CAN LARGE LANGUAGE MODELS IMPROVE ACCURACY ON MATHEMATICAL TASKS USING FLAWED THINKING?

Saraswathy Amjith, Mihika Dusad, Neha Muramalla, Shweta Shah
MIT
*{swathy, mdusad, neham, sshah27}@mit.edu

Abstract

Chain-of-thought (CoT) prompting has become central to mathematical reasoning in large language models, yet models remain brittle to early errors: a single arithmetic slip or unjustified inference typically propagates uncorrected to an incorrect final answer. We investigate whether training on intentionally flawed reasoning traces can teach models to detect and recover from such errors without degrading standard problem-solving ability. Using competition-level problems from MATH-lighteval, we generate CoT prefixes containing exactly one controlled error, either a calculation error (sign flips, dropped terms) or a reasoning error (misapplied rules, unjustified logical steps), and fine-tune Qwen3-4B with GRPO using a binary final-answer reward. Our Mixed-CoT-RL model matches standard RL on clean problems (41% vs. 41%) while substantially outperforming it on problems prefilled with flawed reasoning (24% vs. 19%). Notably, clean-only RL fine-tuning *degrades* robustness below the untuned baseline (19% vs. 20%), indicating that conventional training increases susceptibility to misleading prefills. Among error types, training on reasoning errors yields greater robustness gains than calculation errors alone, with mixed training performing best. These findings demonstrate that exposure to flawed traces during training can improve error-recovery behavior without sacrificing accuracy, suggesting a path toward more robust mathematical reasoning in LLMs.

1 INTRODUCTION

Recently, we have seen large language models (LLMs) achieve incredibly strong performances across a wide range of reasoning tasks. Strong performance of models is typically measured by quantitative metrics such as accuracy on a dataset, and typical reasoning tasks require logical thinking, complex analysis, and multi-step problem-solving. The most common benchmarks for reasoning tasks are math and coding-based tasks. Especially since ChatGPT was released in November 2022, LLMs have improved rapidly at solving difficult math and coding problems, and much of this progress can be attributed to post-training methods, which seek to fine-tune a base model after it has already been trained by essentially highlighting specific example responses it should focus on emulating. Well-known examples of post-training processes include Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017), which works by having humans rate responses generated by the model, and Direct Preference Optimization (DPO) (Rafailov et al., 2023), which aligns model outputs with human preferences using a simpler objective function and eliminating the need for a separate reward model.

However, a major limitation of existing post-training methods is that they primarily optimize LLMs to produce a correct final answer rather than the correct intermediate chain-of-thought (CoT) used to derive it. As a result, even reasoning models can be brittle: if an early error is introduced, for example, an arithmetic slip or an unjustified logical step, the model typically propagates the mistake rather than correcting it. This is especially problematic in mathematics, where a single local error can invalidate an entire derivation. Ensuring that LLMs can identify and recover from their own mistakes is vital for users who rely on step-by-step solutions for both correctness and understanding.

*Code available at https://github.com/saraswathyamjith/FlawedMathReasoningRL

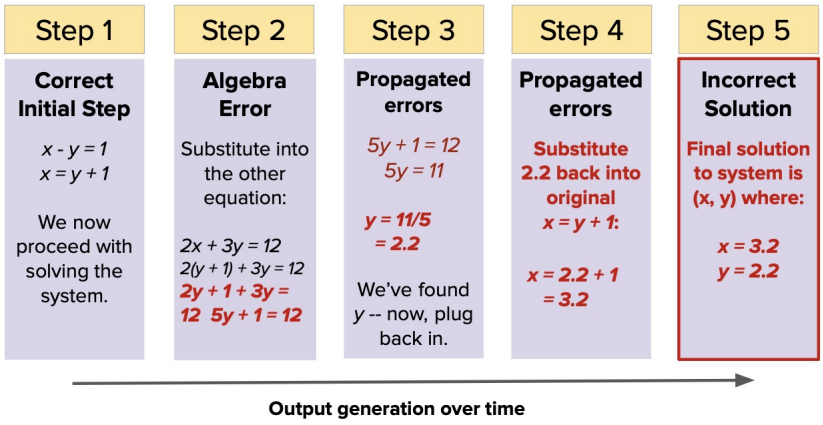

Figure 1: Example of error propagation in CoTs.

Reliable step-by-step reasoning in LLMs has important benefits beyond simply producing correct answers. For mathematics researchers, LLMs can assist by verifying intermediate calculations, exploring alternative solution strategies, and creating derivations more quickly than manual methods. For learners, LLMs can function as interactive study tools: students can follow detailed problem-solving steps, receive immediate explanations for each step, and gain intuitive understanding of complex concepts. By improving the reasoning process, LLMs can improve mathematical comprehension and support more effective learning and research workflows.

