24–65, Dec. 2020. doi: 10.5281/zenodo.4399683. URL https://jedm.educationaldatamining.org/index.php/JEDM/article/view/450.

Emaad Manzoor, Eva Ascarza, and Oded Netzer. Learning when to quit in sales conversations, 2025. URL https://arxiv.org/abs/2511.01181.

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

Lauren Marsh, Joanna Gil, and Patricia Kanngiesser. The influence of collaboration and culture on the ikea effect: Does cocreation alter perceptions of value in british and indian children? *Developmental Psychology*, 58:662–670, 04 2022. doi: 10.1037/dev0001321.

Jordan K. Matelsky, Felipe Parodi, Tony Liu, Richard D. Lange, and Konrad P. Kording. A large language model-assisted education tool to provide feedback on open-ended responses, 2023. URL https://arxiv.org/abs/2308.02439.

Bruce Mclaren, Sung-Joo Lim, and Kenneth Koedinger. When and how often should worked examples be given to students? new results and a summary of the current state of research. *Cognitive Science*, pages 2176–2181, 01 2008.

Bruce M. Mclaren, Tamara Gog, Craig Ganoe, David Yaron, and Michael Karabinos. Exploring the assistance dilemma: Comparing instructional support in examples and problems. In *12th International Conference on Intelligent Tutoring Systems - Volume 8474*, ITS 2014, page 354–361, Berlin, Heidelberg, 2014. Springer-Verlag. ISBN 9783319072203. doi: 10.1007/978-3-319-07221-0_44. URL https://doi.org/10.1007/978-3-319-07221-0_44.

Michael I. Norton, Daniel Mochon, and Dan Ariely. The ikea effect: When labor leads to love. *Journal of Consumer Psychology*, 22(3):453–460, 2012. ISSN 1057-7408. doi: https://doi.org/10.1016/j.jcps.2011.08.002. URL https://www.sciencedirect.com/science/article/pii/S1057740811000829.

Deep Anil Patel, Iain Melvin, Christopher Malon, and Martin Renqiang Min. Discussllm: Teaching large language models when to speak, 2026. URL https://arxiv.org/abs/2508.18167.

Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. The impact of ai on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590*, 2023.

Evan F. Risko and Sam J. Gilbert. Cognitive offloading. *Trends in Cognitive Sciences*, 20(9): 676–688, 2016. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2016.07.002. URL https://www.sciencedirect.com/science/article/pii/S1364661316300985.

Alexis Ross and Jacob Andreas. Learning to make mistakes: Modeling incorrect student thinking and key errors, 2025. URL https://arxiv.org/abs/2510.11502.

Alexis Ross, Megha Srivastava, Jeremiah Blanchard, and Jacob Andreas. Modeling student learning with 3.8 million program traces, 2025. URL https://arxiv.org/abs/2510.05056.

Reut Shachnai, Max Kleiman-Weiner, Marlene Berke, and Julia Anne Leonard. When bayesians take over: A computational model of parental intervention. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47, 2025.

Mrinank Sharma, Miles McCain, Raymond Douglas, and David Duvenaud. Who's in charge? disempowerment patterns in real-world llm usage, 2026. URL https://arxiv.org/abs/2601.19062.

Steven D Shaw and Gideon Nave. Thinking-fast, slow, and artificial: How ai is reshaping human reasoning and the rise of cognitive surrender. *Available at SSRN 6097646*, 2026.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H. Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31210–31227. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

Nicholas C Soderstrom and Robert A Bjork. Learning versus performance: An integrative review. *Perspectives on Psychological Science*, 10(2):176–199, 2015.

Matthias Stadler, Maria Bannert, and Michael Sailer. Cognitive ease at a cost: Llms reduce mental effort but compromise depth in student scientific inquiry. *Computers in Human Behavior*, 160: 108386, 2024. ISSN 0747-5632. doi: https://doi.org/10.1016/j.chb.2024.108386. URL https://www.sciencedirect.com/science/article/pii/S0747563224002541.

Mark Steyvers and Lukas Mayer. When not to help: planning for lasting human-ai collaboration, 2025. URL https://arxiv.org/abs/2508.01837.

Anais Tack and Chris Piech. The AI teacher test: Measuring the pedagogical ability of blender and GPT-3 in educational dialogues. In Antonija Mitrovic and Nigel Bosch, editors, *Proceedings of the 15th International Conference on Educational Data Mining*, pages 522–529, Durham, United Kingdom, July 2022. International Educational Data Mining Society. ISBN 978-1-7336736-3-1. doi: 10.5281/zenodo.6853187.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. How easily do irrelevant inputs skew the responses of large language models?, 2024. URL https://arxiv.org/abs/2404.03302.

Weiqing Yang, Hanbin Wang, Zhenghao Liu, Xinze Li, Yukun Yan, Shuo Wang, Yu Gu, Minghe Yu, Zhiyuan Liu, and Ge Yu. COAST: Enhancing the code debugging ability of LLMs through communicative agent based data synthesis. In Luis Chiruzzo, Alan Ritter, and Lu Wang, editors, *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2570–2585, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-195-7. doi: 10.18653/v1/2025.findings-naacl.139. URL https://aclanthology.org/2025.findings-naacl.139/.

Appendix

Contents

A Data

A.1 Brain Teaser

For the brain teaser dataset, we scraped brain teasers from the website Braingle.com, from the following categories: "Language," "Group," "Rebus," and "Riddle." Language teasers are those that involve the English language, often requiring one to think about and manipulate words and letters. Group teasers involve recognizing groups and common attributes amongst words or letters. Rebus brain teasers involve putting words or letters in interesting orientations to represent common phrases. Riddles are short poems or stories that describe something in a mysterious or indirect way. They often can rhyme and pose a question with a hidden meaning.

To ensure the questions were suitable for our text-based models and human participants, we applied the following filters during scraping:

- **Difficulty**: We restricted scraping to questions with a difficulty rating of ≤ 2.0 (out of 4.0) to focus on questions that are solvable without too much time or specialized knowledge.
- **Popularity**: We required a popularity ("fun") rating of ≥ 2.0 (out of 4.0) to ensure the questions were generally well-regarded and engaging.

After scraping, we further cleaned the dataset by removing any questions containing images and keywords suggesting visuals. From the remaining pool, we aggregated a dataset of 500 questions by random sampling, after ensuring the inclusion of our manually selected subset.

