Critique-GRPO: Advancing LLM Reasoning with Natural Language and Numerical Feedback

Anonymous authors
Paper under double-blind review

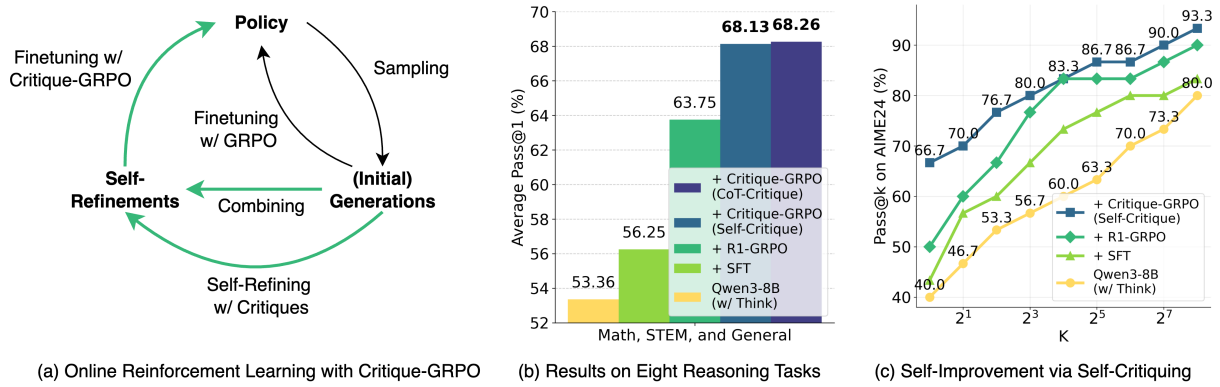
Abstract

Recent advances in reinforcement learning (RL) with numerical feedback, such as scalar rewards, have significantly enhanced the complex reasoning capabilities of large language models (LLMs). Despite this success, we identify three key challenges encountered by RL with solely numerical feedback: performance plateaus, limited effectiveness of self-reflection, and persistent failures. We then demonstrate that RL-finetuned models, even after exhibiting performance plateaus, can generate correct refinements on persistently failed problems by leveraging natural language feedback in the form of critiques. Building on this insight, we propose Critique-GRPO, an online RL framework that integrates both natural language and numerical feedback for effective policy optimization. Critique-GRPO enables LLMs to learn from initial responses and critique-guided self-refinements simultaneously while maintaining exploration. Additionally, we employ a shaping function to amplify learning from correct, especially unfamiliar, refinements and penalize incorrect ones. Extensive experiments with Qwen2.5-7B-Base, Qwen2.5-Math-7B-Base, and Qwen3-8B demonstrate that Critique-GRPO consistently outperforms supervised learning and RL-based fine-tuning methods across eight challenging mathematical, STEM, and general reasoning tasks, improving average pass@1 scores by approximately 4.4% and 3.8% on Qwen2.5-7B-Base and Qwen3-8B, respectively. Notably, Critique-GRPO enables effective self-improvement through self-critiquing and weak-to-strong generalization, achieving consistent gains over GRPO, such as 16.7% and 10.0% pass@1 improvements on AIME 2024, respectively.

1 Introduction

Reinforcement learning (RL) has been a key driver of recent advancements in enhancing the reasoning capabilities of large language models (LLMs) (Yang et al., 2025; DeepSeek-AI et al., 2025; OpenAI et al., 2024; OpenAI, 2025). In particular, reinforcement learning with numerical feedback, typically in the form of scalar rewards and often referred to as the R1-Zero training paradigm (DeepSeek-AI et al., 2025), enables base LLMs to learn from their own generations through trial-and-error learning. High-quality generations are rewarded positively, while low-quality generations are penalized. This paradigm has revolutionized the post-training pipeline for LLMs, shifting from imitation learning of expert demonstrations to learning from the model's own generations (*i.e.,* experiences) (Zhang et al., 2022; Silver & Sutton, 2025), resulting in significant performance improvements.

Despite recent advancements, RL with solely numerical feedback faces significant challenges. Our analysis of Qwen2.5-7B-Base (Qwen et al., 2025) and Qwen3-8B (Yang et al., 2025) highlights three key issues: (*i*) *Performance Plateaus:* Peak performance does not improve even when scaling training data by 8x (from 4k to 32k examples). (*ii*) *Limited Effectiveness of Self-Reflection:* Spontaneous self-reflection during fine-tuning, often described as "Aha moments," has limited impact on enhancing problem-solving success. (*iii*) *Persistent Failures:* Models consistently fail on certain problems despite extensive trial-and-error fine-tuning. We hypothesize that these limitations stem from the inherent constraints of numerical feedback, which provides limited information about *why* a response is correct or incorrect and *how* to improve it. Additionally, the limited effectiveness of self-reflection behaviors exacerbates these challenges. Together, these issues underscore the need for richer feedback mechanisms to enable more effective learning.



(a) Online Reinforcement Learning with Critique-GRPO (b) Results on Eight Reasoning Tasks (c) Self-Improvement via Self-Critiquing

Figure 1: (a) Critique-GRPO enhances online reinforcement learning by enabling the model to learn from both initial responses and self-refinements through natural language feedback (critiques), highlighted in green, instead of relying solely on numerical feedback (scalar rewards). (b) Critique-GRPO improves the average Pass@1 score on Qwen3-8B by approximately 4.5% across eight reasoning tasks compared to GRPO. (c) Critique-GRPO facilitates self-improvement via self-critiquing, obtaining 66.7% pass@1 on AIME 2024 and consistent gains across pass@k scores (k=1-256) over GRPO.

Natural language feedback (NLF) in the form of textual critiques offers a promising solution by providing detailed, targeted guidance (Saunders et al., 2022; Chen et al., 2024; McAleese et al., 2024). However, existing approaches often fail to fully exploit the potential of textual critiques. Many studies (Kim et al., 2023; Whitehouse et al., 2025; Liu et al., 2025b; Lightman et al., 2024; Zhang et al., 2024) primarily use critiques for evaluation, transforming them into numerical rewards for model improvement via RL algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) or Group Relative Policy Optimization (GRPO) (Shao et al., 2024). This transformation often discards valuable constructive information embedded within the critiques. Some studies (Chen et al., 2024; Wang et al., 2025) utilize critiques to generate refinements and fine-tune models on these refinements through supervised learning. While effective, these offline approaches are limited by their inability to support consistent exploration and online refinement. This raises a key research question: *Can we incorporate critiques into an online reinforcement learning framework to enable LLMs to spontaneously learn from both initial generations and refinements?*

To answer this, we first investigate whether RL-finetuned models, even after their performance has plateaued, can successfully refine responses to persistently failed problems when guided by critiques. Our findings in Section 3 confirm this holds true even for simple indicative critiques ("correct"/"incorrect") and is particularly effective with chain-of-thought (CoT) critiques, which provide a step-by-step evaluation (Wang et al., 2025; Whitehouse et al., 2025). Building on this insight, we propose Critique-GRPO, a novel online RL framework that synergizes numerical and natural language feedback for effective policy optimization. As depicted in Figure 1, Critique-GRPO allows the model to learn from both its initial sampled responses and subsequent self-refinements, which are guided by critiques from a reward system (model-based or rule-based). This dual learning mechanism encourages the model to integrate targeted feedback while preserving policy exploration. Furthermore, we employ a shaping function to amplify learning from correct, unfamiliar refinements while penalizing incorrect ones (Yan et al., 2025).

We evaluate Critique-GRPO on non-reasoning models Qwen2.5-7B-Base (Qwen et al., 2025), Qwen2.5-Math-7B-Base (Yang et al., 2024), and the reasoning model Qwen3-8B (Yang et al., 2025) across five challenging in-distribution mathematical reasoning tasks. Additionally, we assess its generalization capability on three out-of-distribution scientific and general reasoning tasks. Extensive results demonstrate that Critique-GRPO significantly outperforms both supervised and RL-based fine-tuning methods, improving the state-of-the-art (SOTA) average pass@1 by approximately 4.4% on Qwen2.5-7B-Base and 3.8% on Qwen3-8B. Furthermore, exploration into leveraging Critique-GRPO for self-improvement via self-critiquing and weak-to-strong generalization exhibits consistently superior performance over self-improvement with GRPO, *e.g.,* yielding 66.7% *vs.* 50.00% and 60.00% *vs.* 50.00% on AIME 2024, respectively.