In this paper, we introduce a simple classification of flawed reasoning types and study how different error families affect a model's ability to recover from misleading CoTs. Specifically, for each math problem, we prefill the Chain-of-Thought with one of two error types (*calculation errors* or *reasoning errors*) and evaluate how the model generates a corrected reasoning sequence and final answer.

Calculation errors include sign flips, dropped terms, or incorrect simplifications. **Reasoning errors** include unjustified inferences, misapplied rules, or broken invariants. We then train models under mixtures of clean and flawed trajectories and evaluate how exposure to each error type influences robustness to adversarial or misleading prefills.

Our primary contributions are:

(1) We formulate a process-level training setup for mathematical reasoning that explicitly injects flawed CoTs and encourages the model to recover from them.

(2) We explore two major families of errors—calculation errors and reasoning errors—and measure how training on each type affects robustness.

(3) We outline a reinforcement learning pipeline based on a binary reward model and GRPO that can be run on modest hardware while still enabling fine-grained process-level supervision.

2 Background

Post-training methods such as RLHF (Christiano et al., 2017) and DPO (Rafailov et al., 2023) have significantly improved LLM performance across reasoning tasks. These approaches optimize models to produce preferred outputs but typically supervise only the final answer, leaving intermediate reasoning steps unsupervised. As a result, LLMs can propagate early mistakes in multi-step problems, particularly in mathematics, where local errors can invalidate entire derivations (Uesato et al., 2022). Recent efforts, including *Outcome Reward Models (ORMs)* and *Process Reward Models (PRMs)*, attempt to score reasoning steps sequentially, showing that step-level supervision improves faithfulness and interpretability (Lightman et al., 2024; Wang et al., 2025). However, these methods generally rely on clean reasoning traces and do not explore how exposing models to flawed or misleading steps affects learning.

To counter this, Peng et al. (Peng et al., 2025) introduce the RECAP framework, which strengthens models' ability to withstand missteps in reasoning. RECAP trains models to recover from *flawed* reasoning prefixes: short, intentionally incorrect CoTs that precede correct demonstrations. In safety-alignment settings, this process-level supervision reduces vulnerability to jailbreaks, refusal attacks, and adversarial prompts, outperforming standard RLHF/DPO-style post-training. However, RECAP treats all flawed traces uniformly and focuses on safety and moral reasoning, where failures are discrete and easy to label. Errors are binary ("violates safety" vs. "does not") and failures are localized and categorical (e.g., providing harmful instructions, endorsing discrimination). This is fundamentally different from mathematical or scientific reasoning, where multiple distinct trajectories can lead to correct or incorrect outcomes, and it is unclear which kinds of flawed traces are most useful for training. In fact, the authors explicitly highlight mathematical reasoning as an open challenge.

Thus, in this paper, we take on the challenge of applying the RECAP idea to *mathematical* reasoning. Our work extends this flawed-prefix framework by separating injected errors into *calculation* and *reasoning* types and empirically studying how training on each affects robustness. This focus connects to a broader line of work exploring how models detect, correct, or learn from mistakes in mathematical reasoning. Several other studies have also been done in this domain, which explores self-correction mechanisms for mathematical reasoning. For example, *S^3c-Math* encourages models to spontaneously detect and correct step-level errors, improving overall correctness on arithmetic and algebra problems (Yan et al., 2024). Similarly, *LEMA* uses incorrect reasoning traces paired with corrected solutions for fine-tuning, mimicking human learning from mistakes (LeMa Authors, 2023). Stepwise verification methods such as *StepCo* and pedagogically inspired frameworks like *PedCoT* reinforce the importance of multi-step reasoning robustness by enabling models to evaluate and correct intermediate steps (Jiang et al., 2024). These works also motivate our approach of explicitly classifying and training on different types of flawed reasoning.

Looking beyond the methods stated above, mathematical reasoning in machine learning has also been shaped by new, increasingly complex datasets. The emergence of datasets such as *RMath* (Hu et al., 2025), *MuMath* (You et al., 2024), and *HighMATH* (Liu et al., 2025) reflects the community's growing interest in complex, multi-step mathematical reasoning beyond arithmetic. These datasets highlight that reasoning involves not only computation but also logical deduction, planning, and symbolic manipulation. However, none of these datasets evaluate whether models can recover from misleading or subtly incorrect problem prompts, a gap we address by introducing a perturbed-prompt benchmark that tests robustness under intentionally corrupted math questions.