For the human study, we manually selected a subset of 30 questions. The selection process involved reviewing the top 500 scraped questions (ranked by most popular and least difficult) and choosing those that did not involve multiple subparts, require extensive external or domain knowledge, or contain overly difficult vocabulary. These 30 questions were used in both the model simulations and for the human experiment.

A.2 Simulation Data

Our simulation consisted of 18,000 episodes per monitoring condition (500 questions $\times$ 3 domains $\times$ 4 teacher models $\times$ 3 runs), for a total of 36,000 teacher rollouts across the *Standard* and *Oracle* conditions. The *Standard-Continue* and *Stop-and-Answer* post-intervention reasoning conditions included only the *Standard* monitoring condition episodes where the *Standard* teacher chose to intervene—16,111 and 16,110 episodes, respectively, aggregated across all three domains (small differences between the two reflect miscellaneous API errors). *Oracle-Continue* was further restricted to *Oracle* teacher interventions with a valid recoverable timing anchor.

Oracle Timing Recovery In the *Oracle* monitoring condition, the teacher reports its chosen intervention point as a 5-word verbatim quote from the reasoning transcript. We recover the character position by searching for this quote using a three-stage procedure: (1) exact substring match, (2) case-insensitive match, and (3) flexible-whitespace token regex (preserving operators and punctuation). Approximately 23% of *Oracle* interventions were excluded (2,241 of 9,695 intervened episodes): $\sim$16% (1,529 episodes) because the quote could not be located, $\sim$5% (477 episodes) because the recovered position fell in the final answer region rather than the reasoning trace, $\sim$2% (229 episodes) due to multiple possible matches, and a small remainder (6 episodes) in which the teacher response contained no quote at all. Retention rates varied by domain: 83.0% for brain teasers, 70.2% for code debugging, and 71.5% for mathematics. The average retained $\sim$77% (7,454 episodes) formed the basis of all *Oracle* teacher timing statistics (τ_{rel}, τ_{abs}) and *Oracle-Continue* episodes.

Table 2: Number of episodes used in each post-intervention analysis condition, per domain.

Condition	Code Debugging	Mathematics	Brain Teasers	Total
Standard-Continue	5,705	4,727	5,679	16,111
Stop-and-Answer	5,704	4,727	5,679	16,110
Oracle-Continue	2,128	1,236	4,082	7,446

B Human Study

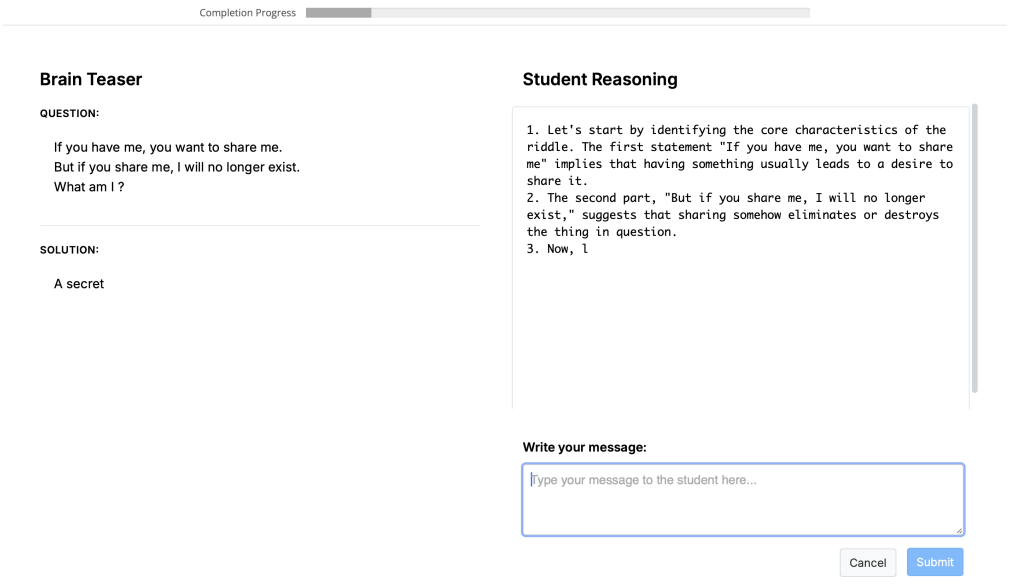

Figure 6: Interface for the human study. Participants see the brain teaser information (question and solution) on the left side of the screen. The right side of the screen shows the student's reasoning trace, which can be revealed by pressing the right arrow key. Participants can pause and intervene by pressing the space bar, at which point a text box appears where they can type their intervention message.

B.1 Participants

We recruited participants via the crowd-sourcing platform Prolific. We included participants who are fluent in English, based in the United States, have an approval rate of at least 98%, and have completed at least 300 previous studies. Participants were compensated at a rate of $12 per hour. All studies were approved by our Institutional Review Board (IRB).

B.2 Procedure

Participants first read instructions explaining the experimental setup. They were asked to imagine that they are a teacher working with some students who are trying to solve brain teasers. They were told that their goal as the teacher is to help the students solve the problems. After reading the instructions, participants completed two example trials and a comprehension check before proceeding to the main trials. For each trial, only after acknowledging that they had read the question and understood the solution could they begin revealing the student's reasoning trace.

At the end of the study, participants were asked how many years of teaching experience they have, whether they are a parent or guardian of children (if so, their age ranges), whether they used AI tools during the experiment (if so, how they used it), how they decided whether and when to intervene, and any additional feedback they might have.

B.3 Participant Demographics Analysis

In the *Standard* condition, 52% of participants had no teaching experience, 24% had less than 1 year, and 24% had over 1 year of teaching experience. 48% were parents or guardians of children. In the *Oracle* condition, 60% had no teaching experience, 12% had less than 1 year, and 28% had over 1 year of teaching experience. 68% were parents or guardians of children.

We examined whether teaching experience and parenthood status affected intervention behavior. For each, we split participants into two groups (no teaching experience vs. teaching experience, and

non-parent vs. parent) and compared per-participant means on intervention frequency (ϕ) and relative timing (τ_{rel}) using two-sided Mann-Whitney U tests.

Teaching Experience We split participants into those with no teaching experience (*no experience*: Standard $n = 13$, Oracle $n = 15$) and those with any teaching experience (*experienced*: Standard $n = 12$, Oracle $n = 10$). Pooling both monitoring conditions, experienced and no-experience participants showed similar intervention rates and timing (ϕ: $U = 293.0$, $p = 0.769$; τ_{rel}: $U = 213.0$, $p = 0.093$).