In summary, our contributions are three-fold:

- We conduct an in-depth analysis to identify three key challenges of RL using solely numerical feedback and highlight the potential of natural language feedback to address these limitations.
- We propose Critique-GRPO, a framework that enables LLMs to learn simultaneously from both initial responses and their refinements during online RL by leveraging both natural language and numerical feedback.
- We validate the efficacy of Critique-GRPO through extensive experiments, demonstrating superior performance across eight mathematical, STEM, and general reasoning tasks.

2 Related Work

Enhancing LLM Reasoning with Reinforcement Learning. Reinforcement Learning (RL) has proven highly effective in enhancing the reasoning abilities of LLMs (OpenAI et al., 2024; DeepSeek-AI et al., 2025; Fatemi et al., 2025; Li et al., 2025). This is typically achieved by fine-tuning models on complex reasoning tasks to incentivize diverse reasoning behaviors (Gandhi et al., 2025; Yue et al., 2025). Recent advancements have utilized RL with numerical feedback (*e.g.,* +1 for correct responses, -1 for incorrect ones) (OpenAI et al., 2024; DeepSeek-AI et al., 2025; Liu et al., 2025a; Yu et al., 2025). These methods often leverage online policy optimization algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), Group Relative Policy Optimization (GRPO) (Shao et al., 2024), REINFORCE (Williams, 1992), and Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025). However, numerical feedback is inherently sparse, and models frequently struggle with tasks that extend beyond their current knowledge boundaries, limiting their ability to achieve meaningful improvement (Xi et al., 2024; Gandhi et al., 2025). While recent approaches address this limitation by incorporating high-quality expert demonstrations alongside online exploration (Yan et al., 2025), our approach distinctly enables models to refine their outputs by incorporating textual feedback (*e.g.,* CoT critiques) that directly addresses potential errors. This integration of textual feedback with online exploration for policy optimization results in superior performance.

Learning from Natural Language Feedback. Natural Language Feedback (NLF), provided in the form of textual critiques, offers a powerful mechanism for improving LLMs. NLF provides detailed and targeted insights into flaws in model-generated outputs, enabling both accurate evaluation and/or response refinement (Saunders et al., 2022; Chen et al., 2024). Many existing methods convert NLF into numerical reward signals for reinforcement learning (Kim et al., 2023; Whitehouse et al., 2025; Liu et al., 2025b; Lightman et al., 2024; Ouyang et al., 2022; Casper et al., 2023; Rafailov et al., 2024). Other approaches explore learning directly from NLF, often by fine-tuning models to imitate the provided feedback (Hancock et al., 2019; Wang et al., 2025). Additional strategies involve either employing a dedicated refinement model (Chen et al., 2024) or using the primary model itself (Wang et al., 2025) to incorporate feedback into erroneous responses. These corrected responses are then used for further optimization, typically through supervised fine-tuning. In contrast, our approach enables LLMs to directly learn from NLF to iteratively refine their responses, while simultaneously maintaining online exploration through reinforcement learning. This integration of textual feedback and RL further enhances the model's ability to address errors dynamically and improve performance.

3 Limitations of RL with Numerical Feedback and the Promise of Natural Language Guidance

3.1 Limitations of Learning with Numerical Feedback

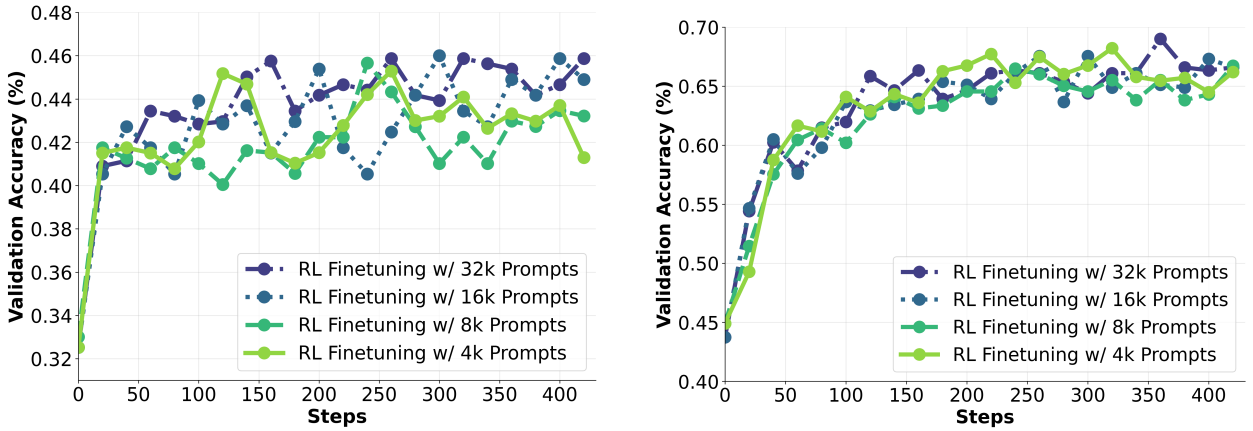
We investigate the limitations of fine-tuning with RL using numerical feedback from three key perspectives: (*i*) How performance evolves over time. (*ii*) The cognitive behaviors that contribute most significantly to successful problem-solving. (*iii*) The model's ability to solve previously failed problems through trial-and-error.

Setup. We conduct experiments on non-reasoning models, Qwen2.5-7B-Base (Qwen et al., 2025) and Qwen3-8B-Base (Yang et al., 2025), and a reasoning model, Qwen3-8B (Yang et al., 2025), for mathematical reasoning tasks. Specifically, we fine-tune the models using GRPO (Shao et al., 2024) with numerical feedback.[1]

Datasets and Evaluation Metrics. We utilize randomly sampled subsets of 4k, 8k, 16k, and 32k examples from a reorganized 45k subset (Yan et al., 2025) of OpenR1-Math-220k (Bakouch et al., 2025). The prompts are sourced from NuminaMath 1.5 (Li et al., 2024), while the ground truth chain-of-thought (CoT) reasoning paths are generated by Deepseek-R1 (DeepSeek-AI et al., 2025). Unless otherwise specified, experiments primarily use 4k training prompts. For validation, we randomly sample 500 examples from the validation set curated by (Yan et al., 2025), which includes examples from Olympiad Bench (He et al., 2024b), MATH (Hendrycks et al., 2021), Minerva-Math (Lewkowycz et al., 2022), AIME 2024 (Li et al., 2024), and AMC 2023 (Li et al., 2024). To enable a holistic evaluation, we assess performance on in-distribution (ID) tasks using Minerva-Math (Lewkowycz et al., 2022) and on out-of-distribution (OOD) tasks using GPQA-Diamond (physics, chemistry, biology) (Rein et al., 2024). During evaluation, we use greedy decoding (temperature = 0) and report accuracy (pass@1)[2].

Reward Design. We employ rule-based evaluation to provide numerical feedback (scalar rewards), using Math-Verify[3] to validate the correctness of generated answers against ground truth during fine-tuning. Binary rewards are assigned as follows: +1 for correct final answers and 0 for incorrect ones. These rewards serve as a proxy for assessing the accuracy of generated responses.

Implementation Details. Our implementation leverages the VERL library (Sheng et al., 2024) and samples four candidate responses per prompt during fine-tuning.



(a) RL fine-tuning dynamics of Qwen2.5-7B-Base.

(b) RL fine-tuning dynamics of Qwen3-8B.

Figure 2: RL fine-tuning dynamics of Qwen2.5-7B-Base (left) and Qwen3-8B (w/ Thinking) (right) using GRPO with numerical feedback on the validation set with varying numbers of training examples.

