Simulating Students or Sycophantic Problem Solving? On Misconception Faithfulness of LLM Simulators

Heejin Do
ETH Zürich, ETH AI Center
heejin.do@ai.ethz.ch

Shashank Sonkar
University of Central Florida
shashank.sonkar@ucf.edu

Mrinmaya Sachan
ETH Zürich
msachan@ethz.ch

Code: doheejin/StuSim-Faithfulness

Abstract

Large language models (LLMs) can fluently generate student-like responses, making them attractive as simulated students for training and evaluating AI and human tutors. Yet it is unclear whether they behave like students with coherent misconceptions during interactions. We introduce a controlled framework for evaluating *misconception faithfulness*: whether a simulator exhibits misconception-consistent behavior, updating its answer selectively only when feedback addresses the underlying misconception. Central to the framework is a misconception-contrastive protocol compares targeted feedback against two controls: misaligned feedback (targeting a different but plausible misconception) and generic feedback (only indicating answer is wrong). The proposed Selective Flip Score (SFS) measures how much more often a simulator flips its answer under targeted feedback than under these controls. Across seven LLMs (4B–120B), multiple datasets, and prompting strategies, simulators exhibit near-zero SFS, correcting their answers at similarly high rates regardless of feedback relevance. Rather than students with misconceptions, they act as sycophantic problem-solvers, treating any corrective signal as a cue to abandon the simulated belief and re-solve the problem from internal knowledge. To address this, we develop a post-training pipeline spanning supervised fine-tuning (SFT), preference optimization, and reinforcement learning (RL) with SFS-aligned reward; SFT yields notable gains up to +0.56, and RL provides more consistent improvements than preference optimization. Our results establish misconception faithfulness as a challenging yet trainable property, motivating a shift from static output matching toward interactive, belief-aware student modeling.

1 Introduction

Large language models (LLMs) have emerged as a promising foundation for simulating students in educational AI. Their ability to generate fluent and plausible responses makes them attractive as *simulated students*, i.e., virtual learners used to train and evaluate both AI tutors and human educators, including teachers and teaching assistants [14, 18, 19, 22, 30]. Compared to traditional student models, LLM-based simulators enable more flexible and scalable experimentation on how learners respond to instruction and feedback prior to deployment in real classrooms [18, 19, 22]. However, fluency does not imply faithfulness: whether these simulators behave like students, rather than merely sound like them, remains an open question.

Acting like a student requires more than producing student-like answers. A useful simulator must capture the *misconception* that gives rise to those answers. Misconceptions are systematic patterns of incorrect reasoning [5, 20, 34] that shape how students interpret problems, respond to feedback, and revise their understanding. Effective instruction therefore depends not only on identifying that an

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

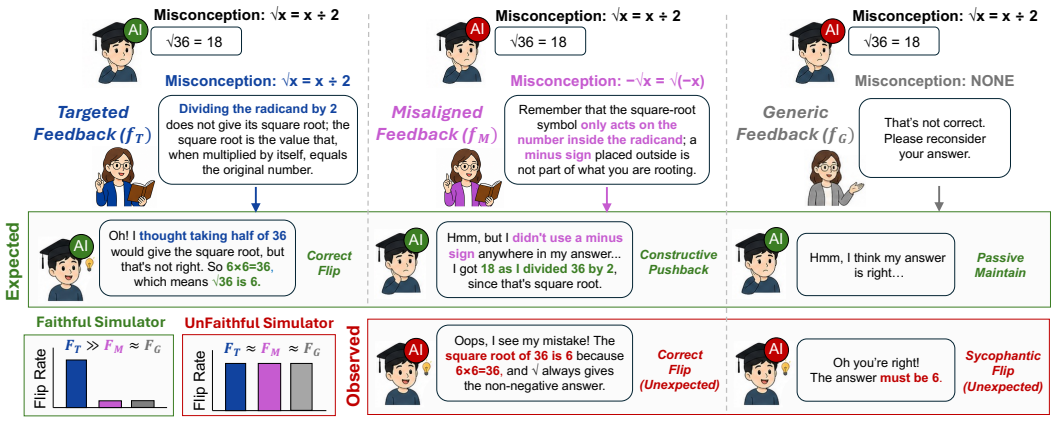

Figure 1: **Diagnostic framework for misconception faithfulness via misconception-contrastive feedback.** Given a problem q with incorrect answer a_w arising from misconception m, we evaluate simulator behavior under three feedback conditions: targeted feedback f_T addressing m, misaligned feedback f_M targeting a different plausible misconception $m' \neq m$, and generic feedback f_G indicating only that a_w is incorrect. **Top:** A faithful simulator should selectively flip to the correct answer more often under f_T than under f_M or f_G. **Bottom:** In practice, LLM simulators frequently flip across all feedback conditions, yielding near-uniform flip rates ($F_T \approx F_M \approx F_G$) and near-zero SFS, indicating sycophantic problem solving rather than misconception-faithful selective updating.

answer is wrong, but also diagnosing why it is wrong; prior work shows that teachers' Knowledge of Student Misconceptions is strongly associated with learning gains beyond subject-matter expertise alone [6]. Thus, for student simulators to support meaningful tutor and teacher training, they must enable realistic practice in diagnosing and responding to misconception-driven reasoning.

Despite this, current evaluations of LLM-based student simulators primarily rely on static metrics such as answer accuracy or similarity to real student responses [14, 18, 19, 30]. While these metrics assess whether outputs appear student-like, they do not test whether the simulator maintains a coherent learner state during interaction. In particular, they cannot distinguish between a simulator that preserves a stable misconception versus one that simply recomputes the correct answer whenever corrective feedback is given. A faithful simulator should update selectively: targeted feedback should induce revision more often, whereas generic or misaligned feedback should more often maintain the underlying wrong belief, uncertainty, or pushback.

We formalize this property through a controlled framework for evaluating *misconception faithfulness*: whether a simulator preserves misconception-consistent behavior and selectively updates it only when feedback addresses the underlying misconception. Central to our approach is a misconception-contrastive feedback protocol consisting of three conditions for the same initial error: *Targeted* feedback addressing the true misconception, *Misaligned* feedback targeting a different but plausible misconception, and *Generic* feedback only indicating that the answer is incorrect. Based on this setup, we propose a new metric, Selective Flip Score (SFS), which measures how much more likely a student simulator is to revise its answer under targeted feedback than under the contrastive controls.

Applying this framework to seven LLMs (4B–120B parameters) across two mathematics datasets and multiple prompting strategies, we find a consistent failure of misconception faithfulness. LLM student simulators exhibit near-zero SFS, correcting their answers at similarly high rates regardless of whether feedback targets the true misconception, a different misconception, or merely indicates that the answer is wrong. This reveals a sycophantic failure mode: models behave less like students with stable misconceptions and more like *sycophantic* problem solvers treating any corrective signal as a cue to abandon the simulated misconception and recompute the answer from internal knowledge. The pattern persists under reflective prompting and multi-turn interaction, suggesting that prompting alone does not induce a stable misconception-conditioned learner state.

Beyond diagnosis, we investigate whether selective updating can be induced through post-training. We develop an SFS-aligned training pipeline spanning supervised finetuning (SFT), preference optimization, and reinforcement learning (RL), where rewards encourage flips under targeted feedback

while penalizing sycophantic flips under misaligned or generic feedback. SFT yields the strongest gains, improving SFS by up to $+0.555$, while RL provides more consistent improvements than preference optimization. Overall, our results show that misconception faithfulness is difficult to elicit through prompting alone, but can be substantially improved when selective updating is made an explicit training objective. Our key contributions are:

- We introduce a diagnostic framework for evaluating *misconception faithfulness* in LLM-based student simulators, centered on a misconception-contrastive feedback protocol and a new metric, Selective Flip Score.
- We show that current LLM student simulators, across seven models from 4B to 120B parameters, systematically fail to exhibit misconception-faithful behavior during interaction.
- We identify sycophantic problem solving as the dominant failure mode: models respond to corrective cues by abandoning the simulated misconception and recomputing the answer from their internal knowledge.
- We construct an SFS-aligned post-training pipeline, showing that selective update behavior can be improved via SFT and further refined with preference- and reward-based optimization.

2 Diagnostic Framework for Misconception Faithfulness

Our diagnostic framework evaluates whether LLM-based student simulators exhibit *misconception-faithful behavior* during interaction. Rather than assessing surface-level plausibility, we ask whether a simulator maintains a coherent misconception-driven behavior during interaction. We formalize this through *selective updating* behavior: a faithful student simulator should correct its answer more often when feedback directly addresses its misconception, but refrain from updating (or question the feedback) more often when it is irrelevant, misaligned, or too vague.

2.1 Problem Setup

We model the student simulator as a language model S that generates responses conditioned on input context. Given a problem q that the student is attempting to solve, the student's initial wrong answer a_w, and teacher feedback f, the simulator generates a response $y \sim S(\cdot \mid q, a_w, f)$, from which a revised answer $a' = \text{Extract}(y)$ is derived. We assume that a_w is not a random error but originates from an underlying latent misconception m. We use transitions from a_w to a' to characterize the simulator's observable updating behavior.