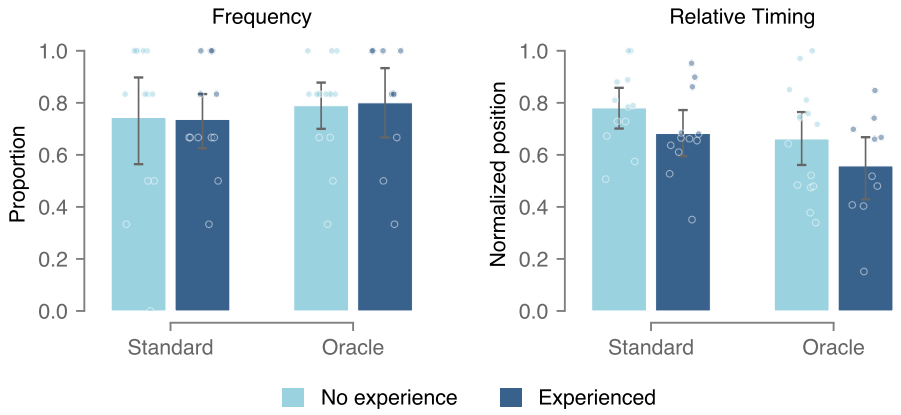

Figure 7: Intervention frequency (ϕ, left) and relative timing (τ_{rel}, right) for no-experience vs. experienced participants, shown for Standard and Oracle monitoring conditions. Error bars represent 95% bootstrapped CIs. Participants are indicated by individual points.

Parenthood For parenthood, we split participants into parents (or guardians) (*parent*: Standard $n = 12$, Oracle $n = 17$) and non-parents (*non-parent*: Standard $n = 13$, Oracle $n = 8$). Neither intervention frequency nor relative timing differed between groups (ϕ: $U = 366.0$, $p = 0.213$; τ_{rel}: $U = 318.0$, $p = 0.635$).

Overall, these results suggest that neither teaching experience nor parenthood reliably modulates intervention frequency or timing in our paradigm, supporting the robustness of the human baseline across participant backgrounds.

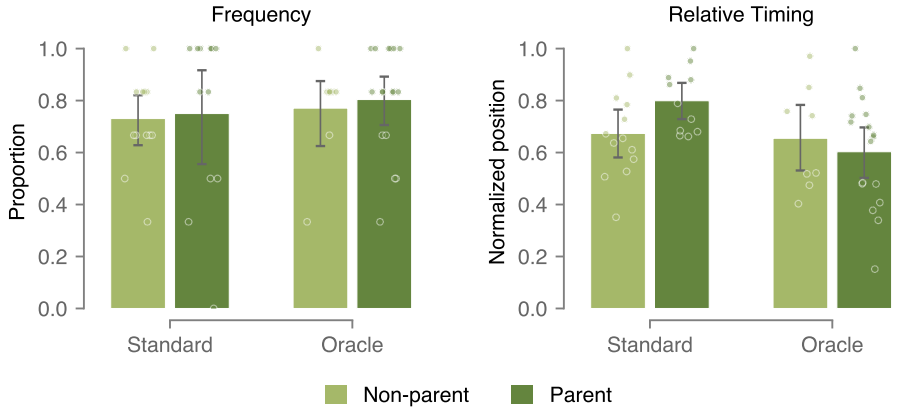

Figure 8: Intervention frequency (ϕ, left) and relative timing (τ_{rel}, right) for non-parent vs. parent participants, shown for Standard and Oracle monitoring conditions. Error bars represent 95% bootstrapped CIs. Participants are indicated by individual points.

Table 3: Specifications of the structured problem-generation pipeline. The skill extractor $\mathcal{E}$ identifies latent reasoning skills, the clusterer $\mathcal{C}$ constructs a taxonomy of high-level categories of skills, the generator $\mathcal{G}$ produces skill-preserving problem variants, and the validator $\mathcal{V}$ enforces correctness of generated problems across domains.

Component	Description	Inputs	Outputs
Skill Extractor $\mathcal{E}$	Identifies the latent reasoning skill or capability required to solve a reference problem.	Problem statement; and baseline solution	Fine-grained skill label s_i.
Skill Clusterer $\mathcal{C}$	Groups similar fine-grained skills into high-level categories to construct a consistent dataset-level taxonomy.	Set of extracted skills $\{s_i\}$ from skill extractor $\mathcal{E}$.	High-level skill categories $\mathcal{S} = \{S_1, \ldots, S_K\}$.
Problem Generator $\mathcal{G}$	Generates candidate variants that preserve the target skill category while modifying surface form, context, or parameters.	Reference problem Q; assigned skill category S_k; generation constraints; N.	Candidate question set $\tilde{\mathcal{Q}} = \{\tilde{Q}_1, \ldots, \tilde{Q}_N\}$ and candidate answer set $\tilde{\mathcal{Y}} = \{\tilde{Y}_1, \ldots, \tilde{Y}_N\}$.
Problem Validator $\mathcal{V}$	Verifies correctness of generated variants and filters out invalid or misaligned problems.	Candidate variants $(\tilde{\mathcal{Q}}, \tilde{\mathcal{Y}})$; skill categories $\mathcal{S}$.	Validated set $(\tilde{\mathcal{Q}}, \tilde{\mathcal{Y}})_{\text{valid}} \subseteq (\tilde{\mathcal{Q}}, \tilde{\mathcal{Y}})$.

C Pipeline Details

C.1 Math Evaluation

For the MATH-500 dataset, we used a deterministic grading methodology rather than an LLM-based judge (Lightman et al., 2023). The grading process includes two stages: (1) normalizing the student's answer and the ground truth to a canonical format, and (2) checking for mathematical equivalence using SymPy. This ensures that our evaluation is objective and reproducible, avoiding the potential variability and biases of LLM-based grading for math questions.

C.2 Problem Generation

Table 3 summarizes the functionality and role of each component in our structured problem-generation pipeline. In our setup, we generated 5 candidate variants per reference problem. We used GPT-5.2 as the model for all parts of the pipeline. Prompts for each component are provided in §F.5.

C.3 Categorization of Interventions

We developed a bottom-up categorization pipeline to label both human and LLM interventions along two complementary dimensions: (i) the *functional role* of the intervention in supporting the student, and (ii) the *degree of solution information revealed*.