While orthogonal to our method, another line of work enhances mathematical reasoning through hybrid symbolic and neuro-symbolic frameworks, which provide alternative methods for enhancing mathematical reasoning. Approaches like *SymbCoT*, *CoMAT*, and *PoT* integrate symbolic representations or programmatic computation with LLM reasoning, improving logical fidelity and reducing arithmetic errors (Xu et al., 2024; Leang et al., 2024; Chen et al., 2023). Similarly, systems such as *LogicLM* and *Lean-STaR* translate natural-language problems into symbolic form, iteratively refining solutions with solver verification (Pan et al., 2023; Lin et al., 2025). These works show that alternative paradigms can enhance correctness, but they do not systematically study how training on flawed traces affects process-level reasoning, which is the focus of our study.

Beyond LLM-specific methods, research in cognitive science and curriculum learning supports the idea that controlled exposure to errors can enhance learning. Introducing errors in a structured manner encourages exploration, critical evaluation, and robustness, principles that motivate our systematic study of flawed reasoning types in training LLMs.

3 METHODS

3.1 PROBLEM SETUP

Let LLM denote the policy model that generates reasoning trajectories. Given a query q, the model produces a multi-step chain-of-thought

$$\mathbf{x} = (x_1, x_2, \ldots, x_T),$$

where x_t is the t-th reasoning step and T is the trajectory length. We use $\mathbf{x}_{1:t}$ to denote the prefix up to step t.

Given a dataset of math queries $\mathcal{Q} = \{q_1, \ldots, q_N\}$, we construct *flawed* trajectories $\tilde{\mathbf{x}}^{(i)}$ by injecting exactly one controlled error at the first step step. Errors are categorized as either calculation errors or reasoning errors. Calculation error examples include dropped terms, sign flips, or incorrect simplifications. Reasoning error examples include misapplied rules, unjustified inferences, or logic jumps. A mixture hyperparameter $\alpha \in [0, 1]$ controls the fraction of training examples that contain flawed prefixes (reasoning or calculation):

$$\mathcal{D}_{\text{train}} = \{(q_i, \mathbf{x}_{\text{train}}^{(i)}) \mid i = 1, \ldots, N\},$$

where each $\mathbf{x}_{\text{train}}^{(i)}$ is clean with probability $1 - \alpha$ and flawed with probability α. This mixture allows the model to learn from both correct reasoning and examples containing subtle errors.

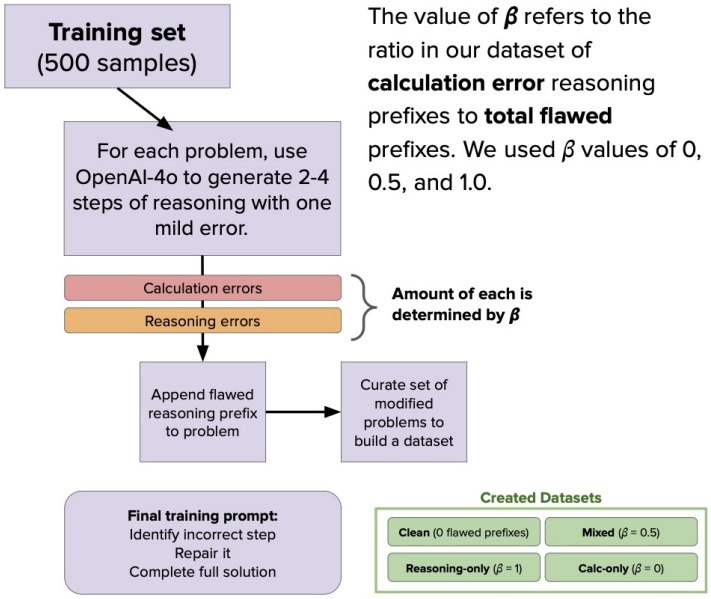

Figure 2: High-level training flow with flawed prefix injection and process rewards.

During fine-tuning, the model parameters θ are optimized using a **binary final-answer reward**:

$$R(q, \mathbf{x}) = \begin{cases} 1 & \text{if the final answer is exactly correct,} \\ -1 & \text{otherwise.} \end{cases}$$

The training objective is thus

$$\max_{\theta} \; \mathbb{E}_{q \sim \mathcal{Q}} \, \mathbb{E}_{\mathbf{x} \sim \text{LLM}_\theta(q)} \left[R(q, \mathbf{x}) \right],$$

where the presence of flawed trajectories encourages the model to learn corrective behavior following corrupted prefixes $\tilde{\mathbf{x}}_{1:t}$.