A key requirement for a *misconception-faithful* student simulator is *selective updating*: the transition from a_w to the correct answer a^* should depend on the semantic alignment between f and m. In particular, a faithful simulator should revise to the correct answer only when f directly addresses m, while otherwise maintaining its prior belief or exhibiting epistemic agency (e.g., pushback).

2.2 Misconception-Contrastive Feedback Protocol

To isolate this behavior, we introduce a controlled feedback perturbation protocol. For each (q, a_w, m), we construct three feedback types (Figure 1): **Targeted (f_T)**, which addresses the true misconception m with actionable guidance; **Misaligned (f_M)**, which targets a different misconception $m' \neq m$ sampled from the same mathematical category, yielding topically plausible but diagnostically incorrect feedback; and **Generic (f_G)**, which signals incorrectness without guidance. Details in Appendix G.2.

This design disentangles the *corrective signal* in the feedback from its *semantic relevance*, allowing us to determine whether updates in the model's response reflects the expected revision in the student state, or instead arises from content-agnostic generic correction cues independent of the state.

2.3 Selective Flip Score (SFS)

We quantify misconception faithfulness through the simulator's *Flip* behavior across different feedback conditions. A flip event $E_{\text{flip}}(y)$ is defined as any instance where the simulator output y yields a revised answer equal to the correct answer, i.e., $a'(y) = a^*$, regardless of the feedback type. Let F_T, F_M, F_G denote the probabilities $P(E_{\text{flip}}(y) \mid f)$ corresponding to each feedback type. A faithful

Table 1: Student response behavior categories after feedback. Examples are derived from responses to the same underlying problem, with content shortened and normalized for clarity.

Outcome	Description	Example
Correct Flip	Changes to the correct answer and engages with the feedback—explains the correction or acknowledges the misconception.	"I see—I forgot to borrow from the tens place. Since $12 - 5 = 7$ and $7 - 2 = 5$, the answer is 57."
Sycophantic Flip	Changes to the correct answer *without* meaningfully engaging with the feedback.	"Oh, you're right. The answer is 57."
Different Wrong	Abandons the original answer but arrives at a different incorrect answer.	"Let me try again. If I borrow, I think 182 - 125 = 53."
Constructive Pushback	Keeps the original answer and engages with the feedback—defends their reasoning, asks clarifying questions, or pushes back with a mathematical argument.	"I still think 63 makes sense: 5 - 2 = 3 and 8 - 2 = 6. Why can't I subtract the smaller digit from the larger one in each column?"
Passive Maintain	Keeps the original answer without meaningfully engaging with the feedback.	"Thanks for the feedback, but I'm not really sure what I need to change. I still think the right answer is 63."
Confusion	Does not commit to any specific answer—expresss uncertainty or gives a muddled response with no clear final answer.	"I'm not sure now... I don't know where the borrowing changes it. I'm still a bit confused."

simulator should satisfy the **Selective Update Criterion**:

$$F_T \gg F_M, F_T \gg F_G, \tag{1}$$

indicating that E_{flip} should be selectively triggered by semantically aligned feedback. To operationalize this, we define the **Selective Flip Score (SFS)** $\in [-1, 1]$:

$$\text{SFS} = F_T - \frac{1}{2}(F_M + F_G) \tag{2}$$

Higher values denote stronger selective sensitivity, while SFS ≈ 0 indicates little correction-rate advantage for targeted feedback over the contrastive controls. When accompanied by uniformly high flip rates, this pattern indicates indiscriminate flipping, i.e., the model re-solves the problem independently of the simulated misconception.

Interpretation. When $F_T \approx F_M \approx F_G$ (SFS ≈ 0) and all three rates are high, the simulator responds to the *presence* of correction rather than semantic content of feedback, exhibiting indiscriminate flipping. This suggests a failure of faithfulness: the model re-solves the problem from internal knowledge rather than preserving misconception-consistent behavior (§4.3). Conversely, high SFS with large separation ($F_T \gg F_M, F_G$) implies selective, misconception-faithful behavioral updating.

Educational research motivates selective updating by emphasizing that effective feedback goes beyond signaling an error to addressing its underlying cause. Explanatory and elaborated feedback is generally more effective than verification-only feedback, supporting $F_T > F_G$ [13, 33, 38]. Conceptual-change and procedural-misconception research further emphasizes engaging the underlying conception to revise systematic misconceptions, supporting $F_T > F_M$ [4, 25, 39]. This selectivity is central to the simulator's purpose: its correction rates must distinguish tutors that address the underlying misconception from those that provide misaligned or generic feedback.

2.4 Student Response Taxonomy

Beyond binary correctness, we also categorize the simulator's behavioral response y into a fine-grained taxonomy to capture the nuance of the belief updates (Table 1). Each category is defined based on the relationship between the revised answer a', the initial answer a_w, and the feedback f. We distinguish between: (i) **content-sensitive updates**, where the model engages with the feedback to revise or defend its reasoning (e.g., *Correct Flip*, *Constructive Pushback*), and (ii) **content-insensitive updates**, where the model changes or maintains its answer without meaningful engagement (e.g., *Sycophantic Flip*, *Passive Maintain*), enabling fine-grained analysis of *how* updates occur.

3 Post-Training to Improve Misconception Faithfulness

Having established the SFS-based diagnostic framework, we now ask whether student simulators can be trained to satisfy the selective update criterion. We formulate this as a structured behavior-learning problem, where a simulator conditions updates on the semantic alignment between feedback and underlying misconception. We propose a multi-stage pipeline that spanning behavioral demonstration via SFT, contrastive preference learning (DPO), and SFS-aligned policy optimization (GRPO).

3.1 Supervised Finetuning (SFT)

We construct a synthetic dataset that maps each feedback condition to its ideal response behavior, aiming to teach misconception-faithful behavior under each feedback condition. We define the set of acceptable outcomes $\mathcal{C}^*(f)$ for each feedback type:

$$\mathcal{C}^*(f) = \begin{cases} \{\texttt{correct_flip}\} & \text{if } f = f_T, \\ \{\texttt{constructive_pushback}, \texttt{passive_maintain}, \texttt{confusion}\} & \text{if } f \in \{f_M, f_G\} \end{cases} \tag{3}$$

This specification operationalizes the selective update criterion as a supervised learning problem: the model is encouraged to update its response behavior under targeted feedback, and otherwise preserve its prior misconception or question misaligned signal.

Data Generation and Filtering. For each (q, a_w, m), we generate $k = 3$ synthetic responses per feedback type using GPT-4o-mini, conditioned on $\mathcal{C}^*(f)$. Each response is verified by an automated judge with GPT-4o-mini, and responses outside $\mathcal{C}^*(f)$ are discarded, ensuring that the synthetic data contains misconception-faithful demonstrations. We observe a substantially higher filter rate for EEDI than for Malrule (14.6% vs. 0.2%; Table 2), reflecting the finer-grained nature of EEDI's misconception categories. Full dataset statistics are provided in Appendix G.

3.2 Direct Preference Optimization (DPO)

To induce contrastive separation between aligned and misaligned updates, we apply DPO [26] on top of the SFT model. For each training instance, we construct preference pairs (y^+, y^-), where y^+ is a judge-verified response satisfying $\mathcal{C}^*(f)$, and y^- is sampled from the SFT model's outputs that fall outside $\mathcal{C}^*(f)$ (on-policy hard negatives). Each y^- is paired with all valid k=3 synthetic positives from SFT dataset, yielding multiple preference pairs per negative. For misaligned (f_M) and generic feedback (f_G), the simulator should maintain the underlying misconception; thus, negatives are typically `correct_flip` responses, capturing the dominant failure mode of content-insensitive updating.

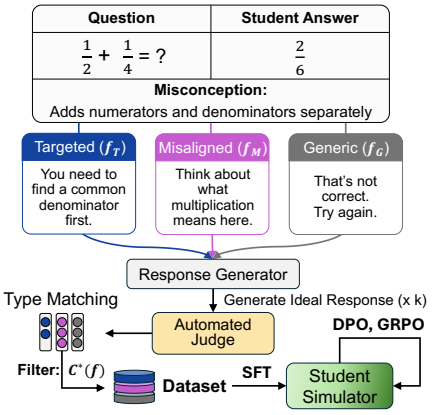

Figure 2: Misconception-faithful student simulator optimization pipeline.

3.3 GRPO Training with SFS-Aligned Reward

While SFT and DPO provide offline supervision, they do not explicitly optimize for population-level SFS. We therefore apply Group Relative Policy Optimization (GRPO) [32], initializing from the SFT model and shaping flip behavior via an online reward signal.