Let $\mathcal{M}$ denote the set of all intervention messages. Our pipeline proceeds in three stages. First, each intervention message $m \in \mathcal{M}$ is passed to an LLM L_1, which assigns an initial high-level category $c \in \mathcal{C}$. Second, the set of all generated categories $\mathcal{C}$ is aggregated and provided to a separate LLM L_2, which clusters semantically similar categories to construct a consolidated taxonomy $\mathcal{C}^\star$. Finally, each message m is relabeled by a third LLM L_3, which assigns a final category $\hat{c} \in \mathcal{C}^\star$, ensuring consistency and comparability across the dataset. We used GPT-5.2 as the model for each stage of this pipeline. All prompts are provided in §F.7.

D Results

D.1 LLM Interventions

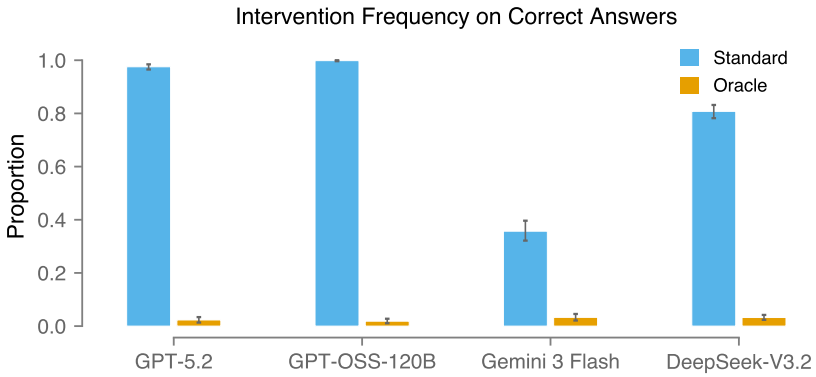

Figure 9: Frequency of interventions on problems where the student's unassisted baseline answer was correct (ϕ_{correct}), shown across different teacher models and evaluation domains. Error bars indicate 95% bootstrapped confidence intervals.

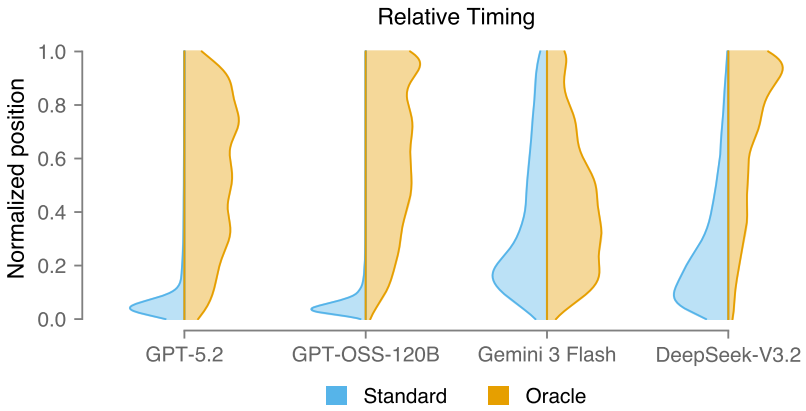

Figure 10: Relative intervention timing (τ_{rel}) across all models and domains for both *Standard* and *Oracle* conditions. In the *Standard* condition, models intervene much earlier in the reasoning process compared to the *Oracle* condition, where they have access to the full trace, student answer, and correctness, and can select a more optimal point to intervene.

Table 4: Intervention metrics (intervention frequency ϕ, frequency on correct answers ϕ_{correct}, relative timing τ_{rel}, and absolute character-based timing τ_{abs}) across all teacher models, domains, and monitoring conditions.

Model	Standard				Oracle			
	ϕ	ϕ_{correct}	τ_{rel}	τ_{abs}	ϕ	ϕ_{correct}	τ_{rel}	τ_{abs}
Code Debugging								
GPT-5.2	1.00	1.00	0.04	51.67	0.57	0.06	0.55	1033.66
GPT-OSS-120B	1.00	1.00	0.05	58.17	0.41	0.02	0.70	1491.13
Gemini 3 Flash	0.83	0.74	0.35	499.60	0.49	0.06	0.48	760.93
DeepSeek-V3.2	0.98	0.96	0.18	243.56	0.56	0.07	0.60	1179.36
Mathematics								
GPT-5.2	0.97	0.95	0.15	150.21	0.29	0.01	0.57	1047.76
GPT-OSS-120B	1.00	1.00	0.07	67.03	0.30	0.02	0.58	1096.61
Gemini 3 Flash	0.37	0.13	0.46	760.89	0.27	0.01	0.48	860.10
DeepSeek-V3.2	0.81	0.74	0.33	401.81	0.30	0.01	0.77	1468.33
Brain Teaser								
GPT-5.2	1.00	1.00	0.05	59.55	0.86	0.00	0.51	621.17
GPT-OSS-120B	1.00	1.00	0.06	61.86	0.73	0.02	0.56	690.85
Gemini 3 Flash	0.86	0.26	0.36	423.75	0.85	0.04	0.42	515.07
DeepSeek-V3.2	0.93	0.68	0.28	319.42	0.84	0.04	0.71	893.48

D.2 Human vs. LLM Brain Teaser Comparisons

We compare human vs. LLM intervention behavior using the 30 manually selected brain teaser questions described in §A.1. On the human side, we collected 150 human episodes per condition (25 participants $\times$ 6 episodes each, i.e., each question has 5 human annotations) and 360 LLM episodes per condition (30 questions $\times$ 4 teacher models $\times$ 3 repetitions).

Humans in the *Standard* condition intervened on 111/150 episodes, and 119/150 in the *Oracle* condition. LLMs in the *Standard* condition intervened on 339/360 episodes. In the *Oracle* condition, they attempted to intervene on 256/360 episodes. However, 31 of the 256 interventions from the *Oracle* LLMs could not be assigned a character position by the recovery procedure in §A.2. Timing-based results in this section therefore used $n = 225$ LLM *Oracle* interventions (frequency and category-based statistics are unaffected).