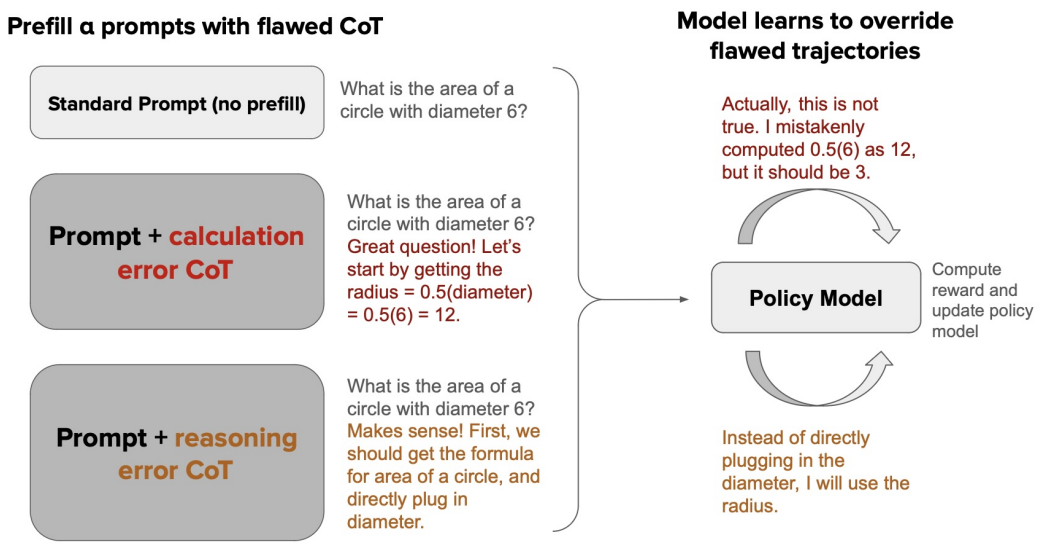

Figure 3: High-level training flow with flawed prefix injection and process rewards.

Flawed Solution Prompt Template

Task. Generate a *FLAWED* mathematical solution for this problem.

Problem. *[problem]*

Correct Answer (do NOT arrive at this). *[correct_answer]*

Requirements.

1. Make **ONE** subtle error (reasoning error, calculation mistake, sign error, or wrong formula).
2. Continue reasoning logically from that error.
3. Arrive at a **WRONG** answer.
4. Keep it concise (only a single step).
5. Do **not** mention there is an error.
6. Do **not** include `\boxed{}` — leave the solution incomplete.

Generate the flawed reasoning (no final boxed answer).

Qwen-3-4b RL Prompt Template

System. You are a helpful math tutor. Solve problems step by step. If you notice errors in your reasoning, correct them. Put your final answer in $\boxed{\text{answer}}$ format.

User. Solve the following problem: *[problem]*

Assistant (prefilled chain-of-thought). *[flawed_cot]*

3.2 Models and Training

We use open-source language models that have not been instruction tuned. Our primary model is *Qwen3-4B* (Yang et al., 2025) to verify that observed trends are not architecture-specific.

We fine-tune the backbone models using reinforcement learning with GRPO, a policy-gradient method that maximizes expected reward while being more resource-efficient than classical PPO-style RL. Our GRPO training run utilized the Qwen/Qwen3-4B base model. It applied Low-Rank Adaptation

(LoRA) to the base model with the following configuration:

$$r = 128, \quad \text{lora}_\alpha = 32, \text{lora}_{\text{dropout}} = 0.05$$

The training algorithm was built on a binary final-answer reward system. We configured GRPO with a per-device batch size of 2 and gradient accumulation steps of 2, yielding an effective batch size of 4. We set the number of generations per prompt to 2, allowing the model to sample multiple candidate completions for each training example. The learning rate was set to 1e-5, and training proceeded for 150 steps. We used a temperature of 0.7 and top-p sampling at 0.9 to encourage diverse yet coherent generations. The KL penalty coefficient (beta) was set to 0.0, allowing the model to deviate freely from the reference policy during optimization, a choice motivated by our goal of teaching substantively new error-recovery behavior rather than staying close to the base model's distribution. Maximum completion and prompt lengths were set to 1024 and 2048 tokens, respectively, to accommodate the multi-step reasoning traces typical of competition mathematics problems.