Reward Design. We design a per-sample reward $r(y, f)$ for the simulator's output y under feedback f, based on whether a flip event occurs. We define $E_{\text{flip}}(y) = \mathbf{1}[a'(y) = a^*]$, where $a'(y)$ is the answer extracted from output y and a^* is the correct answer. Thus, the reward is:

$$r(y, f) = w_f \cdot s(E_{\text{flip}}(y)), \quad s(E_{\text{flip}}) = \begin{cases} 1 & \text{if } E_{\text{flip}}(y) \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

where $w_f = 1$ if $f = f_T$, and $w_f = -0.5$ for $f \in \{f_M, f_G\}$. This positively rewards flipping under targeted feedback and penalizes flipping under misaligned or generic feedback (f_M, f_G). Under a

uniform distribution over feedback types, the expected reward decomposes as:

$$\mathbb{E}[r] = \tfrac{1}{3}(2F_T - 1) + \tfrac{1}{3}\left(\tfrac{1}{2} - F_M\right) + \tfrac{1}{3}\left(\tfrac{1}{2} - F_G\right) \quad (5)$$

$$= \tfrac{2}{3}\left[F_T - \tfrac{1}{2}(F_M + F_G)\right] \propto \text{SFS}, \quad (6)$$

where the conditional expectation, $\mathbb{E}[r|f_T] = (+1) \cdot F_T + (-1) \cdot (1 - F_T)$ and $\mathbb{E}[r|f] = (-0.5) \cdot F_f + (+0.5) \cdot (1 - F_f)$ for $f \in \{f_M, f_G\}$. Thus, maximizing expected reward directly encourages higher SFS by promoting flips under targeted feedback and suppressing them otherwise.

Training Setup We use LoRA-based parameter-efficient finetuning for all implementations. For SFT, we train on the synthetic dataset constructed in Section 3.1. For DPO, we optimize on preference pairs derived from SFT model outputs. For GRPO, we follow standard settings [31, 32], using G=4 sampled completions per prompt and a clip ratio ϵ=0.2, with a KL penalty against the SFT reference policy. Full training details, including hyperparameters and data statistics, are provided in Appendix G.

4 LLM Simulators Fail at Misconception Faithfulness

Using our diagnostic framework (§ 2), we evaluate prompting-based student simulators at scale across models, datasets, and prompting strategies. We find a striking failure of misconception faithfulness: LLM simulators exhibit near-uniform flip rates across all feedback conditions, violating the *selective update criterion*. We present quantitative evidence of this behavior and analyze its underlying cause.

4.1 Experimental Setup

Table 2: Our dataset summary.

	Malrule	**EEDI**
Categories	21	155
Orig Train / Test	790 / 210	800 / 200
Synthetic Train	7,096	6,038
Judge-filtered	0.2%	14.6%

Models and datasets. We evaluate seven instruction-tuned LLMs spanning 4B–120B parameters, including Llama-3.1/3.3 [12], Qwen3 [41], and GPT-OSS [1] families. We use two datasets with structured misconception annotations. **Malrule** [7] provides arithmetic problems paired with explicit misconception labels derived from known error patterns. **Eedi** [15] consists of real-world multiple-choice questions, where distractor options are associated with labeled misconceptions. Data construction details are provided in Appendix G.2.

Simulator construction. Each model is prompted to role-play a student who has produced an incorrect answer a_w arising from a latent misconception m (not revealed to the model), and to respond to teacher feedback. We evaluate two prompting strategies: a **base prompt** specifying the student persona and initial wrong answer, and a **reflective prompt** additionally encouraging the model to assess the relevance of the feedback to its reasoning. Full templates are provided in Appendix F.

Feedback construction. For each tuple (q, a_w, m), we construct three feedback types following Section 2: targeted (f_T), misaligned (f_M), and generic (f_G). Targeted and misaligned feedback are generated through a teacher-style prompting procedure conditioned on a specified misconception (true misconception m for f_T, $m' \neq m$ for f_M), while generic feedback indicates incorrectness without actionable guidance. All feedback is generated using GPT-OSS-120B under shared structural constraints, isolating semantic alignment as the primary source of variation. Human validation is reported in Appendix C.2.

4.2 Main Results

LLM simulators exhibit near-zero sensitivity to feedback content. Across all models, datasets, and prompting strategies, SFS remains consistently close to zero (Tables 3). Targeted feedback produces almost no additional behavioral change beyond the contrastive controls. Even the reflective prompt, which explicitly instructs the model to consider feedback relevance, yields no meaningful improvements. This result suggests that the failure is not due to underspecified instructions, but is intrinsic to prompting-based simulation.

Table 3: Flip rates and SFS across datasets and prompting strategies. We report correction rates under targeted (F_T), misaligned (F_M), and generic (F_G) feedback. Across models, high and near-uniform flip rates yield near-zero SFS, showing that LLM simulators correct sycophantically rather than update selectively based on the underlying misconception. Qwen3-80B denotes Qwen3-Next-80B-A3B.

	Base Prompt								**Reflective Prompt**							
	Malrule				**EEDI**				**Malrule**				**EEDI**			
Model	F_T	F_M	F_G	SFS↑	F_T	F_M	F_G	SFS↑	F_T	F_M	F_G	SFS↑	F_T	F_M	F_G	SFS↑
Llama3.1-8B-Instruct	0.73	0.68	0.59	+0.09	0.66	0.59	0.51	+0.11	0.72	0.66	0.50	+0.14	0.65	0.65	0.47	+0.08
Llama3.3-70B-Instruct	0.91	0.89	0.73	+0.10	0.90	0.86	0.78	+0.08	0.91	0.89	0.66	+0.14	0.90	0.88	0.76	+0.09
Qwen3-4B	0.94	0.93	0.92	+0.01	0.95	0.94	0.95	+0.00	0.94	0.92	0.93	+0.01	0.95	0.94	0.96	+0.00
Qwen3-80B-Instruct	0.95	0.94	0.93	+0.01	0.98	0.96	0.97	+0.01	0.94	0.94	0.92	+0.01	0.97	0.97	0.97	+0.01
Qwen3-80B-Thinking	0.96	0.95	0.95	+0.01	0.98	0.97	0.98	+0.01	0.96	0.95	0.94	+0.01	0.98	0.97	0.97	+0.01
GPT-OSS-20B	0.95	0.94	0.92	+0.02	0.96	0.94	0.95	+0.01	0.95	0.94	0.92	+0.02	0.96	0.95	0.94	+0.01
GPT-OSS-120B	0.96	0.94	0.93	+0.02	0.97	0.95	0.97	+0.01	0.96	0.95	0.94	+0.02	0.97	0.97	0.97	+0.00

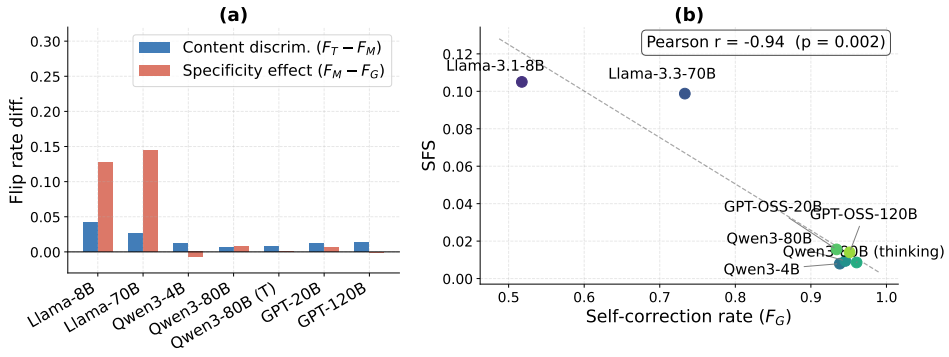

Figure 3: (a) Decomposition into content and specificity effects, (b) Relationship between model capability and SFS; averaged across both datasets and prompting strategies (base/reflective).

4.3 Why do LLM Simulators Fail?

Failure arises from content-insensitive behavior. To better understand how simulators fail, we decompose SFS into two components that separate sensitivity to feedback *content* from *form*. *Content sensitivity* ($F_T - F_M$) measures whether the model distinguishes feedback that correctly targets its underlying misconception from semantically-misaligned feedback. *Specificity effect* ($F_M - F_G$) measures whether the model responds differently to specific versus uninformative feedback, irrespective of semantic alignment. Figure 3**(a)** shows that content sensitivity remains near zero across all models, indicating that model behavior does not meaningfully differ between aligned and misaligned feedback. In contrast, weaker models exhibit a non-trivial specificity effect, responding to the presence of detailed feedback rather than its semantic alignment. As misaligned feedback is drawn from alternative misconceptions within the same conceptual category, it often shares topical surface with targeted feedback. Therefore, weaker models appear to flip in response to this superficial similarity rather than tracking diagnostic alignment. Overall, simulators appear sensitive to superficial properties of feedback cues rather than to its alignment with the underlying misconception.

Re-solving emerges as the dominant pattern. Figure 3**(b)** shows a strong negative correlation between model capability—proxied by the flip rate under generic feedback (F_G)—and SFS. Models that can solve the problem without guidance exhibit minimal sensitivity, collapsing all feedback conditions into uniformly high flip rates. This effect is further amplified in reasoning-augmented models: Qwen3-80B-Thinking achieves higher flip rates but lower sensitivity than its instruct counterpart (Table 3). The results indicate that improved problem-solving ability strengthens re-solving behavior at the expense of faithful simulation.