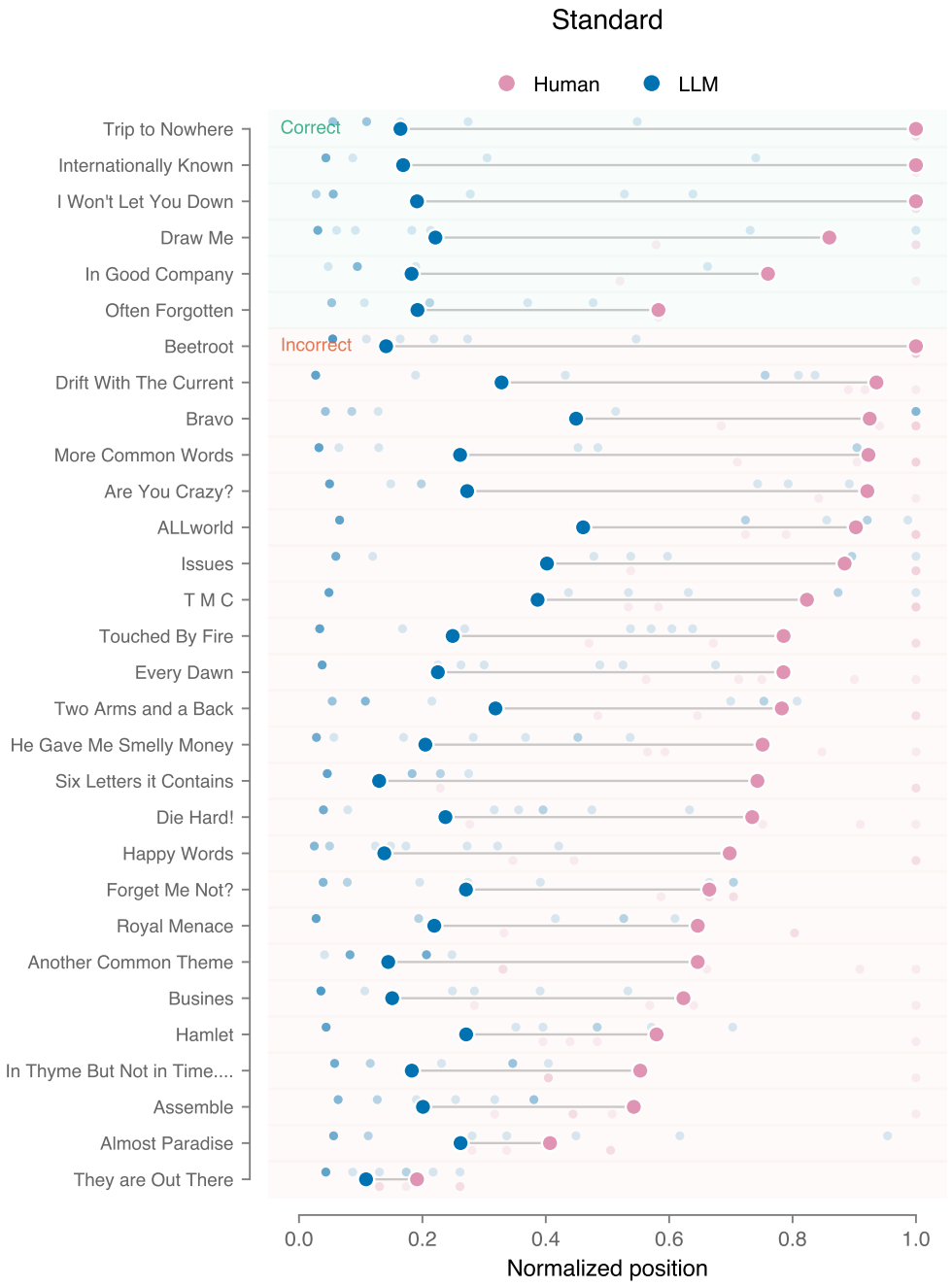

Figure 11: Trial comparison of mean intervention timing between human and LLM teachers in the *Standard* condition across the 30 brain teaser questions used in the human study. LLM results are averaged across all four evaluated models. Questions are grouped by whether the student's baseline initial answer (i.e., prior to any intervention) was correct or incorrect. Small dots represent individual episodes.