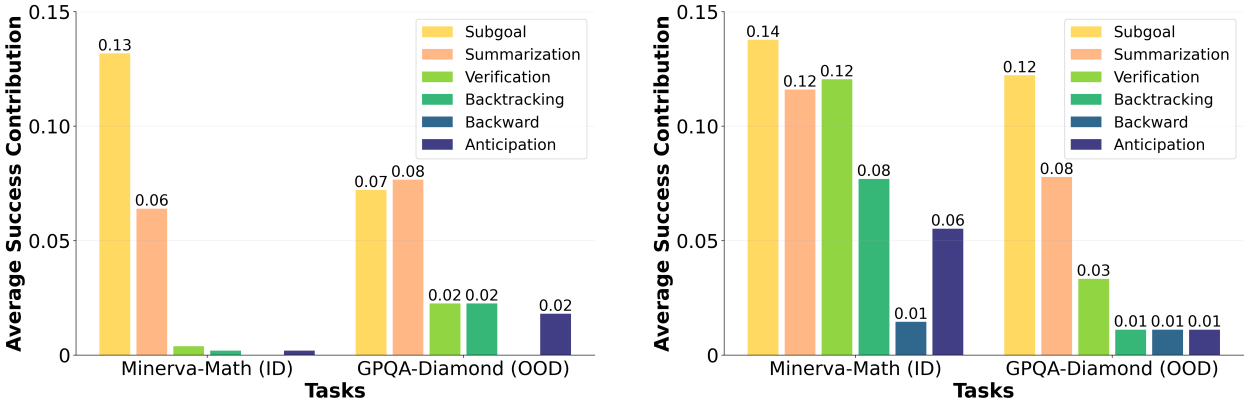
Results. RL with solely numerical feedback frequently encounters performance plateaus. Figure 2 illustrates the RL fine-tuning dynamics of Qwen2.5-7B-Base and Qwen3-8B across varying numbers of training examples. On-policy RL frequently stagnates, as reflected in validation set accuracy: Qwen2.5-7B-Base reaches its highest performance at approximately 45-46% accuracy after 120 steps (Figure 2a), while Qwen3-8B plateaus at 65-67% accuracy after 200 steps (Figure 2b). Notably, both models show minimal improvement in peak performance, even with an 8x increase in the number of training prompts.

Spontaneous self-reflection has limited impact on enhancing problem-solving success. Cognitive behaviors are widely recognized as key contributors to successful complex reasoning (DeepSeek-AI et al.,

[1] GRPO is used without loss of generality, as RL algorithms such as PPO and GRPO exhibit comparable performance.
[2] pass@k measures the percentage of problems where the model produces a correct solution within its first k attempts.
[3] `https://github.com/huggingface/Math-Verify`

2025; Gandhi et al., 2025). In particular, increased self-reflection behaviors after RL fine-tuning, which mimic humans reflecting on past experiences and refining their approach to reach a solution (commonly referred to as the "Aha moment" (DeepSeek-AI et al., 2025)), have drawn significant attention. However, does spontaneous self-reflection play the most critical role in improved performance?

To address this question, we characterize six key cognitive behaviors that contribute to self-improving reasoning during RL fine-tuning: (*i*) *Subgoal Setting*: Decomposing complex problems into smaller, manageable subtasks. (*ii*) *Summarization*: Summarizing the current state by identifying completed subtasks and determining the next steps in reasoning. (*iii*) *Verification*: Systematically checking intermediate results or computations to ensure correctness. (*iv*) *Backtracking*: Identifying errors or dead-ends in reasoning and revising previous methods or approaches. (*v*) *Backward Chaining*: Reasoning from desired outcomes back to the initial inputs or steps required to achieve the result. This is particularly useful in multiple-choice questions where answer options are provided. (Gandhi et al., 2025) (*vi*) *Anticipation*: Anticipating potential inaccuracies or considering alternative solutions to a problem. We refer to the first two behaviors as ***planning behaviors***, while the last four are categorized as ***self-reflection behaviors***. We analyze the contribution of these behaviors to solving problems that the base model was previously unable to solve. For Qwen2.5-7B-Base, we identify 87 previously unsolved problems from the Minerva-Math dataset and 37 from the GPQA-Diamond dataset. For Qwen3-8B, we identify 33 previously unsolved problems from the Minerva-Math dataset and 15 from the GPQA-Diamond dataset. More details can be found in Appendix C.



(a) Reasoning behavior contributions to successful problem-solving in RL-tuned Qwen2.5-7B-Base.

(b) Reasoning behavior contributions to successful problem-solving in RL-tuned Qwen3-8B.

Figure 3: Impact of different reasoning behaviors on solving previously failed problems after RL fine-tuning with numerical feedback for Qwen2.5-7B-Base (left) and Qwen3-8B (right) on test tasks.

Figure 3 presents the average success contribution of various behaviors, showing that planning behaviors are the primary contributors to successful problem-solving, while self-reflection behaviors contribute less in both the mathematical (Minerva-Math) and STEM (GPQA-Diamond) domains. In Figure 3a, spontaneous self-reflection behaviors barely contribute in the mathematical domain for the non-reasoning model. Thanks to extensive training on expert demonstrations with diverse reasoning behaviors in the mathematical domain (Yang et al., 2025), the reasoning model shows that self-reflection behaviors make a noticeable contribution (Figure 3b). Nevertheless, self-correction-related behaviors, such as backtracking, backward chaining, and anticipation, still contribute considerably less. These observations suggest the limited effectiveness of self-reflection. Further analysis on Qwen3-8B-Base is provided in Appendix D.

Models exhibit persistent failures on a subset of problems despite trial-and-error fine-tuning. We evaluated the best-performing RL-finetuned Qwen2.5-7B-Base and Qwen3-8B models on the 4k training prompts. As shown in the left panel of Table 1, these models consistently failed on approximately 29% and 3.75% of problems, respectively, with pass@4 = 0, despite undergoing trial-and-error fine-tuning, where correct responses are rewarded, and incorrect responses are penalized. While the reasoning model (Qwen3-

8B) demonstrates higher performance with considerably fewer persistent failures, the results suggest that both models struggle with certain problems when relying solely on numerical feedback.

A likely cause of these performance plateaus and persistent failures is the sparse informational content of numerical feedback. Scalar rewards often fail to convey *why* a response is correct or incorrect or *how* to improve multi-step reasoning. Furthermore, the limited effectiveness of internally generated self-reflection exacerbates these challenges, making it difficult for models to address problems beyond their existing knowledge boundaries without additional guidance. Together, these limitations highlight the necessity of richer feedback mechanisms to enable more effective learning.

3.2 Promise of Learning from Natural Language Feedback

To move beyond the limitations of purely numerical reward signals, we explore the potential of leveraging natural language feedback to help models identify errors and refine their responses. Specifically, we examine three types of critiques: (*i*) *Indicative Critique*: A heuristic-based critique that merely indicates the binary correctness of the generated solution. (*ii*) *Indicative Critique with Ground Truth (Critique w/ GT)*: A heuristic-based critique that provides both the binary correctness indication and the ground truth answer. (*iii*) *CoT Critique*: A model-generated critique providing step-by-step reasoning to justify correctness or incorrectness, concluding with a binary correctness indication.[4] Examples of these three critique types are shown later.

We summarize the process for leveraging textual critiques to guide the refinement of LLM-generated responses in Algorithm 1. A more detailed description is provided in Appendix F, and an illustrative example of the refinement process with a CoT critique is available in Appendix H.