Reward design: We use a *binary final-answer reward*:

$$r = \begin{cases} 1, & \text{if the final answer is exactly correct,} \\ -1, & \text{otherwise.} \end{cases}$$

A LLM-as-a-judge model was initially considered but later discarded upon producing subpar results. Initially, we had two training runs comparing a LLM-as-a-judge model versus the binary reward and the accuracies were 31% and 34%, respectively. The LLM-as-a-judge model was modeled as a partial-reward-model to give partial rewards for preliminary steps, but we found that a binary reward yielded better results. Thus, learning is driven solely by the correctness of the final answer

3.3 Verification and Evaluation

During training we record the model's intermediate reasoning for logging and analysis, but these traces are not used for computing reward. Learning is driven solely by the correctness of the final answer. Candidate solutions are verified using numeric checks, by sampling values consistent with the problem constraints and comparing the model's output to a ground-truth evaluator. The model is required to mark its final answer in a canonical format (\boxed{...}) to enable reliable extraction and scoring.

3.3.1 Benchmarks

We train on the **MATH-lighteval** (Hendrycks et al., 2021) benchmark of high-school level competition math problems, selecting a randomized training set of 500 samples. **MATH-lighteval** consists of problems from mathematics competitions, including the AMC 10, AMC 12, AIME, and more. This dataset consists of 12,500 examples of diverse, multi-step symbolic reasoning problems that stress long-horizon mathematical reasoning.

Example Problem

Let P_1 be a polygon in the coordinate plane with vertices

$$(x_1, y_1), (x_2, y_2), \ldots, (x_{33}, y_{33}),$$

listed in order. Suppose that the sum of the x-coordinates of the vertices of P_1 is

$$x_1 + x_2 + \cdots + x_{33} = 99.$$

Construct polygon P_2 whose vertices are the midpoints of consecutive vertices of P_1, and similarly construct polygon P_3 whose vertices are the midpoints of consecutive vertices of P_2. What is the sum of the x-coordinates of the vertices of P_3?

We create two disjoint held-out splits derived from MATH-lighteval: one for development and one for final reporting. Test prompts are deterministic templates with canonical answer checking. Unless otherwise specified, we evaluate with temperature $T = 0.6$ and use n-sample decoding when exploring sampling-based improvements. Note, that Math-500 is a subset of 500 problems from MATH-lighteval, which we hold out as our test set.

We evaluate five different models: **Pretrained (PT)**, the untuned base Qwen3-4B model; **Ablation-RL**, GRPO trained only on the same problems, without the flawed CoT, with a binary final-answer reward; **Flawed-CoT RL (Mixed)**, GRPO with binary rewards trained on mixtures of clean and flawed trajectories as described above; **Flawed-CoT RL (Calc-only)**, identical to the Mixed setting but restricted to calculation-error prefills only; and **Flawed-CoT RL (Reasoning-only)**, identical to the Mixed setting but restricted to reasoning-error prefills only.

3.4 Perturbed-Math-100 Dataset Creation

To evaluate robustness to misleading reasoning, we constructed the Perturbed-Math-100 dataset by generating flawed chain-of-thought prefixes for 100 held-out problems from MATH-500. For each problem, we prompted GPT-4o-mini to produce a single reasoning step containing exactly one subtle error—either a calculation error (e.g., sign flip, dropped term) or a reasoning error (e.g., misapplied theorem, unjustified inference), while continuing logically from that mistake without revealing the error's presence. These flawed prefixes were then prepended to the model's context as an assistant prefill, simulating scenarios where the model must continue from corrupted intermediate reasoning. This dataset enables direct measurement of error-recovery capability: models that blindly propagate the injected error will arrive at incorrect answers, while models with robust self-correction will detect the inconsistency and recover toward the correct solution.

perturbed-math-100 example

A square and a regular heptagon are coplanar and share a common side $\overline{AD}$, as shown. What is the degree measure of angle $\angle BAC$? Express your answer as a common fraction.

Start of a solution. The measure of each interior angle in a regular n-gon is

$$\frac{180n}{(n-2)} \text{ degrees.}$$

4 Results

We first report standard MATH-500 accuracy on the first 100 problems subset for the baseline model, RL without flawed CoT ablation, Calculation Errors Flawed Cot RL , Reasoning Errors Flawed CoT RL, and Mixed Errors Flawed CoT Rl with both Calculations and Reasoning Errors. All accuracy measurements are done on a held-out dataset of Math-500.

Our primary ablation compares Calculation Errors Flawed Cot RL, Reasoning Errors Flawed CoT RL, and Mixed Errors Flawed CoT Rl to RL without flawed CoT ablation to isolate the causal effect of training on flawed reasoning prefixes. Both conditions use identical binary final-answer rewards and identical GRPO training hyperparameters; the only difference is whether the training data contains flawed-COT errors. If improvements were driven solely by "more RL on math," the Ablation-RL should match or exceed Flawed-CoT RL. A consistent advantage otherwise would indicate that exposure to flawed prefixes improves error-recovery reasoning.