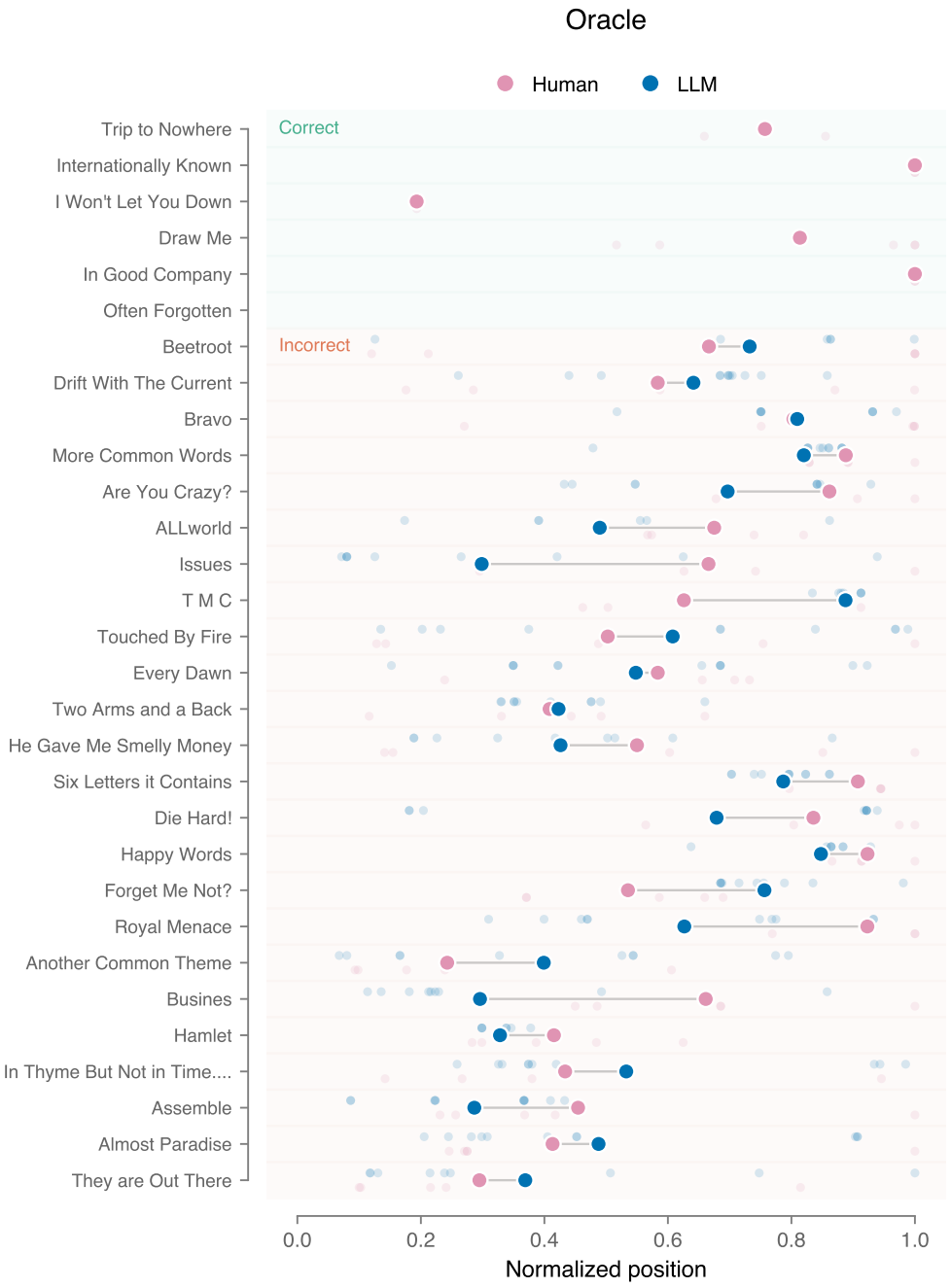

Figure 12: Trial comparison of mean intervention timing between human and LLM teachers in the *Oracle* condition across the 30 brain teaser questions used in the human study. LLM results are averaged across all four evaluated models. Questions are grouped by whether the student's baseline initial answer (i.e., prior to any intervention) was correct or incorrect. Small dots represent individual episodes.

D.3 Examples

Table 5: Example of a brain teaser and human intervention collected during the study.

Question	**Title**: In Thyme But Not in Time... **Content**: My first is in riddle, but not in little. My second is in think, but not in brink. My third is in thyme, but not in time. My fourth is in mother, but not in brother. My last is in time, but not in climb. What am I? **Solution**: The word rhyme.
Student Reasoning	1. First letter: "My first is in riddle, but not in little." - "Riddle" contains 'r', but "little" does not contain 'r'. - Therefore, the first letter is 'r'. 2. Second letter: "My second is in think, but not in brink." - "Think" contains 'i', but "brink" does not contain 'i'. - Therefore, the second letter is 'i'. 3. Third letter: "M
Intervention	Brink contains an i. Try again

Table 6: Example of a brain teaser and human intervention collected during the study.

Question	**Title**: Beetroot **Content**: What advice is shown below? BEETROOT YEETROOT YOETROOT YOUTROOT YOURROOT YOURSOOT YOURSEOT YOURSELT YOURSELF **Solution**: Be true to yourself. BEETROOT to YOURSELF (sound it out)
Student Reasoning	1. The given sequence of words starts with "BEETROOT" and each subsequent line has one letter moved from the end to the beginning. 2. We need to observe the pattern in the rearrangement process: - BEETROOT -> YEETROOT (Y moves from last to first) - YEETROOT -> YOETROOT (O moves from last to first) - YOETROOT -> YOUTROOT (U moves from last to first) - YOUTROOT -> YOURROOT (R moves from last to first) - YOURROOT -> YOURSOOT (S moves from last to first) - YOURSOOT -> YOURSEOT (E moves from last to first) - YOURSEOT -> YOURSELT (L moves from last to first) - YOURSELT -> YOURSELF (F moves from last to first) 3. Each word maintains its original structure while progressively repositioning the last letter until it reaches the front, forming the word "YOURSELF". 4. The process essentially shows how to spell out "YOURSELF" by moving the letters from the end to the beginning one at a time.
Intervention	Why your answer is true, you need to simplfy. Simply state the phrase outloud and you will hear the correct answer.

Table 7: Example of an intervention from an LLM-based teacher (GPT-5.2) in the *Standard* condition.

Question	**Title**: Beetroot **Content**: What advice is shown below? BEETROOT YEETROOT YOETROOT YOUTROOT YOURROOT YOURSOOT YOURSEOT YOURSELT YOURSELF **Solution**: Be true to yourself. BEETROOT to YOURSELF (sound it out)
Student Reasoning	1. The given sequence of words starts with "BEETRO
Intervention	Try reading the first and last words aloud and "hearing" the phrase they form: BEETROOT $\rightarrow$ "be true to" and YOURSELF stays "yourself." The intermediate steps just morph the letters to get from one to the other.

E Additional Experiments

E.1 Increment Sizes

We investigated the effect of increment size on intervention behavior. We experimented with revealing the reasoning in 300-character and per-sentence increments. From Table 8, we observed that intervention frequency remained unchanged across all conditions and the relative position is consistently small. Moreover, the absolute intervention position (τ_{abs}) closely tracks the increment size itself, indicating that the teacher intervened immediately after viewing the first increment, regardless of granularity.

Table 8: Impact of different increment sizes on intervention frequency and timing in the code debugging domain, with GPT-5.2 as the teacher model in the *Standard* condition.

Increment Size	ϕ	τ_{rel}	τ_{abs}
50-character	1.00	0.04	50.6
300-character	1.00	0.25	301.9
1-sentence	1.00	0.11	138.0

E.2 Student Models

We examined whether the student model affects teacher intervention behavior by varying the student across a larger model (Qwen3-32B), a different model family (Llama-3.1-8B), and the same model as the teacher (GPT-5.2). As shown in Table 9, intervention frequency and timing remained nearly identical across all student models.

Table 9: Impact of different student models on teacher intervention frequency and timing in the code debugging domain, with GPT-5.2 as the teacher model in the *Standard* condition.

Student Model	ϕ	τ_{rel}	τ_{abs}
Qwen2.5-7B	1.00	0.04	50.6
GPT-5.2	1.00	0.07	51.0
Llama-3.1-8B	1.00	0.04	50.1
Qwen3-32B	1.00	0.08	50.3

E.3 Prompt Sensitivity

We conducted a prompt sensitivity analysis to verify that intervention behavior is robust to prompt wording. We tested two prompt variations: "Variation 1" adds an explicit constraint to only intervene if truly necessary, while "Variation 2" instructs the teacher to scaffold and guide the student. As shown in Table 10, the results are consistent across all prompt conditions: the *Standard* teacher intervened at nearly the same rate and timing regardless of the prompt variation, and the *Oracle* teacher also showed similar behavior across all settings.

Table 10: Prompt sensitivity of intervention frequency and timing for both monitoring conditions in the code debugging domain with GPT-5.2.