Algorithm 1 Leveraging Textual Critiques for Refinement of LLM-Generated Responses

Require: LLM π_θ, Reasoning-based reward model π_{RM}, evaluation function Eval, set of questions $Q = \{q\}$, predefined instructions I_c and I_{refine}, number of samples k

Ensure: Refined responses $\{y_{\text{refined}}^{(j)}\}$ for persistently failed questions

for each question $q \in Q$ **do** # Step 1: Initial Response Sampling
  Sample k initial responses: $\{y_0^{(i)}\}_{i=1}^k \sim \pi_\theta(\cdot \mid q)$
end for
for each question $q \in Q$ **do** # Step 2: Response Evaluation and Critique Generation
  for each initial response $y_0^{(i)} \in \{y_0^{(i)}\}_{i=1}^k$ **do**
    Generate CoT critique: $c_{\text{CoT}}^{(i)} \sim \pi_{RM}(\cdot \mid I_c, q, y_0^{(i)})$
    Evaluate correctness: $\text{Eval}(q, y_0^{(i)}) \in \{0, 1\}$
    if $\text{Eval}(q, y_0^{(i)}) = 0$ **then**
      Construct heuristic-based critiques: $c_{\text{I}}^{(i)}$ (indicative) and $c_{\text{GT}}^{(i)}$ (with ground truth)
    end if
  end for
end for
Identify persistently failed questions:
  $Q_{\text{failed}} \leftarrow \{q \in Q \mid \forall i : \text{Eval}(q, y_0^{(i)}) = 0\}$
Form triplets $(q, y_0^{(j)}, c^{(j)})$ for each $q \in Q_{\text{failed}}$ and each $y_0^{(j)}$, where $c^{(j)} \in \{c_{\text{CoT}}^{(j)}, c_{\text{GT}}^{(j)}, c_{\text{I}}^{(j)}\}$
for each triplet $(q, y_0^{(j)}, c^{(j)})$ **do** # Step 3: Self-Refinement Generation
  Generate refined response: $y_{\text{refined}}^{(j)} \sim \pi_\theta(\cdot \mid I_{\text{refine}}, q, y_0^{(j)}, c^{(j)})$
end for
  return Refined responses $\{y_{\text{refined}}^{(j)}\}$ for all $(q, y_0^{(j)}, c^{(j)})$

Setup. As described in the previous section, We evaluate the best-performing RL-finetuned Qwen2.5-7B-Base and Qwen3-8B models, generating four responses per question for a set of 4k prompts. Persistently

[4]The CoT critique includes step-by-step evaluation and an improvement recipe but does not contain an expert demonstration.

failed question-response pairs are identified as those with pass@4 = 0. For each such pair, we prompt GPT-4o (Hurst et al., 2024) to generate a CoT critique and then construct the two corresponding heuristic-based critiques. Examples of these critique types are shown below. Subsequently, we prompt the models to refine these failed responses.

Evaluation Metrics. To assess the efficacy of the critique and refinement process, we define the following metrics: (*i*) **Valid Critiques Generated**: The percentage of initially erroneous responses for which the critique model (π_ϕ) successfully generated a well-formed and usable critique. This accounts for potential failures in the critique generation process itself. (*ii*) **Successful Refinements**: The percentage of initially erroneous responses (that received a valid critique) for which the LLM (π_θ) subsequently generated a correct refined response. (*iii*) **Critiqued Questions**: The percentage of unique questions (all of whose initial k responses were erroneous) for which at least one of their erroneous responses received a valid critique. (*iv*) **Questions Successfully Refined**: The percentage of unique questions (all of whose initial k responses were erroneous) for which at least one of their erroneous responses was successfully corrected through the refinement process. We evaluate the correctness of the refinements using the rule-based method described earlier.

CoT Critique

Let's analyze the student's solution step-by-step and identify any errors:

Step 1: Geometry Understanding The student correctly identifies the geometry of the problem, noting that the sphere is tangent to the base and sides of the cone, and that the axial section forms an isosceles triangle.

Step 2: Variable Definitions The student defines r as the radius of the sphere, R as the radius of the base of the cone, and h as the height of the cone. These definitions are appropriate.

Step 3: Relating Sphere and Cone The student attempts to relate r, R, and h using similar triangles. However, the equation derived:

$$r = \frac{Rh}{h + R}$$

is not directly applicable to the problem as it does not consider the specific condition that the areas are equal.

Step 4: Equating Areas The student correctly equates the areas of the sphere and the base of the cone:

$$4\pi r^2 = \pi R^2 \implies R = 2r$$

This step is correct.

Step 5: Substitution and Error The student substitutes $R = 2r$ into the earlier equation, leading to:

$$r = \frac{(2r)h}{h + 2r}$$

This substitution is unnecessary and results in incorrect simplification. The correct approach would involve using the inradius formula for the triangle formed by the axial section of the cone.

Step 6: Angle Calculation The student calculates $\tan\theta = 1$ and concludes $\theta = 45°$, leading to $\cos(90°) = 0$. This is incorrect because the angle calculation should be based on the relationship between h and R, derived from the inradius formula.

Step 7: Precise Calculation The student attempts to verify the angle calculation but repeats the same incorrect logic, leading to the same incorrect conclusion.

Correct Solution Approach The correct approach involves using the inradius formula for the triangle formed by the axial section of the cone: - Calculate the inradius using the area and semi-perimeter of the triangle. - Substitute $R = 2r$ and solve for h in terms of r. - Relate h and θ using trigonometric identities. - Calculate $\cos(2\theta)$ using the double angle identity.

The correct final answer is:

$$\cos(2\theta) = \frac{7}{25}.$$

Conclusion: incorrect [END]

Indicative Critique The generated solution is incorrect.

Indicative Critique w/ GT The generated solution is incorrect, the ground truth is $\frac{7}{25}$.

Table 1: Analysis of performance gains from critique-based self-refinement.

Method	% Failed Questions (Pass@4=0)	Critique Type	% Valid Critiques	% Valid Refinements	% Critiqued Questions	% Questions Refined
RL-finetuned Qwen2.5-7B-Base	29.07	Indicative Critique	100.00	2.09	100.00	7.05
		Indicative Critique w/ GT	100.00	1.98	100.00	6.88
		CoT Critique	60.06	**36.47**	95.10	**55.37**
RL-finetuned Qwen3-8B (w/ Thinking)	3.75	Indicative Critique	100.00	3.33	100.00	8.67
		Indicative Critique w/ GT	100.00	3.67	100.00	10.67
		CoT Critique	50.17	**10.63**	88.67	**20.00**

Results. CoT Critiques facilitate effective model refinement. Table 1 shows that refinement guided by CoT critiques achieves the highest valid refinement rate (36.47% and 10.63%) and the largest percentage of successfully refined questions (55.37% and 20.00%) on Qwen2.5-7B-Base and Qwen3-8B, respectively. This performance significantly surpasses refinement based on indicative critiques or critiques with ground truth, even though the CoT critique generation process produces valid critiques for only 60.06% and 50.17% of erroneous responses. The effectiveness of CoT critiques can be attributed to their richness: by providing a step-by-step explanation of the reasoning *along with* targeted guidance on the correct solution approach, they offer substantially more informative feedback than simpler alternatives.

Binary correctness signals alone can provide refinement benefits. Refinement with indicative critiques with/without ground truth also yields some successful refinements, albeit at a substantially lower rate (approximately 2%-4% valid refinements). This suggests that even simply indicating the correctness of a response can provide a minimal benefit, indicating some promise in leveraging natural language feedback to augment learning from numerical signals. However, the lack of a substantial difference between indicative critiques and critiques with ground truth suggests that providing only the ground-truth answer, without any explanation or reasoning, provides little additional guidance to the model. The model appears to struggle to effectively leverage the ground truth answer *without* an understanding of *why* the initial response was incorrect or *how* to arrive at the correct solution. Additional results on Qwen3-8B-Base are shown in Appendix D.

4 Critique-GRPO

Motivated by the potential of leveraging critiques, particularly CoT critiques, for effective model refinement (Section 3), we introduce Critique-GRPO, an online optimization algorithm that learns from both natural language and numerical feedback. As illustrated in Figure 4, Critique-GRPO facilitates effective online learning and exploration by enabling the model to learn from both its generated responses and its effective refinements by incorporating natural language feedback (specifically, critiques). Before delving into the details of Critique-GRPO (Section 4.1), we briefly review Group Relative Policy Optimization (GRPO) (Shao et al., 2024) (Section 4.2).