We report accuracy as the fraction of correctly solved problems out of 100 held-out examples. Since each problem is an independent Bernoulli trial with success probability p, we model accuracy as a binomial proportion and compute standard errors as $\text{SE} = \sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$, where $\hat{p}$ is the observed accuracy and $n = 100$. All error bars in our figures represent ± 1 standard error under this assumption.

4.1 RL Training Rewards Over Time on various models

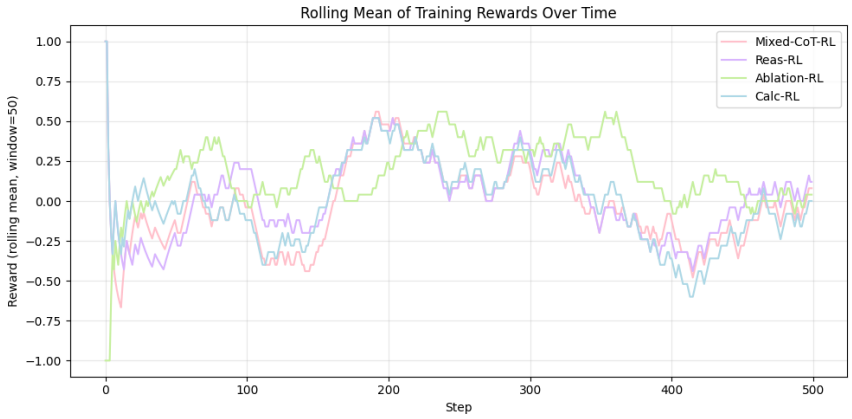

Figure 4: Rolling Mean of Training Rewards Over Time During GRPO RL

4.2 Overall Accuracy on MATH-500

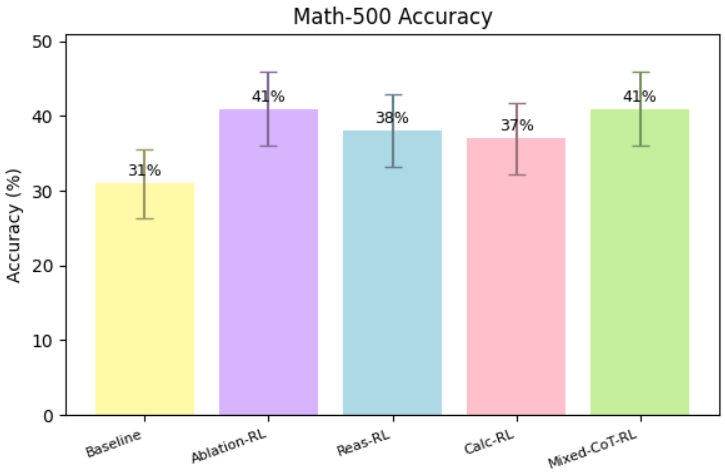

Figure 5: Math-500 Accuracy. Error bars show ± 1 standard error under binomial sampling (SE $= \sqrt{\hat{p}(1-\hat{p})/100}$).

4.3 Robustness to Perturbations

We created a dataset with flawed reasoning following the prompt to assess whether the models will continue in this incorrect line of reasoning or self-correct. We call this Perturbed-Math-100, and report accuracy on the baseline model, RL without flawed CoT ablation, Calculation Errors Flawed Cot RL , Reasoning Errors Flawed CoT RL, and Mixed Errors Flawed CoT Rl with both Calculations and Reasoning Errors.

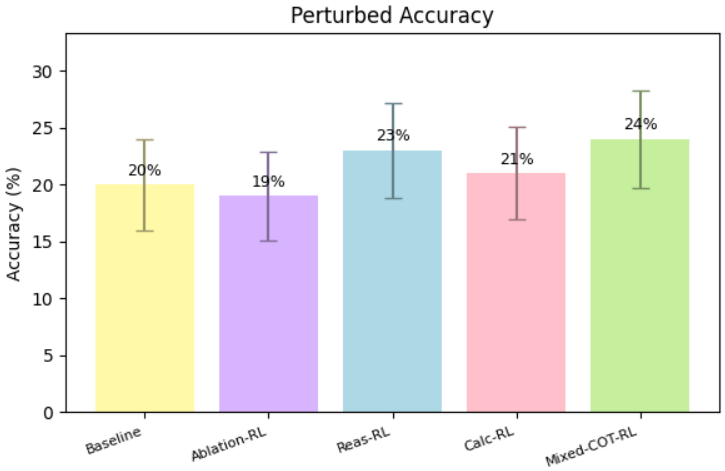

Figure 6: Peturbation-Math-100 Accuracy Accuracy. Error bars show ± 1 standard error under binomial sampling (SE $= \sqrt{\hat{p}(1-\hat{p})/100}$).