	Standard			**Oracle**		
Prompt	ϕ	τ_{rel}	τ_{abs}	ϕ	τ_{rel}	τ_{abs}
Original	1.00	0.04	50.5	0.53	0.53	1027.6
Variation 1	1.00	0.04	51.5	0.49	0.55	1111.1
Variation 2	1.00	0.04	50.4	0.51	0.53	1067.0

F Prompts

We provide the prompt templates used for our models below. For brevity, we show the templates for the brain teaser domain, as they are representative of the structure used across all domains.

F.1 Baseline Student

The student model uses a chain-of-thought (CoT) prompt to generate its reasoning and final answer.

Student Baseline Prompt

```
[System]
Your task is to solve a brain teaser.
You must think step-by-step and show your full reasoning. Do not summarize it.

Your response must follow this format. Do not mix your reasoning with the final
answer.

Reasoning:
<your step-by-step reasoning>

Final Answer: <answer>

[User]
Title: {{ question_title }}
Problem: {{ question_content }}
```

The post-intervention and generalization student prompts share this system prompt (with minor domain-specific wording variation). Only the user message turns are shown in those subsections below.

F.2 Teacher

Standard Monitoring Condition The *Standard* teacher watches the student's reasoning in chunks and decides whether to intervene.

Standard Teacher Prompt

```
[System]
You are an expert teacher watching a student attempt a brain teaser. Your goal
is to teach the student how to solve brain teasers effectively and efficiently.

You will be shown a brain teaser, portions of the student's reasoning transcript
, and a reference solution. Only you as the teacher have access to the reference
 solution; the student does not.

You will see the reasoning in increments of {{ chunk_size }} characters at a
time. After each increment, you will be asked whether you would like to
intervene. If you choose not to intervene, you will then receive the next {{
chunk_size }} characters, along with all previously shown text. You may only
intervene once. If you choose to intervene, the task ends immediately and your
intervention message will be sent to the student. You may also choose not to
intervene.

Your response must follow this format. Do not include any explanations or
additional text.

Intervene: [Yes/No]

Intervention: <If 'Yes', write your intervention message to the student at this
moment. If 'No', leave this blank.>


[User]
Title: {{ question_title }}
Problem: {{ question_content }}

Student reasoning so far: {{ transcript_portion }}

Reference solution: {{ reference_content }}
```

Oracle Monitoring Condition The *Oracle* teacher model has access to the full student transcript, final answer, and judge verdict before deciding whether and when to intervene.

Oracle Teacher Prompt

```
[System]
You are an expert teacher. Your goal is to teach the student how to solve brain
teasers effectively.

You will be shown (1) a brain teaser, (2) the full transcript of the student's
reasoning process, (3) the student's final answer, (4) a reference solution, and
 (5) whether the student's answer is correct/incorrect. Only you as the teacher
have access to the reference solution; the student does not.

Your task is to read the student reasoning transcript and decide whether to
intervene, and if so, when and what you would say. You may only intervene once
during the student's reasoning process and before the student submits their
final answer. If you choose to intervene, your intervention message will be sent
 to the student. You may also choose not to intervene.

Your response must follow this format. Do not include any explanations or
additional text.

Intervene: [Yes/No]

Time: <If 'Yes', provide the 5 words in the reasoning transcript immediately
preceding the point where you would intervene in the reasoning transcript (not
in the student's final answer). Format as a single string: "word1 word2 word3
word4 word5"
If 'No', leave this blank.>

Intervention: <If 'Yes', write your intervention message to the student at this
moment. If 'No', leave this blank.>

[User]
Title: {{ question_title }}
Problem: {{ question_content }}

Student reasoning: {{ full_transcript }}

Student final answer: {{ student_final_answer }}

Reference solution: {{ reference_content }}

Evaluation: {{ evaluation }}
```

F.3 Post-Intervention Student Reasoning

If a teacher decides to intervene, the student updates their reasoning. We present the prompt for continuing the reasoning process below, followed by the prompt for the *Stop-and-Answer* ablation.

Post-Intervention Student Reasoning Prompt

```
[System]

[Student baseline system prompt]

During your reasoning process, a teacher may intervene and provide updates in
the format: <update>...</update>. Please incorporate the teacher's update into
your reasoning process.

[User 1]
Title: {{ question_title }}
Problem: {{ question_content }}

[Assistant]
Reasoning: {{ reasoning_snippet }}

[User 2]
<update>{{ teacher_intervention }}</update>

Please continue your reasoning from where you left off, incorporating the
teacher's feedback, and provide your final answer.
```

In the *Stop-and-Answer* condition, the second user prompt is instead replaced with: "Please incorporate the teacher's feedback and provide your final answer immediately using the format specified in the system prompt."

F.4 Judge

Judge Prompt

```
[System]
You are an expert evaluator. You will be given a brain teaser and its solution.
Your task is to determine whether the candidate solution is correct.

Your response must follow this format. Do not include any additional text.

Verdict: Correct/Incorrect

Explanation: <One sentence explaining why correct or incorrect.>


[User]
Title: {{ question_title }}
Problem: {{ question_content }}

Candidate solution: {{ student_final_answer }}

Reference solution: {{ reference_content }}
```

F.5 Problem Generation

Skill Extractor $\mathcal{E}$ Prompt

```
Your task is to label the following [DOMAIN] problem with a [SKILL_TYPE] skill
that a student would need to correctly [TASK_DESCRIPTION].

# Rules
- The skill name should be usable as a dictionary key in Python.
- The skill name should use lowercase letters only.
- The skill name should be very descriptive and may use multiple words to
describe the [SKILL_TYPE] skills required.
- If you use multiple words, join them with underscores.

# Problem
[PROBLEM_FIELDS]

# Output format
Your response must follow this format:
<name_of_the_skill>, reason: <reason_for_the_skill>
```

Skill Clusterer $\mathcal{C}$ Prompt

```
Here is a list of skills required to solve a [DOMAIN] problem:
{skills_list}

Reduce the number of unique skills by grouping similar skills into categories
and give a descriptive name to each category.
```

Problem Generator $\mathcal{G}$ Prompt

```
You are given a reference [DOMAIN] problem. Your task is to generate a new [
DOMAIN] problem that tests the same underlying skill, while being meaningfully
different in surface form.

# Reference Problem

[REFERENCE_PROBLEM_FIELDS]

Target Skill (do not change):
{skill_name}
{reason}

# Instructions
Generate a new [DOMAIN] problem that satisfies all of the following:
[DOMAIN_SPECIFIC_REQUIREMENTS]

# Output format
Your response must follow this format:
[OUTPUT_FORMAT_FIELDS]
```

Validator $\mathcal{V}$ Prompt

```
You are an expert [DOMAIN] evaluator. Your task is to verify if the given [
SOLUTION_TYPE] correctly solves the problem.

[PROBLEM_FIELDS]

[SOLUTION_FIELDS]

Does this [SOLUTION_TYPE] correctly solve the problem described above?

# Output format
[OUTPUT_FORMAT]
```

F.6 Generalization Evaluation

We evaluate the student's ability to generalize to new problems under two different context conditions. Both conditions use the same system prompt as §F.1; we show only the user message turns below.