4.1 From GRPO to Critique-GRPO

GRPO is an online RL algorithm widely used during the fine-tuning stage of LLMs. It builds on Proximal Policy Optimization (PPO) (Schulman et al., 2017), but eliminates the need for value function approximation by estimating advantages based on the relative performance of groups of actions. In the context of LLM policy optimization, let the model policy be parameterized by θ. For each question q in a given set Q, a group of responses $\{y^{(i)}\}_{i=1}^{n}$ is sampled from the old policy π_{old}. A reward model then scores these responses, yielding rewards $\{R^{(i)}\}_{i=1}^{n}$. The GRPO training objective is formulated as:

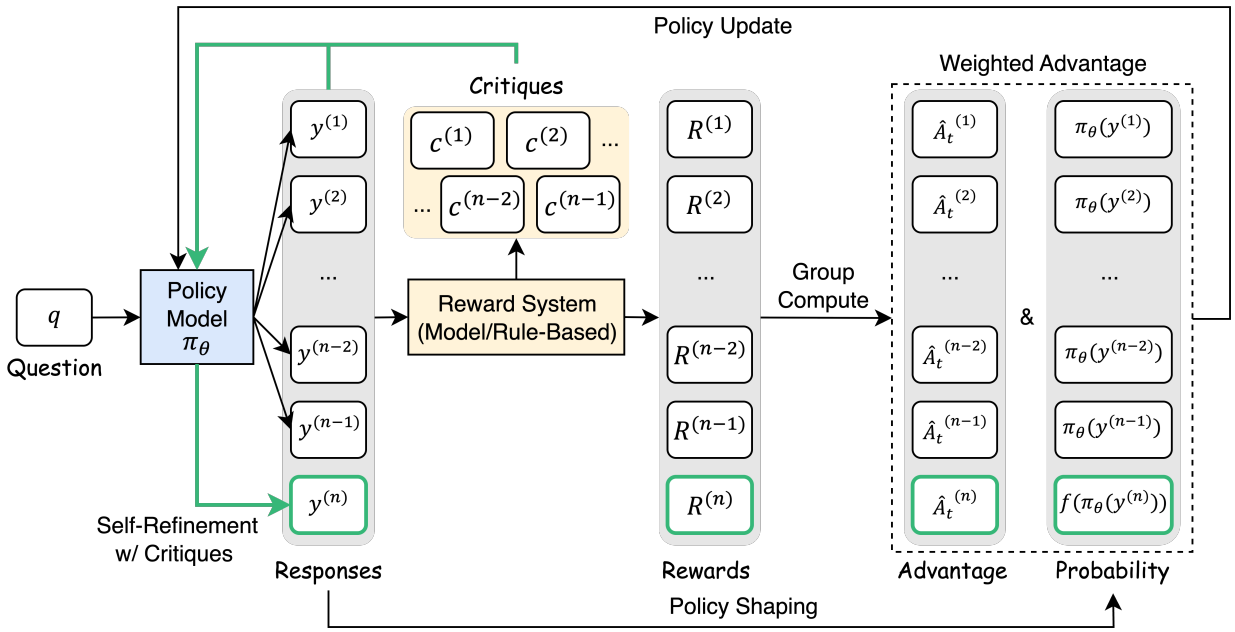


Figure 4: Overview of Critique-GRPO. Given a question, Critique-GRPO samples initial responses and then refines these responses using critiques generated by a reward system (either model-based or rule-based). These refinements are combined with the initial responses to optimize the policy within an online reinforcement learning framework. A weighted advantage function, combined with policy shaping, emphasizes correct refinements while strongly penalizing incorrect ones.

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim Q,\{y^{(i)}\}_{i=1}^{n}\sim\pi_{\text{old}}(\cdot|q)} \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|y^{(i)}|}\sum_{t=1}^{|y^{(i)}|}\left\{\min\left[r_t^{(i)}(\theta)\hat{A}_t^{(i)}, \text{clip}(r_t^{(i)}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{(i)}\right] - \beta D_{\text{KL}}[\pi_\theta||\pi_{\text{ref}}]\right\}, \tag{1}$$

where $r_t^{(i)}(\theta)$ is the probability ratio, comparing the current policy π_θ to the old policy π_{old} from which the responses were sampled:

$$r_t^{(i)}(\theta) = \frac{\pi_\theta(y_t^{(i)}|q, y_{<t}^{(i)})}{\pi_{\text{old}}(y_t^{(i)}|q, y_{<t}^{(i)})}, \quad \text{where } r_t^{(i)}(\theta_{\text{old}}) = 1, \tag{2}$$

Here, ϵ and β are hyperparameters. The term ϵ controls the range of the clipped probability ratio, enforcing a pessimistic lower bound on policy performance to prevent excessively large policy updates. Meanwhile, β regulates the KL divergence penalty, constraining the trained policy from deviating significantly from the reference policy.

The advantage $\hat{A}_t^{(i)}$ for all tokens in a response is calculated by normalizing the rewards $\{R^{(i)}\}_{i=1}^{n}$ using the group mean and standard deviation:

$$\hat{A}_t^{(i)} = \frac{R^{(i)} - \text{mean}(\{R^{(1)}, \ldots, R^{(n)}\})}{\text{std}(\{R^{(1)}, \ldots, R^{(n)}\})}. \tag{3}$$

Recent work (Liu et al., 2025a) suggests that the token-level normalization and the standard deviation term in the advantage calculation (highlighted in gray) may introduce biased optimization. Following their implementation, we remove these terms to obtain an unbiased optimization objective.

4.2 Online Learning with Critique-GRPO

We introduce Critique-GRPO, an online policy optimization framework that enables a model to learn simultaneously from its generated responses and their refinements by incorporating critiques generated by a reasoning-based reward model. This approach helps the model discover new solutions and mitigate repeated failures on specific questions, as discussed in Section 3.

Specifically, Critique-GRPO operates in three main steps (Figure 4):

Step 1: Initial Response Sampling. Given an LLM and a set of questions $Q = \{q\}$, we sample k initial responses for each question from the old policy π_{old}: $\{y^{(i)}\}_{i=1}^{n} \sim \pi_{\text{old}}(\cdot \mid q)$. These responses are evaluated using a reward system to generate both critiques $\{c^{(i)}\}_{i=1}^{n}$ and scalar rewards $\{R^{(i)}\}_{i=1}^{n}$, formulated as:

$$c^{(i)}, R^{(i)} \leftarrow \text{Reward}(q, y^{(i)}), \forall i.$$

We consider two types of reward systems: *model-based* (the primary focus of this work) and *rule-based* (a variant). (*i*) For the model-based reward system, we use a reasoning-based reward model π_{RM} to generate CoT critiques: $c_{\text{CoT}}^{(i)} \sim \pi_{RM}(\cdot \mid I_c, q, y^{(i)})$, where I_c is the critique instruction. These critiques, based on question-response pairs, are described in Section 3. Binary correctness labels from the critiques are converted into scalar reward scores: $R^{(i)} \leftarrow c_{\text{CoT}}^{(i)}$. (*ii*) For the rule-based reward system, we compare the generated responses against ground-truth answers using a string-matching function to compute scalar rewards: $R^{(i)} = \text{is_equivalent}(y^{(i)}, y_{\text{GT}})$.[5] From these evaluations, two heuristic-based critiques, *i.e.,* $c_{\text{I}}^{(i)}$ (indicative critique) and $c_{\text{GT}}^{(i)}$ (critique with ground truth) are derived: $c_{\text{I}}^{(i)}, c_{\text{GT}}^{(i)} \leftarrow R^{(i)}$, as detailed in Section 3.