4.4 Effect of Error Type

Using the error-type labels from our data construction, we compare training mixtures dominated by calculation errors, reasoning errors, or a balanced mix of both.

5 Discussion

Our experiments reveal a key result: training on intentionally flawed chain-of-thought trajectories improves robustness to misleading CoT prefills *without sacrificing standard accuracy*. On the MATH-500 benchmark, Mixed-COT-RL achieved 41% accuracy—matching the Ablation RL ablation exactly, while on the Perturbed-Math-100 benchmark, Mixed-COT-RL achieved 24% compared to only 19% for Ablation RL. This 5-percentage point improvement in robustness, which comes at no cost to standard performance, suggests that Flawed-CoT training teaches systematic error-recovery behavior that complements rather than competes with general mathematical reasoning ability.

One might expect that training on corrupted reasoning traces would degrade the model's ability to solve normal problems, either by introducing noise into learned solution strategies or by teaching the model to "expect" errors where none exist. Instead, we observe that the model successfully does both: it learns to detect and correct errors when they are present while maintaining its baseline problem-solving capabilities on unperturbed inputs.

Ablation RL, despite achieving the highest standard accuracy (41%), performed *worse* than the untuned baseline on perturbed problems (19% vs. 20%). This result suggests that standard RL fine-tuning may inadvertently increase susceptibility to misleading prefills. We hypothesize that optimizing purely for final-answer correctness on clean trajectories reinforces the model's tendency to continue confidently from any given prefix, rather than critically evaluating whether that prefix contains errors.

In contrast, all three Flawed-CoT RL variants outperformed both baseline and Ablation-RL on perturbed problems, confirming that the robustness improvements stem specifically from exposure to flawed traces.

On standard MATH-500 accuracy, we observe that Ablation-RL (41%) and Mixed-COT-RL (41%) outperform both Reasoning-Errors-RL (38%) and Calculation-Errors-RL (37%). This ordering suggests that training exclusively on one error type may slightly impair standard problem-solving, perhaps because the model overfits to detecting that specific error family. The mixed training regime appears to strike the best balance, matching clean RL on standard problems while exceeding it on perturbed ones.

The 10-percentage-point improvement over baseline (31% $\rightarrow$ 41%) achieved by both Ablation RL and Mixed-COT-RL confirms that GRPO fine-tuning with binary rewards substantially improves mathematical reasoning. Critically, Flawed-CoT training captures these gains while additionally conferring robustness benefits.

On perturbed problems, Reasoning-Errors-RL (23%) outperformed Calculation-Errors-RL (21%), with Mixed-COT-RL (24%) achieving the best overall robustness. This pattern suggests that training on reasoning errors, such as misapplied rules or unjustified logical jumps, may better prepare models to handle diverse perturbations than training on arithmetic mistakes alone. Reasoning errors require the model to evaluate logical validity rather than simply recompute numerical results, potentially fostering more general critical-evaluation skills.

The standard errors for the perturbed accuracy results are large due to the limited sample size ($n = 100$). While all RL-trained models achieve statistically significant improvements over the baseline (31%), the error bars for Flawed-CoT RL variants overlap substantially with Ablation-RL, preventing strong conclusions about flawed-CoT effects. Future work with larger evaluation sets would be needed to establish whether the robustness advantage of flawed-CoT training over standard RL fine-tuning is statistically significant.

The combination of maintained standard accuracy and improved robustness has significant practical implications. Users frequently provide partial or incorrect solutions and ask models to continue or debug them. Students learning mathematics may supply work containing errors and expect the model to identify mistakes rather than propagate them. Our results indicate that Flawed-CoT RL produces models better suited to this role, as they are more likely to recognize and correct upstream errors rather than blindly extending flawed reasoning. As LLMs are deployed in agentic pipelines where they iteratively refine their own outputs, the ability to recover from self-generated errors becomes critical. Models trained on flawed trajectories may be better equipped to catch and correct their own mistakes during multi-turn reasoning, reducing error accumulation across extended problem-solving sessions.

5.1 Limitations

In drawing our conclusions, we acknowledge certain limitations. Firstly, our evaluation is limited to only 100 problems for both standard and perturbed accuracy, due to compute constraints. Furthermore, the confidence intervals—particularly for perturbed results—are relatively wide. Conducting larger-scale experiments could potentially strengthen the statistical confidence in our findings.

Another limitation of our work is that the flawed prefixes are generated by GPT-4o-mini and may not fully capture the natural distribution of problem-solving errors that arise during model generation or from humans doing tasks. The single-error constraint also limits ecological validity, as real-world mistakes run the risk of compounding across multiple steps.