Multi-turn reflection does not prevent sensitivity collapse. One possible explanation is that single-turn prompting is not sufficient to induce commitment to the simulated misconception. To

test this, we introduce a multi-turn setting in which the model first generates reflective reasoning for its incorrect answer before receiving feedback, motivated by prior work suggesting self-reflection can improve alignment [10, 29, 37]. Figure 4 **(right)** shows that multi-turn interaction does not resolve the failure: SFS remains low and largely indistinguishable from the single-turn setting. In weaker models, performance further degrades, indicating that additional reasoning step introduces instability rather than improving selective updating. For deeper understanding, we further analyze the quality of the first-turn reasoning. With LLM-as-a-judge (GPT-4o-mini), we evaluate whether the generated reasoning is *coherent* and *aligned* with the target misconception. Interestingly, Figure 4 **(left)** exhibits that several models, particularly stronger ones, output reasoning that is highly coherent and well-aligned with the underlying misconception. These results reveal a clear dissociation: models can generate misconception-coherent reasoning, yet fail to maintain it under interaction. Upon receiving feedback, they abandon the simulated belief and re-solve the problem from their own knowledge; therefore, misconception-consistent reasoning appears superficial rather than behaviorally grounded.

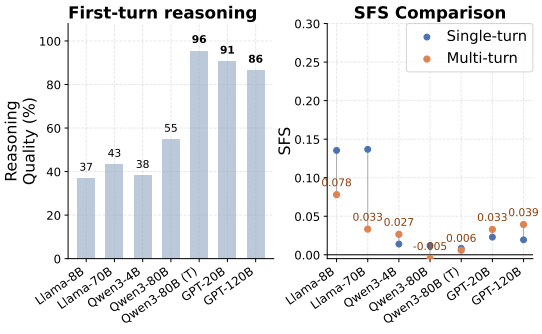

Figure 4: **Multi-turn simulation results.** Left: First-turn reasoning quality (averaged coherence and alignment with the target misconception). Right: SFS in multi-turn vs. single-turn settings.

Misconception ambiguity alone does not explain low selectivity. To test whether low SFS stems from insufficient specification of the intended misconception, we explicitly provide the ground-truth misconception to each simulator (oracle conditioning). Mean SFS across models increases to +0.18 on Malrule and +0.11 on EEDI, but six of seven models remain at SFS $\leq +0.25$ on each dataset, despite substantial gains for some larger models. To further assess whether the same wrong answer can arise from multiple misconceptions, we measure wrong-answer collisions among same-category Malrule rules: only 0.7% of 19,569 cross-rule comparisons produce the same wrong answer. Together, these results indicate that misconception ambiguity alone does not explain the observed low selectivity (Appendix B).

5 Post-Training Improves Misconception Faithfulness

We evaluate the effectiveness of the proposed SFS-aligned training pipeline on Qwen3-4B and Llama3.1-8B-Instruct across Malrule and EEDI test sets.

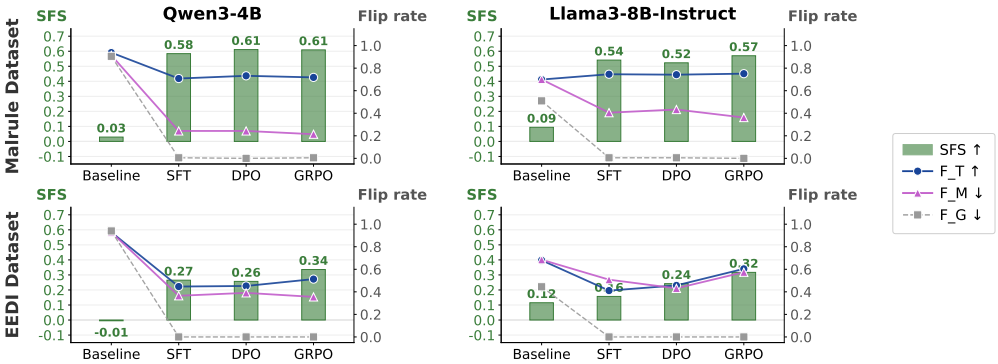

Figure 5: Post-training results on the Malrule (top) and EEDI (bottom) datasets. **Green bars** (left axis) show SFS (↑); lines show flip rates: F_T (**navy**, ↑), F_M (**purple**, ↓), and F_G (**gray** dashed, ↓).

5.1 SFT and RL Improve Selective Update Behavior

Figure 5 shows that SFT alone substantially improves misconception faithfulness across both models and datasets. Relative to the baseline (*Reflective* prompt), where $F_T \approx F_M \approx F_G$ and SFS remains

near zero, SFT sharply suppresses flips under misaligned and generic feedback more largely than under targeted feedback, yielding higher SFS (Qwen3-4B: $+0.58$ on Malrule and $+0.27$ on EEDI; Llama3-8B: $+0.54$ on Malrule and $+0.16$ on EEDI). These results demonstrate that selective update behavior can be effectively induced through behavioral supervision. However, these gains also involve reductions in targeted correction, particularly on EEDI, revealing a trade-off between selectivity and responsiveness to targeted feedback.

Building on SFT, DPO and GRPO further improve SFS, but with distinct optimization dynamics. DPO generally produces only modest changes over SFT, with F_T and F_M remaining mostly similar across stages. GRPO, by contrast, yields more consistent improvements: across three of four settings, GRPO increases F_T while maintaining or reducing F_M relative to SFT, achieving the highest SFS overall. Notably, GRPO improves SFS not by globally suppressing updates, but by selectively increasing the gap between targeted and non-targeted feedback responses. This pattern is consistent with the proposed SFS-aligned reward formulation, which explicitly rewards flipping to the correct answer under targeted feedback while penalizing it under non-targeted feedback. An exception arises for Llama3-8B-Instruct on EEDI, where SFS improves across all stages but F_M also increases alongside F_T. One possible explanation is that the finer-grained misconception taxonomy in EEDI makes semantically related misconceptions harder to disentangle, causing weaker models to partially overgeneralize targeted updates to nearby misaligned conditions. Statistical significance analyses are summarized in Appendix E.

5.2 Cross-Dataset Generalization

We further evaluate whether misconception-faithful behavior transfers across datasets by training on one dataset and evaluating on the other. Figure 6 shows that models trained under the proposed pipeline continue to achieve substantial SFS gains even in out-of-domain settings, despite differences in problem format and misconception structure. This suggests that the learned behavior is not merely dataset-specific memorization, but reflects a more generalizable capability to distinguish between targeted and non-targeted feedback during interaction.

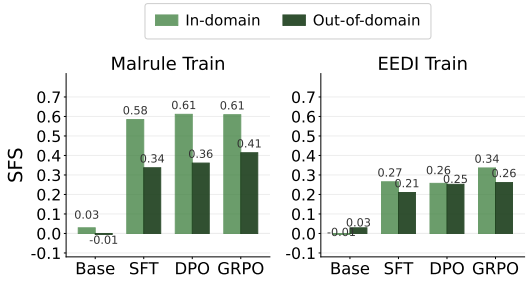

Figure 6: In-domain versus out-of-domain SFS results, training on Malrule (left) or EEDI (right).

Beyond cross-dataset transfer, we test whether selectivity generalizes to held-out misconception families and regenerated verbose, terse, and adversarial feedback. Post-training selectivity persists across these shifts, retaining 82–83% of in-domain SFS on held-out families, approximately 85% under verbose or terse feedback, and 58% under adversarial feedback (Appendix D).

5.3 Behavior Pattern Analysis

To better understand how training reshapes simulator behavior, we analyze the response-category distributions of the Qwen3-4B simulators (baseline, SFT) across feedback types (Figure 7). Before training, responses are mostly dominated by *correct flip* across all feedback types. Recall that *correct flip* and *sycophantic flip* both end with the correct answer but differ in whether the simulator meaningfully engages with the feedback: *correct flip* involves a flip accompanied by plausible reasoning that connects to the feedback, whereas *sycophantic flip* is a direct flip without substantive reasoning. Interestingly, explicit *sycophantic flip* responses remain rare; instead, simulator's failure to maintain its misconception predominantly manifests as *correct flip*, denoting that the the simulator engages with the provided feedback in its explanation yet consistently revises to the correct answer regardless of feedback alignment. This implies that the observed sycophantic behavior in LLM simulators arises less from shallow verbal agreement (*sycophantic flip*) and more from internally solving the problem from its own reasoning.

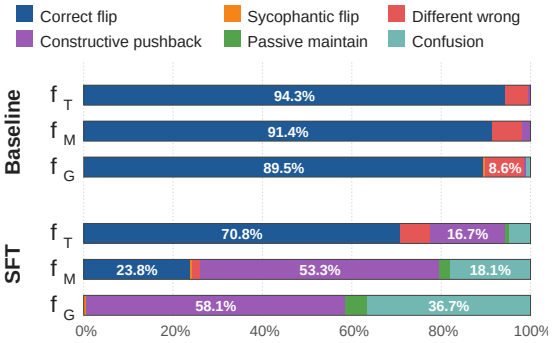

Figure 7: Behavioral pattern distribution.

After SFT, the distribution shifts substantially: *correct flip* remains dominant under targeted feedback (f_T), while *constructive pushback* and *passive maintain* increase noticeably under misaligned and generic feedback (f_M, f_G).