Step 2: Critique-Guided Self-Refinement. Next, we prompt the LLM to generate refined responses conditioned on the question-response-critique triplet $(q, y^{(i)}, c^{(i)})$ and a refinement instruction I_{refine} (detailed in Appendix E): $y_{\text{refined}}^{(i)} \sim \pi_{\text{old}}(\cdot \mid I_{\text{refine}}, q, y^{(i)}, c^{(i)})$, where $c^{(i)} \in \{c_{\text{CoT}}^{(i)}, c_{\text{GT}}^{(i)}, c_{\text{I}}^{(i)}\}$. The reward model scores these self-generated refinements, producing $\{R_{\text{refine}}^{(i)}\}_{i=1}^{n}$ (alternatively, the rule-based evaluation function discussed in Section 3 could be used). To mitigate potential distributional shifts induced by the refinements, we randomly sample a subset of k refinements, denoted by $\{y_{\text{refined}}^{(i')}\}_{i'=1}^{k}$, from the full refinement set $\{y_{\text{refined}}^{(i)}\}_{i=1}^{n}$. This sampling prioritizes correct refinements; if no correct refinements are generated, incorrect refinements are sampled randomly. We then combine the sampled subset of refinements with the initial responses to form a mixed group of responses.[6]

Step 3: Online Policy Optimization. Finally, the model is fine-tuned on a mixed set of initial and refined responses using scalar rewards. The training objective, adapted from GRPO, is given by:

$$\begin{aligned}\mathcal{J}_{\text{Critique-GRPO}}(\theta) =& \mathbb{E}_{q\sim Q, \{y^{(i)}\}_{i=1}^{n}\sim\pi_{\text{old}}(\cdot|q), \{y_{\text{refined}}^{(i')}\}_{i'=1}^{k}\sim\pi_{\text{old}}(\cdot|I_{\text{refine}}, q, y^{(i')}, c^{(i')})} \\ & \Bigg[\underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{|y^{(i)}|} r_t^{(i)}(\theta) A_t^{(i)}}_{\text{Objective for Initial Responses}} + \underbrace{\frac{1}{k}\sum_{i'=1}^{k}\sum_{t=1}^{|y_{\text{refined}}^{(i')}|} f\left(r_{\text{refined},t}^{(i')}(\theta)\right) A_t^{(i')}}_{\text{Objective for Refined Responses}} \Bigg],\end{aligned} \tag{4}$$

where the advantages $A_t^{(i)}$, $A_t^{(i')}$ for all tokens in a response are defined as:

$$\begin{aligned} A_t^{(i)} &= R^{(i)} - \text{mean}\big(\{R^{(j)}\}_{j=1}^{n} \cup \{R_{\text{refined}}^{(j')}\}_{j'=1}^{k}\big), \\ A_t^{(i')} &= R^{(i')} - \text{mean}\big(\{R^{(j)}\}_{j=1}^{n} \cup \{R_{\text{refined}}^{(j')}\}_{j'=1}^{k}\big), \end{aligned} \tag{5}$$

$r_t^{(i)}(\theta)$ and $f(r_{\text{refined},t}^{(i')}(\theta))$ represent the token-level probability ratios:

[5] To ensure consistency, we align the results of model- and rule-based evaluations, isolating the effects of incorporating natural language feedback.

[6] Currently, only one refined response is retained. Future work may explore the optimal data ratio.

$$r_t^{(i)}(\theta) = \frac{\pi_\theta(y_t^{(i)}|q, y_{<t}^{(i)})}{\pi_{\text{old}}(y_t^{(i)}|q, y_{<t}^{(i)})}, \quad f(r_{\text{refined},t}^{(i')}(\theta)) = \frac{\pi_\theta(y_{\text{refined},t}^{(i')}|q, y_{\text{refined},<t}^{(i')})}{\pi_\theta(y_{\text{refined},t}^{(i')}|q, y_{\text{refined},<t}^{(i')}) + \gamma}. \tag{6}$$

The shaping function $f(\cdot)$ (Yan et al., 2025) ($0 < \gamma < 1$), depicted in the lower right corner of Figure 4, reweights the gradients to assign greater importance to low-probability tokens in refined responses. This encourages the model to effectively learn from unfamiliar (*i.e.,* low-probability under the current policy) but correct refinements while penalizing failed refinements. In addition, we remove the clipping function for probability ratios and the KL-divergence penalty term (present in the original GRPO formulation) to reduce restrictions on policy updates. This enables more substantial model adjustments and facilitates effective learning from refinements. We summarize Critique-GRPO in Algorithm 2 (Appendix G).

5 Experiments

In this section, we evaluate the efficacy of Critique-GRPO on challenging mathematical, scientific and general reasoning tasks.

5.1 Experimental Setup

Datasets and Evaluation Metrics. We use randomly sampled subsets of 4k examples from a reorganized 45k subset (Yan et al., 2025) of OpenR1-Math-220k (Bakouch et al., 2025) as the training set (as described in Section 3). For validation, we use the curated validation set provided by (Yan et al., 2025). We evaluate the model on five well-established mathematical reasoning benchmarks: MATH-500 (Hendrycks et al., 2021), Minerva-Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024a), MATH (Hendrycks et al., 2021), AIME 2024 (Li et al., 2024), AIME 2025 (Li et al., 2024), and AMC 2023 (Li et al., 2024). For broader analysis, we assess the model's generalization ability on three scientific and general reasoning tasks: TheoremQA (Math, Physics, EE&CS, and Finance) (Chen et al., 2023), GPQA-Diamond (Physics, Chemistry, and Biology) (Rein et al., 2024), and MMLU-Pro (Business, Computer Science, Law, *etc.*) (Wang et al., 2024). During evaluation, we use greedy decoding (temperature = 0) and report pass@1 over three runs.

Reward Design. During RL fine-tuning, we use model-based evaluation to generate critiques and rule-based evaluation to provide binary scalar rewards, as described in Section 3.

Compared Methods. We compare Critique-GRPO against the following representative approaches, categorized into supervised learning and reinforcement learning-based finetuning. All differences are considered significant at $p < 0.01$.

Supervised Learning-based Finetuning:

(*i*) *Supervised Finetuning (SFT)*: Finetuning the base model on high-quality annotated training data using supervised learning.

(*ii*) *Reward rAnked Finetuning (RAFT)* (Dong et al., 2023): Finetuning on self-generated correct responses, sampled based on rule-based evaluation.

(*iii*) *Refinement Finetuning (Refinement FT)* (Chen et al., 2024): Finetuning on refined correct responses generated conditionally on the question, initial response, and CoT critiques.

(*iv*) *Critique Finetuning (Critique FT)* (Xi et al., 2024): Finetuning on annotated CoT critique data to train the model to critique a given query-response.

(*v*) *Critique-in-the-Loop Finetuning (CITL-FT)* (Xi et al., 2024): Finetuning on mixed data consisting of self-generated correct responses and refined correct responses, conditioned on the question-initial response-CoT critique triplet.

Reinforcement Learning-based Finetuning:

(*vi*) *R1-GRPO* (DeepSeek-AI et al., 2025): Finetuning the base model on its own generations using the GRPO algorithm with binary scalar rewards.

(*vii*) *R1-Dr.GRPO* (Liu et al., 2025a): Finetuning the base model on its own generations using the Dr.GRPO algorithm, which removes terms that cause biased optimization, with binary scalar rewards.

(*viii*) *Critique-GRPO (Indicative Critique)*: A variant of our framework that utilizes indicative critiques (as discussed in Section 3) for refinements.

(*ix*) *Critique-GRPO (Critique with Ground Truth)*: A variant of our framework that utilizes indicative critiques paired with ground-truth answers (as discussed in Section 3) for refinements.

Implementation Details. We conduct experiments using Qwen2.5-7B-Base, Qwen2.5-Math-7B-Base, and Qwen3-8B, with GPT-4o (which can be replaced by other reasoning-based reward models) serving as the reasoning-based reward model, as outlined in Section 3. For supervised finetuning baselines, models are finetuned until convergence, and the best performance is reported. For reinforcement learning approaches, models are finetuned for 400 steps, and the best performance is recorded. To ensure a fair comparison: In R1-GRPO, 8 responses (rollouts) are sampled per training prompt with temperature = 1. In LUFFY, 7 responses are sampled per prompt along with one ground truth response (expert demonstration). In Critique-GRPO, 7 responses are sampled per prompt, along with one refined response from the refinement sets. All experiments are conducted on 40 NVIDIA A800 80G GPUs. To ensure consistency, we only use critiques generated by the reward model that align with rule-based evaluations; otherwise, the reward model is prompted to regenerate the critiques. More implementation details are provided in Appendix B.

5.2 Main Results

Table 2 presents the evaluation results, highlighting the following key observations:

Incorporating natural language feedback (critiques) into online reinforcement learning enhances policy optimization. Critique-GRPO consistently outperforms both supervised learning-based and RL-based fine-tuning approaches on Qwen2.5-7B-Base and Qwen3-8B across in-distribution and out-of-distribution tasks. Specifically, Critique-GRPO (CoT critique) improves state-of-the-art (SOTA) average pass@1 scores by approximately 4.4 points (42.66% → 47.08%) on Qwen2.5-7B-Base and 3.8 points (64.46% → 68.26%) on Qwen3-8B.