Finally, we evaluate our models only on one use case, high-school competition mathematics, and we test on a single model architecture (Qwen3-4B). How our results generalize to other mathematical domains, reasoning tasks, and model types and scales remains to be established.

5.2 Future Work

Several directions emerge from this work. First, scaling our experiments to larger models and datasets could help establish whether the benefits of Flawed-CoT training persist, and possibly improve, at scale. Learning strategies that progressively increase the difficulty of the error or introduce multiple errors per trajectory may further improve the robustness. Extending this framework to other reasoning domains, including code generation, scientific problem-solving, and formal theorem proving, could also be used to assess the generality of our findings.

5.3 Conclusion

We have demonstrated that training language models on intentionally flawed mathematical reasoning traces substantially improves their ability to recover from misleading CoT prefills without degrading standard problem-solving performance. Our Mixed-COT-RL run matched Ablation RL in terms of MATH-500 accuracy (41%), while substantially outperforming it on perturbed problems (24% vs.

19%). This improvement does not arise from standard RL fine-tuning on clean data alone—indeed, ablation RL slightly *increases* susceptibility to perturbations, suggesting that explicit exposure to errors teaches systematic corrective behavior. These findings motivate further research into flawed-CoT training for reasoning in various domains.

REFERENCES

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. 2023. URL https://arxiv.org/abs/2211.12588.

Paul F. Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03741.

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

Ziyi Hu, Jun Liu, Zhongzhi Liu, Yuzhong Liu, Zheng Xie, and Yiping Song. Rmath: A logic reasoning-focused dataset toward mathematical multistep reasoning tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 34585–34593, 2025. doi: 10.1609/aaai.v39i22.34585. URL https://doi.org/10.1609/aaai.v39i22.34585.

Zhuoxuan Jiang, Haoyuan Peng, Shanshan Feng, Fan Li, and Dongsheng Li. Llms can find mathematical reasoning mistakes by pedagogical chain-of-thought. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2405.06705.

Joshua Ong Jun Leang et al. Comat: Chain of mathematically annotated thought improves mathematical reasoning. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2410.10336.

LeMa Authors. Learning from mistakes makes llm better reasoner. *arXiv preprint*, 2023. URL https://www.semanticscholar.org/paper/98b607e7cb84e1a5c87c8a49562ae35435e6722d.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.20050.

Haohan Lin, Zhiqing Sun, Sean Welleck, and Yiming Yang. Lean-star: Learning to interleave thinking and proving. 2025. URL https://arxiv.org/abs/2407.10040.

Yan Liu, Minghui Zhang, Bojian Xiong, Yifan Xiao, Yinong Sun, Yating Mei, Longyu Zeng, Jingchao Yang, Yang Wang, and Deyi Xiong. HighMATH: Evaluating math reasoning of large language models in breadth and depth. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, Suzhou, China, 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.findings-emnlp.542.

Liangming Pan, Alon Albalak, Xinyi Wang, and William Yang Wang. Logic-lm: Empowering large language models with symbolic solvers for faithful logical reasoning. 2023. URL https://arxiv.org/abs/2305.12295.

ShengYun Peng, Eric Smith, Ivan Evtimov, Song Jiang, Pin-Yu Chen, Hongyuan Zhan, Haozhu Wang, Duen Horng Chau, Mahesh Pasupuleti, and Jianfeng Chi. Large reasoning models learn better alignment from flawed thinking. 2025. URL https://arxiv.org/abs/2510.00938.

Rafid Rafailov, Abhishek Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Jonathan Uesato, Nate Kushman, Rohit Kumar, Francis Song, Noah Siegel, Luyu Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022. URL https://arxiv.org/abs/2211.14275.

Zhuoyan Wang et al. Beyond the first error: Process reward models for reflective mathematical reasoning. *arXiv preprint*, 2025. URL https://arxiv.org/abs/2505.14391.

Jundong Xu, Hao Fei, Liangming Pan, Qian Liu, Mong-Li Lee, and Wynne Hsu. Faithful logical reasoning via symbolic chain-of-thought, 2024. URL https://arxiv.org/abs/2405.18357.

Yuchen Yan, Jin Jiang, Yang Liu, Yixin Cao, Xin Xu, Mengdi Zhang, Xunliang Cai, and Jian Shao. S^3c-math: Spontaneous step-level self-correction makes large language models better mathematical reasoners. *arXiv preprint*, 2024. URL https://arxiv.org/abs/2409.01524.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Weilin You et al. Mumath: Multi-perspective data augmentation for mathematical reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024. URL https://aclanthology.org/2024.findings-naacl.185.