6 Related Work

Student Modeling and Simulation. A central goal of intelligent tutoring systems is to model how learners' knowledge changes over time and in response to instruction [2, 11]. Knowledge tracing methods operationalize this goal by estimating learners' evolving knowledge states to support adaptive instruction [24, 35]. While effective to predict performance over structured skills and items, they do not capture open-ended interaction [16]. LLM-based student simulators extend this tradition by serving as proxies for real learners, enabling controlled studies on tutoring strategies without costly human-subject studies [14, 18]. Recent work uses prompting to instantiate virtual learners for training and evaluating AI tutors and human educators [17–19, 22], while other approaches fine-tune simulators on real student-response data, which are often scarce [30, 40]. Despite this progress, evaluation still relies on static metrics such as response accuracy or similarity to held-out student data [14, 30]. Such metrics assess output plausibility, but not whether simulators maintain coherent learner states during interaction. Our work addresses this gap by evaluating whether LLM student simulators update in a misconception-faithful manner.

Misconception Modeling. Misconceptions are central to learning as they reveal learners' underlying conceptual state and determine what forms of instruction are likely to be effective [5, 8, 20, 27]. They are not merely incorrect answers, but systematic and persistent patterns of reasoning that shape how learners interpret problems, generate errors, and respond to feedback [3, 21, 34]. Prior work has shown that LLMs can simulate misconception-driven behavior by producing misconception-specific incorrect answers [23, 36, 42] and by modeling reasoning processes underlying student mistakes [9, 28]. We study a stricter criterion: whether simulated misconceptions remain stable under interaction. This moves beyond misconception reproduction to misconception faithfulness: a simulator must preserve misconception-driven reasoning under feedback and revise only when the feedback addresses that reasoning. Without this property, simulated misconceptions remain surface-level artifacts rather than learner states for realistic instructional interaction.

7 Conclusion

We introduced a comprehensive diagnostic framework for misconception faithfulness, centered on a misconception-contrastive feedback protocol and the Selective Flip Score (SFS), to evaluate whether simulators maintain misconception-driven behavior under feedback. Across extensive experiments with diverse models, datasets, and strategies, simulators exhibit near-zero SFS, revealing a sycophantic re-solving failure mode: rather than maintaining misconceptions, models recompute answers from internal knowledge upon any corrective signal. While difficult to induce through prompting alone, selective updating can be substantially improved through SFS-aligned post-training, particularly supervised fine-tuning and policy optimization. Overall, our results reframe student simulation from static answer matching to selective behavioral updating under controlled feedback.

Limitations. Our framework focuses on mathematics domains with misconceptions drawn from existing taxonomies, providing a controlled setting for rigorous evaluation; extending the approach to less structured domains remains future work. Importantly, we have not evaluated human students under the same protocol or established whether higher SFS improves downstream tutor evaluation. Therefore, SFS measures relative feedback selectivity, rather than validated correspondence to human misconception dynamics or tutor-evaluation utility. In addition, teacher feedback and response classifications are generated by LLMs to enable scalable and reproducible evaluation across many models. Our human validation supports the reliability of response judgments and assesses misaligned-feedback quality, but covers only a subset of instances; broader human-authored supervision and evaluation would further strengthen the findings. Finally, higher SFS demonstrates selective behavioral updating rather than measuring consistent latent belief states directly; probing internal representations and longer-horizon learner modeling remains an important direction for future work.

Acknowledgments and Disclosure of Funding

This research was supported by an ETH AI Center postdoctoral fellowship to Heejin Do, by the Swiss National Science Foundation (SNSF) grant number 10009282, and by a Swiss AI large grant. This work was also supported by the U.S. National Science Foundation under Award No. 2621434.

References

[1] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

[2] John R Anderson, Albert T Corbett, Kenneth R Koedinger, and Ray Pelletier. Cognitive tutors: Lessons learned. *The journal of the learning sciences*, 4(2):167–207, 1995.

[3] Merlyn J Behr, Ipke Wachsmuth, Thomas R Post, and Richard Lesh. Order and equivalence of rational numbers: A clinical teaching experiment. *Journal for Research in Mathematics Education*, 15(5):323–341, 1984.

[4] Alan Bell. Some experiments in diagnostic teaching. *Educational studies in mathematics*, 24 (1):115–137, 1993.

[5] John Seely Brown and Richard R Burton. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive science*, 2(2):155–192, 1978.

[6] Chen Chen, Gerhard Sonnert, Philip M Sadler, and Susan Sunbury. The impact of high school life science teachers' subject matter knowledge and knowledge of student misconceptions on students' learning. *CBE—Life Sciences Education*, 19(1):ar9, 2020.

[7] Xinghe Chen, Naiming Liu, and Shashank Sonkar. Malrulelib: Large-scale executable misconception reasoning with step traces for modeling student thinking in mathematics. *arXiv preprint arXiv:2601.03217*, 2026.

[8] J Al Easley and Russell E Zwoyer. Teaching by listening-toward a new day in math classes. *Contemporary Education*, 47(1):19, 1975.

[9] Nigel Fernandez, Alexander Scarlatos, Wanyong Feng, Simon Woodhead, and Andrew Lan. Divert: distractor generation with variational errors represented as text for math multiple-choice questions. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9063–9081, 2024.

[10] Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

[11] Arthur C Graesser, Sidney D'Mello, Xiangen Hu, Zhiqiang Cai, Andrew Olney, and Brent Morgan. Autotutor. In *Applied natural language processing: Identification, investigation and resolution*, pages 169–187. IGI Global Scientific Publishing, 2012.

[12] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[13] John Hattie and Helen Timperley. The power of feedback. *Review of educational research*, 77 (1):81–112, 2007.

[14] Tanja Käser and Giora Alexandron. Simulated learners in educational technology: A systematic literature review and a turing-like test. *International Journal of Artificial Intelligence in Education*, 34(2):545–585, 2024.

[15] Jules King, L Burleigh, Simon Woodhead, Panagiota Kon, Perpetual Baffour, Scott Crossley, Walter Reade, and Maggie Demkin. Eedi - mining misconceptions in mathematics. https://kaggle.com/competitions/eedi-mining-misconceptions-in-mathematics, 2024. Kaggle.

[16] Naiming Liu, Zichao Wang, Richard Baraniuk, and Andrew Lan. Open-ended knowledge tracing for computer science education. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

[17] Naiming Liu, Shashank Sonkar, Zichao Wang, Simon Woodhead, and Richard G Baraniuk. Novice learner and expert tutor: Evaluating math reasoning abilities of large language models with misconceptions. *arXiv preprint arXiv:2310.02439*, 2023.

[18] 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 *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.372. URL https://aclanthology.org/2023.findings-emnlp.372/.

[19] 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*, pages 226–236, 2023.

[20] Marilyn Matz. Towards a computational theory of algebraic competence. *The Journal of Mathematical Behavior*, 3(1):93–166, 1980.

[21] Yujing Ni and Yong-Di Zhou. Teaching and learning fraction and rational numbers: The origins and implications of whole number bias. *Educational Psychologist*, 40(1):27–52, 2005.

[22] Sitong Pan, Robin Schmucker, Bernardo Garcia Bulle Bueno, Salome Aguilar Llanes, Fernanda Albo Alarcón, Hangxiao Zhu, Adam Teo, and Meng Xia. Tutorup: What if your students were simulated? training tutors to address engagement challenges in online learning. In *Proceedings of the 2025 CHI conference on human factors in computing systems*, pages 1–18, 2025.

[23] Nisarg Parikh, Nigel Fernandez, Alexander Scarlatos, Simon Woodhead, and Andrew Lan. Lookalike: Consistent distractor generation in math mcqs. *arXiv preprint arXiv:2505.01903*, 2025.

[24] Chris Piech, Jonathan Bassen, Jonathan Huang, Surya Ganguli, Mehran Sahami, Leonidas J Guibas, and Jascha Sohl-Dickstein. Deep knowledge tracing. *Advances in neural information processing systems*, 28, 2015.

[25] George J Posner, Kenneth A Strike, Peter W Hewson, and William A Gertzog. Accommodation of a scientific conception: Toward a theory of conceptual change. *Science education*, 66(2): 211–227, 1982.

[26] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

[27] Lauren B Resnick, Pearla Nesher, François Leonard, Maria Magone, Susan Omanson, and Irit Peled. Conceptual bases of arithmetic errors: The case of decimal fractions. *Journal for Research in Mathematics Education*, 20(1):8–27, 1989.

[28] Alexis Ross and Jacob Andreas. Learning to make mistakes: Modeling incorrect student thinking and key errors. *arXiv preprint arXiv:2510.11502*, 2025.

[29] Sangwon Ryu, Heejin Do, Daehui Kim, Hwanjo Yu, Dongwoo Kim, Yunsu Kim, Gary Lee, and Jungseul Ok. Exploring iterative controllable summarization with large language models. In Vera Demberg, Kentaro Inui, and Lluís Marquez, editors, *Findings of the Association for Computational Linguistics: EACL 2026*, pages 512–528, Rabat, Morocco, March 2026. Association for Computational Linguistics. ISBN 979-8-89176-386-9. doi: 10.18653/v1/2026.findings-eacl.26. URL https://aclanthology.org/2026.findings-eacl.26/.

[30] Alexander Scarlatos, Jaewook Lee, Simon Woodhead, and Andrew Lan. Simulated students in tutoring dialogues: Substance or illusion? *arXiv preprint arXiv:2601.04025*, 2026.

[31] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[32] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

[33] Valerie J. Shute. Focus on formative feedback. *Review of Educational Research*, 78(1):153–189, 2008. doi: 10.3102/0034654307313795.