Online self-refinements are more effective than offline self-refinements. Critique-GRPO (CoT critique) substantially outperforms Refinement FT by approximately 11.9 points (47.08% *vs.* 35.21%) and 8.81 points (68.26% *vs.* 59.45%) in average pass@1 on Qwen2.5-7B-Base and Qwen3-8B, respectively. Furthermore, it surpasses CITL-FT by approximately 11.4 points (47.08% *vs.* 35.66%) and 12.4 points (68.26% *vs.* 55.84%) on Qwen2.5-7B-Base and Qwen3-8B, respectively.

Incorporating online refinements facilitates effective policy optimization. Critique-GRPO with three types of critiques consistently outperforms R1-GRPO and R1-Dr.GRPO on nearly all tasks. Notably, Critique-GRPO (CoT critique) achieves an average Pass@1 improvement of +5.9% over R1-GRPO and +4.4% over R1-Dr.GRPO on Qwen2.5-7B-Base. Similarly, on Qwen3-8B, it achieves improvements of +4.5% over R1-GRPO and +3.8% over R1-Dr.GRPO. This confirms the effectiveness of natural language feedback in guiding the model to explore valid responses for problem-solving, consistent with the findings in Section 3. Furthermore, Critique-GRPO significantly enhances the model's generalization ability, particularly in science and general reasoning tasks.

Higher-quality refinements, guided by richer critiques, lead to more effective policy optimization. Critique-GRPO (CoT critique) consistently outperforms its two variants across all tasks, achieving average pass@1 gains of +1.8-2.4% and +2-2.3% on Qwen2.5-7B-Base and Qwen3-8B, respectively. This improvement arises from the superior ability of CoT critiques to facilitate effective model refinements compared to binary correctness signals, with or without ground-truth answers, as discussed in Section 3. The detailed guidance provided by CoT critiques enables more precise and impactful policy updates.

Table 2: Zero-shot evaluation results (Pass@1) on Mathematical reasoning (ID) and Scientific and General (OOD) reasoning tasks. "Expert Demo. (Demonstration)" refers to ground-truth CoT generated by Deepseek-R1, as described in Section 3. "Num. Feedback" and "Lang. Feedback" denote numerical and natural language feedback, respectively.

Method	Supervision			Math (ID)					Science & General (OOD)			Avg.
	Expert Demo.	Num. FB	Lang. FB	MATH 500	Minerva MATH	Olympiad Bench	AMC23	AIME24	Theorem QA	GPQA Diamond	MMLU Pro	
Non-Reasoning Model												
Qwen2.5-7B-Base	-	-	-	60.80	20.20	30.40	35.00	13.30	21.60	28.79	46.24	32.04
Supervised Learning-based Finetuning												
+ SFT	✓	×	×	61.60	24.30	23.40	40.00	6.70	26.50	30.30	51.49	33.04
+ RAFT	×	✓	×	67.00	19.50	32.40	50.00	10.00	24.40	23.74	47.12	34.27
+ Refinement FT	×	✓	✓	65.80	21.30	32.10	47.50	13.30	24.40	29.80	47.51	35.21
+ Critique FT	×	×	✓	66.00	19.10	29.30	47.50	13.3	29.60	28.79	44.46	34.76
+ CITL-FT	×	✓	✓	70.20	19.90	30.70	42.50	16.70	28.70	28.28	48.31	35.66
Reinforcement Learning-based Finetuning												
+ R1-GRPO	×	✓	×	74.00	32.00	38.50	42.50	16.70	40.60	33.33	51.81	41.18
+ R1-Dr.GRPO	×	✓	×	**78.40**	34.90	39.90	40.00	13.30	43.10	**38.89**	52.83	42.66
+ Critique-GRPO (Ours) (Indicative Critique)	×	✓	✓	76.00	36.00	41.00	55.00	13.30	41.80	37.88	**55.97**	44.62
+ Critique-GRPO (Ours) (Critique w/ GT)	×	✓	✓	76.80	35.70	39.60	62.50	10.00	44.00	**38.89**	54.88	45.30
+ Critique-GRPO (Ours) (CoT Critique)	×	✓	✓	77.80	**36.80**	**42.40**	**62.50**	**20.00**	**44.00**	37.88	55.28	**47.08**
Reasoning Model (w/ Thinking)												
Qwen3-8B	-	-	-	82.00	41.20	44.10	67.50	40.00	46.90	35.86	69.31	53.36
Supervised Learning-based Finetuning												
+ SFT	✓	×	×	83.20	43.80	46.40	82.50	40.00	48.90	38.38	66.81	56.25
+ RAFT	×	✓	×	82.80	44.10	46.40	75.00	36.70	46.80	37.88	69.00	54.84
+ Refinement FT	×	✓	✓	87.40	46.00	54.50	80.00	40.00	55.40	45.45	66.82	59.45
+ Critique FT	×	×	✓	84.40	37.10	49.80	80.00	36.70	46.40	35.35	64.10	54.23
+ CITL-FT	×	✓	✓	85.00	43.00	46.80	70.00	43.30	48.00	41.92	68.73	55.84
Reinforcement Learning-based Finetuning												
+ R1-GRPO	×	✓	×	91.00	52.60	65.60	82.50	50.00	57.90	40.40	70.00	63.75
+ R1-Dr.GRPO	×	✓	×	91.20	51.10	63.60	82.50	53.30	59.00	44.44	70.51	64.46
+ Critique-GRPO (Ours) (Indicative Critique)	×	✓	✓	91.00	47.80	63.30	85.00	**63.30**	**60.40**	47.47	70.00	66.03
+ Critique-GRPO (Ours) (Critique w/ GT)	×	✓	✓	92.00	50.00	66.80	87.50	56.70	59.00	47.47	**70.87**	66.29
+ Critique-GRPO (Ours) (CoT Critique)	×	✓	✓	**92.00**	**52.90**	**66.80**	**92.50**	**63.30**	60.10	**47.98**	70.47	**68.26**

5.3 Investigation on Math-Centric Backbone Models

Table 3: Investigation of RL finetuning with Critique-GRPO on Qwen2.5-Math-7B-Base (Yang et al., 2024). Results marked with an asterisk are cited from Yan et al. (2025).

Method	Math (ID)					Science & General (OOD)			Avg.
	MATH 500	Minerva MATH	Olympiad Bench	AMC23	AIME24	Theorem QA	GPQA Diamond	MMLU Pro	
Qwen2.5-Math-7B-Base	51.20	13.20	17.60	47.50	13.30	26.40	26.77	39.70	29.46
+ SimpleRL-Zero*	76.00	25.00	34.70	54.90	27.00	-	23.20	34.50	-
+ PRIME-Zero*	81.40	**39.00**	40.30	54.00	17.00	-	18.20	32.70	-
+ Oat-Zero*	78.00	34.60	43.40	61.20	**33.40**	-	23.70	41.70	-
+ Critique-GRPO (Ours) (CoT-Critique)	**83.20**	**39.00**	**44.00**	**67.50**	26.70	**51.40**	**40.40**	**43.79**	**49.50**