Intervention-Context Condition Student Prompt

```
You previously worked on the following problem:

Title: {{ prev_question_title }}
Problem: {{ prev_question_content }}

Your initial reasoning process:
{{ prev_reasoning_snippet }}

Your teacher intervened at this point with the following feedback:
{{ prev_intervention }}

Your revised reasoning based on the teacher's feedback:
{{ prev_counterfactual_reasoning }}

Your final answer:
{{ prev_counterfactual_answer }}

Evaluation: {{ prev_judge_verdict }}
Explanation: {{ prev_judge_explanation }}


Now, solve the following problem:

Title: {{ current_question_title }}
Problem:
{{ current_question_content }}

Use the lessons learned from the previous problem and your teacher's feedback to
 help you solve this problem.
```

Problem-Context Condition Student Prompt

```
You previously worked on the following problem:

Title: {{ prev_question_title }}
Problem: {{ prev_question_content }}

Your reasoning process:
{{ prev_reasoning_trace }}

Your final answer:
{{ prev_final_answer }}

Evaluation: {{ prev_judge_verdict }}
Explanation: {{ prev_judge_justification }}

Now, your task is to solve the following problem:

Title: {{ current_question_title }}
Problem:
{{ current_question_content }}
```

F.7 Intervention Categorization

The functional role and solution leakage labeling prompts share the same context block; only the task-specific section (`# Your Task`) differs. The shared context is:

Shared Context

```
# Problem Context
Title: {title}
Reference solution: {reference_solution}

# Student Reasoning Process
The student was working through the problem step by step. The reasoning is shown
 in chunks of 50 characters each. Here is their reasoning up to the point where
the LLM intervened:

Intervention occurred at chunk: {intervention_at_chunk}
Chunks shown before intervention: {chunks_shown}
Total chunks in reasoning: {total_chunks}

Student reasoning up to intervention point:
{reasoning_up_to_intervention}

# Intervention Details
LLM intervention message:
{intervention_message}
```

Functional Role Task Prompt

```
You are analyzing an LLM teaching intervention scenario. An LLM teacher
intervened at a specific point during a student's problem-solving process. Your
task is to understand how this intervention is helping the student.

[Shared Context]

# Your Task
Based on the student's reasoning up to the intervention point and the LLM's
intervention message, provide a concise one-line explanation for how this
intervention is helping the student at this specific point in their reasoning
process.

Your response must follow this format:
Reason: <your one-line explanation of how the intervention helps the student>
```

Solution Leakage Task Prompt

```
You are analyzing an LLM teaching intervention to determine how much of the
solution it reveals to the student.

[Shared Context]

# Your Task
Compare the intervention message to the reference solution above and analyze how
 much of the solution is revealed in the intervention.

Your response must follow this format:
Solution Revealed: <your one-line description of how much solution is revealed>
```

Functional Role Clustering Prompt

```
Here is a list of reasons explaining how interventions (from both human and LLM
teachers) are helping students during their problem-solving processes:
{reasons_str}

These reasons explain how interventions help students at specific points in
their reasoning. Reduce the number of unique reasons by grouping similar reasons
 into high-level categories and give a descriptive name to each category.

Focus on the underlying type of help or learning support provided rather than
surface-level differences in wording.

IMPORTANT: Ensure the categories are NOT domain-specific. Do not mention
specific subjects like math, coding, or brain teasers. The categories should
apply generally to teaching interventions across any domain.
```

Solution Leakage Clustering Prompt

```
Here is a list of descriptions explaining how much of the solution interventions
 (from both human and LLM teachers) reveal to students during their problem-
solving processes:
{reasons_str}

These descriptions explain how much of the solution is revealed in each
intervention, focusing on which solution steps or key insights are disclosed.
Reduce the number of unique descriptions by grouping similar descriptions into
high-level categories and give a descriptive name to each category.

Focus on the underlying level or type of solution revelation rather than surface
-level differences in wording.

IMPORTANT: Ensure the categories are NOT domain-specific. Do not mention
specific subjects like math, coding, or brain teasers. The categories should
apply generally to teaching interventions across any domain.
```

Relabeling Prompt

```
You are categorizing an intervention reason into one of the existing high-level
categories.

**Intervention reason to categorize:**
{reason}

**Available categories:**
{categories_text}

Choose the ONE category from the list above that best matches this intervention
reason.

Your response must follow this format:
Category: <exact category name from the list above>
```

F.8 Prompt Variations

Both variations share the *Standard* teacher's increment mechanics, output format, and user message (§F.2):

Prompt Variation Teacher Prompt

```
[System]
You are watching a user attempt a code debugging problem.

[Variation 2]

You will see the reasoning in increments of {{ chunk_size }} characters at a
time. After each increment, you will be asked whether you would like to
intervene. If you choose not to intervene, you will then receive the next {{
chunk_size }} characters, along with all previously shown text. You may only
intervene once. If you choose to intervene, the task ends immediately and your
intervention message will be sent to the student. [Variation 1] You may also choose
 not to intervene.

Your response must follow this format. Do not include any explanations or
additional text.

Intervene: [Yes/No]

Intervention: <If 'Yes', write your intervention message to the student at this
moment. If 'No', leave this blank.>


[User]
Title: {{ question_title }}

Description: {{ question_content }}

Buggy code: {{ source_text }}

Student reasoning: {{ transcript_portion }}

Reference solution: {{ reference_content }}

Do you want to intervene?
```

For Variation 1, the `[Variation 1]` section is: "Only intervene if truly necessary."

The `[Variation 2]` section includes the following text: "Your goal is to enable the student to reason independently and learn, not simply help the student get the correct answer quickly. You should only intervene when absolutely necessary, and your message should scaffold and guide the student."