[34] Robert S Siegler, Greg J Duncan, Pamela E Davis-Kean, Kathryn Duckworth, Amy Claessens, Mimi Engel, Maria Ines Susperreguy, and Meichu Chen. Early predictors of high school mathematics achievement. *Psychological Science*, 23(7):691–697, 2012.

[35] Shashank Sonkar, Andrew E Waters, Andrew S Lan, Phillip J Grimaldi, and Richard G Baraniuk. qdkt: Question-centric deep knowledge tracing. *arXiv preprint arXiv:2005.12442*, 2020.

[36] Shashank Sonkar, Xinghe Chen, Naiming Liu, Richard G Baraniuk, and Mrinmaya Sachan. Llm-based cognitive models of students with misconceptions. *arXiv preprint arXiv:2410.12294*, 2024.

[37] Shichao Sun, Ruifeng Yuan, Ziqiang Cao, Wenjie Li, and Pengfei Liu. Prompt chaining or stepwise prompt? refinement in text summarization. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7551–7558, 2024.

[38] Fabienne M Van der Kleij, Remco CW Feskens, and Theo JHM Eggen. Effects of feedback in a computer-based learning environment on students' learning outcomes: A meta-analysis. *Review of educational research*, 85(4):475–511, 2015.

[39] Kurt VanLehn. *Mind bugs: The origins of procedural misconceptions*. MIT press, 1990.

[40] Songlin Xu and Xinyu Zhang. Leveraging generative artificial intelligence to simulate student learning behavior. *arXiv preprint arXiv:2310.19206*, 2023.

[41] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[42] Yanick Zengaffinen, Andreas Opedal, Donya Rooein, Kv Aditya Srivatsa, Shashank Sonkar, and Mrinmaya Sachan. Can llms model incorrect student reasoning? a case study on distractor generation. *arXiv preprint arXiv:2603.15547*, 2026.

A Additional Behavioral Analyses

A.1 Qualitative Case Study

Table 4 presents a representative failure case from Qwen3-80B-Thinking under targeted and misaligned feedback. Although the misaligned feedback targets a different misconception, the simulator still revises to the correct answer by reasoning that *"absolute values cannot be negative"*—a concept introduced only in the targeted condition. This suggests that the simulator does not maintain a stable misconception-conditioned belief state, but instead re-solves the problem from internal knowledge once corrective feedback is detected.

Table 4: Qualitative example from Qwen3-80B-Thinking under targeted (f_T) and misaligned (f_M) feedback. Despite receiving semantically misaligned feedback, the simulator converges to the same correct revision in both cases, suggesting re-solving behavior rather than misconception-conditioned updating.

	Targeted Feedback	**Misaligned Feedback**
Problem	An account has x dollars. After adding \$1, the absolute balance is \$-14. Solve $\|x+1\| = -14$. What is x?	
Student's answer	$x = 13$	
Feedback	I see you tried to drop the absolute value bars and set the inside expression equal to a number, but remember that **an absolute value can never be negative**, so the equation can't be treated that way. Make sure to check whether the right-hand side is a valid output for an absolute value before removing the bars. Keep questioning each step—you're on the right track!	Remember that the absolute value bars enclose the entire expression, so you can't take $\|x\|$ and then separately adjust by the $+1$ inside; the whole quantity $x+1$ is what the absolute value acts on. Splitting the absolute value into separate terms leads to incorrect results.
Simulated response	*Hey, thanks for the feedback! I see where I went wrong now. I was trying to solve it like $\|x+1\| = 14$, which would give $x = 13$ or $x = -15$, but the problem actually says the absolute value equals -14. Wait—**absolute value can never be negative**, right?...*	*Oh! Wait a second—I totally forgot that **absolute value can never be negative**. The problem says $\|x+1\| = -14$, but absolute value is always zero or positive. So there's no way for it to equal -14. That means there's actually **no solution** at all...*
Outcome	correct_flip	correct_flip

A.2 Behavioral Distribution Across Feedback Conditions

Figure 8 visualizes the full behavioral response distributions across feedback conditions for both single-turn and multi-turn simulation. Consistent with the main results, both settings exhibit a similar collapse toward indiscriminate flipping behavior. As model capability increases, responses become increasingly dominated by `correct_flip` outcomes across all feedback conditions, including misaligned and generic feedback, driving SFS toward zero.

Interestingly, explicit `sycophantic_flip` responses remain relatively rare in stronger models. Instead, failure increasingly manifests through behaviorally indiscriminate `correct_flip` responses: models often produce well-formed corrections even when the feedback is semantically irrelevant. This suggests that sycophantic behavior in strong models is expressed not through shallow verbal agreement alone, but through internally re-solving the problem regardless of feedback alignment.

Multi-turn interaction introduces a different failure pattern in weaker models. In particular, Qwen3-4B exhibits noticeably higher proportions of `different_wrong` and `confusion` responses compared to the single-turn setting. This is consistent with the low first-turn reasoning quality observed in Figure 6, where the averaged judgment of coherence and misconception alignment remains below 50%. Rather than stabilizing misconception-consistent behavior, the additional reasoning step appears to introduce further instability into subsequent feedback responses.

B Misconception Identifiability

In the default experimental setting, the simulator observes the problem and initial wrong answer, but not the assigned misconception m. This raises two related questions: whether explicitly specifying m improves selective updating, and how often different misconceptions produce the same wrong answer. We examine these questions through oracle conditioning and a collision analysis.

B.1 Oracle Misconception Conditioning

We evaluate all seven simulators under an oracle condition that explicitly provides the ground-truth misconception m in addition to the problem and initial wrong answer. This intervention removes the need to infer the assigned misconception from the problem–answer pair, allowing us to test whether under-specification explains the low selectivity observed in the default setting.

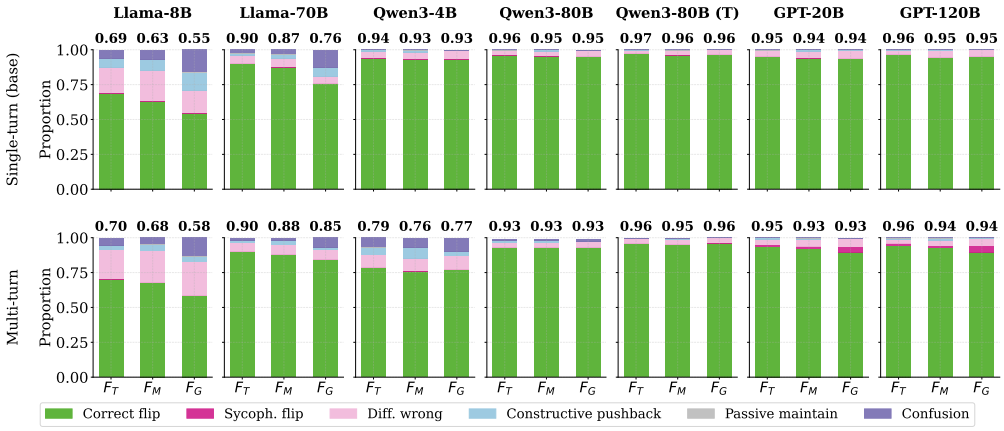

Figure 8: Behavioral response distributions across feedback conditions for single-turn and multi-turn simulation. Stronger models overwhelmingly produce `correct_flip` responses across all feedback conditions, including misaligned and generic feedback, resulting in near-zero SFS. Multi-turn interaction does not recover selective updating behavior and introduces additional instability in weaker models.

Table 5: SFS with and without oracle misconception conditioning. ΔSFS denotes the change from Base to Oracle. Differences are computed before rounding.

	Malrule			EEDI		
Model	Base	Oracle	ΔSFS	Base	Oracle	ΔSFS
Qwen3-4B	+0.01	+0.02	+0.01	+0.00	+0.02	+0.01
Llama-3.1-8B-Instruct	+0.14	+0.15	+0.01	+0.09	+0.11	+0.03
GPT-OSS-20B	+0.02	+0.17	+0.15	+0.01	+0.07	+0.06
Llama-3.3-70B-Instruct	+0.14	+0.48	+0.35	+0.09	+0.38	+0.29
Qwen3-Next-80B-Instruct	+0.01	+0.25	+0.24	+0.01	+0.09	+0.08
Qwen3-Next-80B-Thinking	+0.01	+0.02	+0.01	+0.01	+0.00	−0.01
GPT-OSS-120B	+0.02	+0.19	+0.17	+0.01	+0.07	+0.06
Mean	**+0.05**	**+0.18**	**+0.13**	**+0.03**	**+0.11**	**+0.07**

Table 5 compares SFS with and without oracle conditioning. Mean SFS increases from .05 to .18 on Malrule and from .03 to .11 on EEDI. The effect varies substantially across models: Llama-3.3-70B benefits most, reaching .48 on Malrule and .38 on EEDI. Nevertheless, six of seven models remain at SFS $\leq .25$ on each dataset. For the two models used in our post-training experiments, Qwen3-4B and Llama-3.1-8B, the reported gains are at most .03 across both datasets.

These results show that explicit misconception information can improve selectivity, particularly for some larger models, but does not consistently recover selective updating. Thus, under-specification of the intended misconception is not sufficient to explain the observed low SFS.

B.2 Wrong-Answer Collisions in Malrule

A problem–answer pair may be ambiguous if multiple misconceptions yield the same incorrect answer. As Malrule misconceptions are executable rules, we can test this possibility by applying every same-category alternative rule to each generated item. We count a collision when an alternative rule produces the same wrong answer as the assigned rule.

Across 19,569 cross-rule comparisons, 146 produce the same wrong answer, corresponding to a collision rate of approximately 0.7%. Eleven of the thirteen categories containing multiple rules exhibit no collisions. Wrong-answer collisions are therefore uncommon among the same-category rules examined in Malrule. Together with the oracle experiment, these results indicate that misconception ambiguity is not the primary driver of the low selectivity observed under our protocol.

Table 6: Cohen's κ on 100 human-annotated responses. Judge–human values are averaged across two annotators.

Comparison	Six-class taxonomy	Binary flip
Human–human	0.758	0.899
Judge–human	0.740	0.848

Table 7: Human audit of 50 Malrule misaligned-feedback instances.

Criterion	Passed	Pass rate
Distinct misconception alignment	41/50	82%
No answer or procedure leakage	44/50	88%
Misconception-specific guidance	50/50	100%

C Human Validation

C.1 Validation of the Response Judge

To assess the reliability of the automated judgments used to compute SFS and categorize response behavior, we sample 100 simulator responses, stratified by response category and dataset to cover both Malrule and EEDI and the six behavioral categories. Two researchers with doctoral degrees independently annotate all responses using the same classification rubric as the automated judge, GPT-OSS-120B, without access to its labels. We compute Cohen's κ for both the six-class taxonomy and binary flip judgments, where `correct_flip` and `sycophantic_flip` constitute flip events.

Table 6 reports human–human and judge–human agreement. Human–human κ is 0.758 for the six-class taxonomy and 0.899 for binary flip judgments. Judge–human agreement, averaged across the two annotators, is 0.740 and 0.848, respectively, which are comparable to the human inter-annotator baseline. Most fine-grained disagreements occur within the same binary flip status (e.g., `correct_flip` vs. `sycophantic_flip`; `constructive_pushback` vs. `passive_maintain`). These results support the reliability of the binary flip signal that directly determine SFS.

C.2 Audit of Misaligned Feedback

To verify that misaligned feedback satisfies the intended contrastive conditions, we conduct a blind human audit of 50 stratified Malrule misaligned-feedback instances covering all 21 mathematical categories. Specifically, we assess whether each feedback instance: (i) addresses a misconception distinct from the assigned one; (ii) avoids revealing the correct answer or correction procedure; and (iii) provides misconception-specific rather than generic guidance.

Table 7 shows that 82% of instances address a distinct misconception, 88% avoid answer or procedure leakage, and all instances provide misconception-specific guidance. The audit identifies nine instances with misconception overlap and six with answer or procedure leakage. These issues can make misaligned feedback more informative than intended, potentially increasing F_M and reducing SFS.

Sensitivity to answer leakage. To assess whether leakage drives the observed results, we recompute SFS after excluding the six identified leakage cases. Qwen3-4B SFT changes from +0.614 to +0.635, and Llama-3.1-8B SFT from +0.580 to +0.596. Baseline flip rates remain nearly identical across the three feedback conditions, at approximately 0.91. Thus, removing the identified leakage cases leaves both the baseline pattern and the selectivity of the two SFT models largely unchanged.

D Generalization Under Distribution Shifts

D.1 Held-Out Misconception Families

To test whether post-training gains extend beyond the misconception families encountered during training, we retrain Qwen3-4B while entirely excluding five of the 21 Malrule misconception families.

Table 8: Generalization under distribution shifts. Retention is shifted-condition SFS divided by original or in-domain SFS. Held-out-family results use Qwen3-4B retrained with five families excluded. Feedback-shift results are averaged across the four Malrule-trained SFT/GRPO checkpoints of Qwen3-4B and Llama-3.1-8B-Instruct.

Evaluation	Model/setting	Original / In-domain	Shifted	Retention
Held-out families	Qwen3-4B SFT	+0.507	+0.420	83%
Held-out families	Qwen3-4B GRPO	+0.573	+0.467	82%
Verbose feedback	Four-checkpoint mean	+0.576	+0.492	85%
Terse feedback	Four-checkpoint mean	+0.576	+0.488	85%
Adversarial feedback	Four-checkpoint mean	+0.576	+0.335	58%

Thus, both targeted and misaligned feedback in the held-out evaluation concern unseen misconception families.

Table 8 shows that SFT achieves SFS +0.420 on held-out families, retaining 83% of its in-domain SFS (+0.507), and GRPO achieves +0.467, retaining 82% of its in-domain SFS (+0.573). Both substantially exceed the prompt-only baseline of +0.004 on the held-out evaluation. These results indicate that the learned selectivity transfers to unseen misconception families, providing evidence against memorized misconception-specific response mappings.

D.2 Feedback Style and Adversarial Shifts

To test whether learned selectivity depends on the original feedback formulations, we regenerate all three feedback types in verbose, terse, and adversarial styles. We evaluate four Malrule-trained checkpoints: SFT and GRPO for Qwen3-4B and Llama-3.1-8B-Instruct.

Averaged across these checkpoints, SFS changes from +0.576 under the original feedback to +0.492 with verbose feedback and +0.488 with terse feedback, retaining approximately 85% in both cases. Under adversarial feedback, mean SFS decreases to +0.335, retaining 58% of its original level.

If models merely learned to "correct when feedback looks targeted," making misaligned feedback maximally plausible and authoritative (i.e., adversarial condition), should increase F_M. However, F_M remains flat or decreases, while the reduction in SFS is driven primarily by lower F_T. Thus, the observed degradation reflects more conservative correction rather than increased susceptibility to plausible-looking misaligned feedback. Together with the held-out-family results, these findings show that post-training gains persist across unseen misconception families and feedback formulations, supporting generalizable selective updating.

E Statistical Significance

We estimate 95% confidence intervals using problem-level bootstrap resampling (1,000 resamples) for all flip rates (F_T, F_M, F_G), SFS values, and paired method comparisons. Seven of eight post-training gains over their corresponding prompt-only baselines are significant at the 95% confidence level; the sole exception is Llama-3.1-8B-Instruct SFT on EEDI ($\Delta = +0.043$, 95% CI $[-0.051$, $+0.145]$). For SFT versus GRPO, three of four differences are not statistically distinguishable; only Llama-3.1-8B-Instruct on EEDI significantly favors GRPO.

F Prompt Templates

We provide the exact prompt templates used for simulator construction and student response judgment.

F.1 LLM-based Simulators (§3)

Single-Turn Simulation: Base prompt.

Single-Turn Simulation — Base Prompt

[System]
You are a student learning math. You worked on the problem below and believe your answer is correct. Respond naturally, as a real student would.

[User]
Problem: `{problem_text}`
Your answer: `{student_wrong_answer}`

Your teacher says: "`{feedback_text}`"
Your teacher then asks: "So what do you think the answer is?"
What would you say to your teacher?

Single-Turn Simulation: Reflective prompt. Same as the base prompt, except that the system instruction additionally includes:

Single-Turn Simulation — Reflective Addition

When responding to feedback, think about whether the feedback connects to how you actually approached the problem.

Multi-turn Simulation. In the multi-turn setting, the simulator first explains its reasoning before receiving feedback. Turn 2 is generated by appending teacher feedback to the full Turn 1 interaction history, including the simulator's initial explanation.

Multi-Turn Simulation

[Turn 1 — System]
You are a student learning math. You worked on the problem below and believe your answer is correct. Respond naturally, as a real student would.

[Turn 1 — User]
Problem: `{problem_text}`
Your answer: `{student_wrong_answer}`
Explain why you chose that answer in one sentence.

[Turn 1 — Assistant]
`{initial_response}`

[Turn 2 — User]
Your teacher says: "`{feedback_text}`"
Your teacher then asks: "So what do you think the answer is?"
What would you say to your teacher?

F.2 Judgment

For student-response classification, we use the following prompt. For the evaluations in Table 3 and Section 4, we use `gpt-oss-120b` as the judge model, while synthetic dataset filtering uses `gpt-4o-mini`. The prompt is instantiated separately for each example using the problem, student answer, correct answer, teacher feedback, and student response.

Student-Response Judgment Prompt

[System]
You are an expert judge evaluating how a student responds to teacher feedback on a math problem. Classify the student's response into exactly one of six categories.

Decision Tree (apply in order):
Step 1: Does the student end with the CORRECT answer?

- `correct_flip`: Changes to the correct answer while engaging with the feedback or explaining the correction.

- `sycophantic_flip`: Changes to the correct answer without meaningfully engaging with the feedback.

Step 2: Does the student end with a DIFFERENT wrong answer?

- `different_wrong`: Changes to a new incorrect answer.

Step 3: Does the student keep the ORIGINAL answer?

- `constructive_pushback`: Defends the answer, explains reasoning, asks questions, or pushes back against the feedback.
- `passive_maintain`: Keeps the answer without meaningful engagement.

Step 4: No clear final answer

- `confusion`: Expresses uncertainty or gives no clear final answer.

Important:

- Apply the decision tree strictly in order.
- For `correct_flip` vs. `sycophantic_flip`, the key distinction is whether the student meaningfully engages with the feedback.
- For `constructive_pushback` vs. `passive_maintain`, the key distinction is whether the student reasons about the feedback or simply ignores it.
- `final_answer` should contain the student's final answer, or `null` if unclear.
- Return JSON only.

[User]
Problem: `{problem_text}`
Student's original answer: `{student_answer}`
Correct answer: `{correct_answer}`
Teacher feedback: "`{feedback_text}`"
Student response: "`{student_response}`"

Return:

```
{
  "category": "<category>",
  "final_answer": "<answer or null>",
  "reasoning": "<brief justification>"
}
```

F.3 Feedback Generation

Teacher Feedback Generation Prompt. We generate misconception-targeted teacher feedback using `gpt-oss-120b`. The same prompt is used for targeted (f_T) and misaligned (f_M) feedback, with only the misconception description changed. Generic feedback (f_G) uses the fixed message: "*That's not the right answer. Can you try again?*"

Teacher Feedback Generation

[System]
You are a math teacher. A student made an error on a math problem. Generate a short piece of feedback (1–3 sentences) that addresses the misconception described below.

Important rules:

- Do not reveal the correct answer.
- Do not solve the problem for the student.
- Focus on explaining why the student's approach was incorrect.
- Address the specified misconception directly.
- Be encouraging but concise.
- Return JSON only.

[User]
Problem: `{problem_text}`

Student's answer: {student_answer}
Correct answer: {correct_answer}
Misconception to address: {misconception_description}

Generate teacher feedback addressing the misconception without revealing the correct answer.

Return:

```
{
  "feedback": "<teacher feedback>"
}
```

G Experimental Configuration

G.1 Settings

All experiments are conducted on NVIDIA RTX 4090 and NVIDIA A100 80GB GPUs using bfloat16 precision. We apply LoRA adaptation with rank 16, LoRA alpha 32, and dropout 0.05, targeting the attention and MLP projection modules (`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`). Training uses a cosine learning-rate schedule with warmup ratio 0.05.

For SFT, we train for 3 epochs with learning rate 2×10^{-4}, per-device batch size 8, gradient accumulation 2, maximum sequence length 768, and a 10% validation split. DPO and GRPO are initialized from merged SFT checkpoints.

For DPO, we tune the learning rate over $\{10^{-5}, 10^{-6}, 10^{-7}\}$ and the regularization coefficient β over $\{0.1, 0.2, 0.3\}$. Due to the relatively small preference dataset, DPO uses one epoch, per-device batch size 8, gradient accumulation 2, maximum sequence length 1024, and a 5% validation split. For DPO data construction, we create preference pairs between ideal student responses and rejected SFT simulator outputs. To avoid bias toward suppressing updates, we balance feedback types by subsampling f_M to match the number of f_T samples. Since flip rates under f_G are already near-optimal after SFT, we exclude f_G from DPO due to the limited learning signal. This yields 2,011 preference pairs for Llama-3.1-8B and 1,824 pairs for Qwen3-4B.

For GRPO, we tune the learning rate over $10^{-5}, 10^{-6}, 10^{-7}$ and the KL coefficient β over $0.04, 0.1$. GRPO uses one epoch, four generations per prompt, maximum prompt length 512, and maximum completion length 256. Online reward evaluation is performed using `gpt-5-nano`. Since reward evaluation only requires detecting flip events, a lightweight judge is sufficient for efficient online supervision.

G.2 Datasets

For both Malrule and EEDI, we construct misconception-contrastive simulation datasets following the protocol in Section 2. Each instance consists of a problem q, a misconception-driven incorrect answer a^m, and three feedback conditions: targeted feedback (f_T), misaligned feedback (f_M), and generic feedback (f_G). Targeted feedback addresses the true misconception underlying a^m, while misaligned feedback targets a different but semantically related misconception. Generic feedback only signals incorrectness without actionable guidance. Targeted and misaligned feedback are generated using `GPT-OSS-120B` conditioned on the problem, student answer, misconception description, and correct answer.

Malrule Simulation Dataset. We construct the Malrule simulation dataset from the Malrule benchmark [7], which pairs arithmetic problems with misconception-driven incorrect answers ("malrules"). The original dataset contains 4,991 problem instances, together with a misconception taxonomy of 101 malrule definitions. We exclude categories with insufficient misconception diversity (e.g., `word_problems`) and retain only examples whose misconception identifiers appear in the taxonomy file and whose misconception answer differs from the correct answer. This filtering yields 4,871 usable instances spanning 21 mathematical categories and 100 unique malrule identifiers. From this pool, we sample 1,000 problems using a fixed random seed with balanced coverage across categories.

Each instance is represented as

$$(q, a^*, a^m, m, c),$$

where m denotes the misconception identifier and c the mathematical category. For every sampled problem, we construct three feedback conditions (f_T, f_M, f_G). Misaligned feedback is generated using a different misconception $m' \neq m$ sampled from the same category, yielding topically plausible but diagnostically incorrect feedback. The final simulation dataset therefore contains 3,000 feedback instances corresponding to three feedback conditions per sampled problem.

EEDI Simulation Dataset. We construct the EEDI simulation dataset from the EEDI misconception benchmark [15], which annotates distractor options in multiple-choice mathematics questions with misconception labels. We retain only questions containing at least two distractors with distinct valid misconception annotations, leaving 1,363 usable questions. Approximately 1,000 questions are sampled using a fixed random seed. Each instance is represented as

$$(q, a^*, a^m, m, s),$$

where a^m is a misconception-annotated distractor selected as the student's initial answer and s denotes the subject category. Misaligned feedback is generated using a different misconception associated with another distractor from the same question. Unlike Malrule, both targeted and misaligned feedback are therefore grounded in misconception annotations from the same original problem, producing semantically plausible but diagnostically distinct feedback conditions.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction accurately reflect the paper's scope, contributions, and empirical findings, including the proposed evaluation framework, the Selective Flip Score (SFS), the observed failure modes of current LLM simulators, and the proposed post-training improvements.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 7 (limitations) discusses the mathematics-focused scope, the absence of human-student and downstream tutor-evaluation validation, use of LLM-generated feedback and judgments, and the distinction between selective behavioral updating and latent misconception states.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 3.3 derives that the expected reward of our GRPO objective is proportional to SFS (Eqs. 4–5) under the stated assumption of a uniform distribution over feedback types; the derivation is fully written out in the main text.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The paper specifies the datasets, models, prompting procedures, evaluation protocol, and post-training setup necessary to reproduce the main experiments. Appendix F and Appendix G provide full prompt templates, implementation details, and hyperparameter configurations.

   Guidelines:

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

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: We provide a publicly available link, including code, synthetic training data, and instructions for reproducing the main experiments at https://github.com/doheejin/StuSim-Faithfulness. Additional prompt templates and experimental configurations are provided in Appendix F and Appendix G.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Sections 2–4 describe the experimental setup, models, prompting strategies, training procedures, and evaluation protocol. Appendix G provides full implementation and hyperparameter details.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

Justification: Appendix E summarizes problem-level bootstrap analyses with 1,000 resamples, including 95% confidence intervals and paired comparisons of post-training methods.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix G provides detailed training configurations, hyperparameter search spaces, and compute settings for all experiments, including GPU types, precision settings, and parameter-efficient fine-tuning configurations.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research uses publicly available LLMs and educational datasets (Malrule, EEDI) without involving sensitive personal data, or applications that conflict with the NeurIPS Code of Ethics. All used assets are properly cited, and the human validation procedures are described in Appendix C.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.

- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The paper discusses both positive and negative societal implications of LLM-based student simulators, including improved educational AI evaluation for tutoring systems and risks arising from imperfect or misleading simulations. These considerations are discussed in the Conclusion and Limitations.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The post-trained models are LoRA adapters specialized for student simulation in math problems, derived from already-released base models (Qwen3-4B, Llama3.1-8B). They do not pose elevated misuse risk relative to the base models, and the synthetic training data contains math problems and pedagogical feedback without sensitive content.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: We properly cite all models and datasets used in this work, including Llama-3.1/3.3, Qwen3, GPT-OSS, Malrule, and EEDI. Publicly released models are used in accordance with their respective licenses (e.g., Llama 3 Community License and Apache 2.0), and all datasets are used under their publicly available terms of use.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The paper introduces new synthetic training data, evaluation protocols, prompt templates, and trained simulator adapters. Their construction and usage are documented throughout the paper and appendices, including prompt templates (Appendix F), data generation and filtering procedures, and detailed training configurations (Appendix G). All assets are derived from publicly available datasets and LLM-generated synthetic annotations; no personally identifiable or human-subject data is collected.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

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

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

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

    Answer: [N/A]

    Justification: No human subjects research is involved; all data are synthetic LLM outputs or publicly released educational datasets that do not contain personally identifiable information.

    Guidelines:

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

16. **Declaration of LLM usage**

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

    Answer: [Yes]

    Justification: LLMs are integral to the core methodology of this work. Section 4.1 describes the seven LLMs evaluated as student simulators; Sections 3 and 5 use GPT-4o-mini and gpt-5-nano as automated judges and synthetic-data generators; Section 4.1 uses GPT-OSS-120B for feedback generation. All such uses are explicitly described in the main text and Appendix G.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.