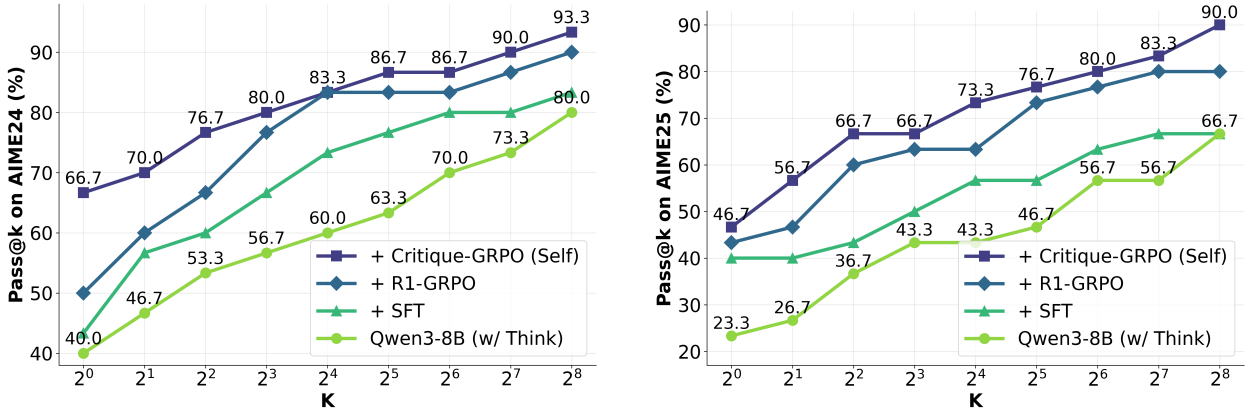
We evaluate the efficacy of RL fine-tuning with Critique-GRPO on the math-centric backbone model, Qwen2.5-Math-7B-Base. Specifically, we compare its performance against three representative RL fine-tuning approaches: (*i*) *SimpleRL-Zero* (Zeng et al., 2025): an open-source reproduction of R1-GRPO; (*ii*) *PRIME-Zero* (Cui et al., 2025a): fine-tuning the base model with both outcome rewards and process rewards; (*iii*) *Oat-Zero* (Liu et al., 2025a): fine-tuning the base model with Dr.GRPO.

As shown in Table 3, Critique-GRPO significantly improves the base model with an average pass@1 gain of +20%, clearly outperforming these representative RL fine-tuning approaches on nearly all tasks, *e.g.,* achieving a +6.3% pass@1 gain on AMC23. This result aligns with the performance gains observed on general-purpose backbone models.

5.4 Self-Improvement via Self-Critiquing

Table 4: Potential for self-improvement through RL fine-tuning using Critique-GRPO with self-generated CoT critiques (self-critiquing) on Qwen3-8B.

Method	w/ External Supervision	Math (ID)					Science & General (OOD)			Avg.
		MATH 500	Minerva MATH	Olympiad Bench	AMC23	AIME24	Theorem QA	GPQA Diamond	MMLU Pro	
Qwen3-8B (w/ Think)	-	82.00	41.20	44.10	67.50	40.00	46.90	35.86	69.31	53.36
+ SFT	✓	83.20	43.80	46.40	82.50	40.00	48.90	38.38	66.81	56.25
+ R1-GRPO	✓	91.00	**52.60**	65.60	82.50	50.00	57.90	40.40	70.00	63.75
+ Critique-GRPO (Ours) (Self-Critique& Self-Evaluation	×	92.00	52.20	65.50	87.50	53.30	59.80	47.47	**70.93**	66.09
+ Critique-GRPO (Ours) (Self-Critique)	✓	**92.60**	**52.60**	**66.20**	**95.00**	**60.00**	**60.60**	**47.98**	70.03	**68.13**



(a) Pass@k on AIME24 of Qwen3-8B.

(b) Pass@k on AIME25 of Qwen3-8B.

Figure 5: Comparison of Pass@k for self-improvement using RL fine-tuning via Critique-GRPO (self-critique), compared to methods relying on external numerical feedback (R1-GRPO), expert demonstrations (SFT), the base model Qwen3-8B.

To explore the potential of Critique-GRPO in enabling an LLM's self-improvement through self-critiquing, we prompt the model itself to serve as a reasoning-based reward model. Specifically, we investigate two types of self-critiquing: (*i*) *Self-critique*, where the model evaluates the correctness of its own responses using CoT critiques with ground truth answers as reference; and (*ii*) *Self-critique & self-evaluation*, where the model evaluates its responses using CoT critiques *without* any reference (Zhang et al., 2024). These approaches result in Critique-GRPO (self-critique) and Critique-GRPO (self-critique & self-evaluation), respectively. Details of the prompts are provided in Appendix E. Table 4 shows the evaluation results on Qwen3-8B, and Figure 5 presents pass@k performance changes on AIME24 and AIME25 (Li et al., 2024).

Critique-GRPO enhances self-improvement through self-critiquing. Table 4 RL fine-tuning with Critique-GRPO (self-critique) significantly outperforms fine-tuning with GRPO using external numerical feedback (R1-GRPO) and supervised fine-tuning with expert demonstrations (SFT). On average, Critique-GRPO (self-critique) improves pass@1 by +4.5% and +12.0% compared to R1-GRPO and SFT, respectively. Additionally, the unsupervised approach—Critique-GRPO (self-critique & self-evaluation)—achieves an average pass@1 improvement of 2.3% over R1-GRPO, highlighting the potential of leveraging self-critique for self-improvement *without any external supervision.*

Self-critiquing aids effective exploration. Figure 5 highlights the consistently superior performance of Critique-GRPO (self-critique) across pass@k metrics, with k ranging from 1 to 256, indicating genuine improvements. Notably, Critique-GRPO (self-critique) achieves remarkable gains over R1-GRPO for pass@k with $k = 1$ to 4, yielding improvements of 10-16.7% on AIME24 (Figure 5a).

5.5 Exploration of Weak-to-Strong Generalization

Table 5: Exploration of RL fine-tuning with Critique-GRPO for weak-to-strong generalization on Qwen3-8B. Refinements, termed as "weaker refinement," are generated by a weaker model.

Method	Math (ID)					Science & General (OOD)			Avg.
	MATH 500	Minerva MATH	Olympiad Bench	AMC23	AIME24	Theorem QA	GPQA Diamond	MMLU Pro	
Qwen3-8B (w/ Think)	82.00	41.20	44.10	67.50	40.00	46.90	35.86	69.31	53.36
+ R1-GRPO	**91.00**	**52.60**	**65.60**	82.50	50.00	57.90	40.40	70.00	63.75
+ Critique-GRPO (Ours) (Weaker Refinement via Critique w/ GT)	90.40	50.70	64.90	**85.00**	**60.00**	**59.00**	**43.43**	**70.94**	**65.55**

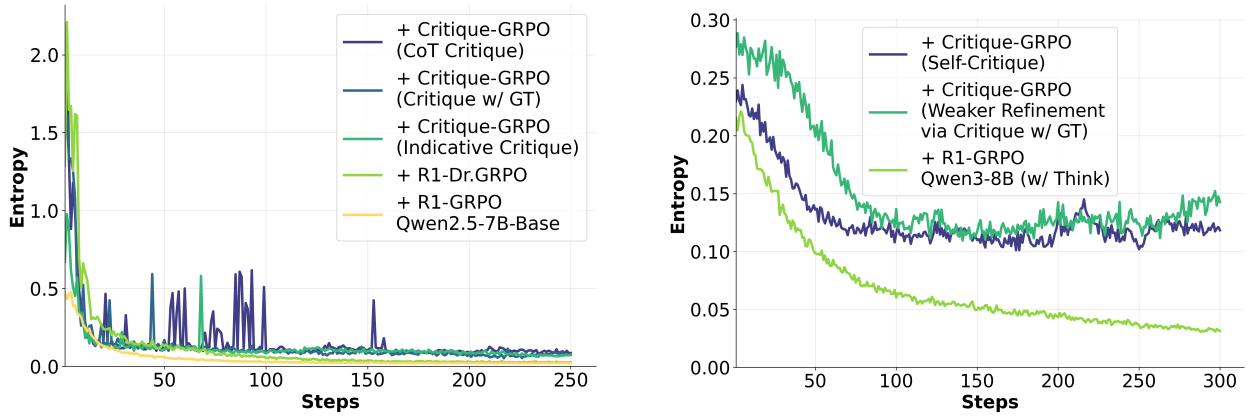
We investigate the potential of weak-to-strong generalization (Burns et al., 2023) using Critique-GRPO, where a strong model learns from refinements generated by a weaker teacher model. Specifically, we use Qwen3-8B-Base (Yang et al., 2025) as the weaker teacher to generate refinements based on indicative critiques with the ground truth answers, guiding the improvement of Qwen3-8B.

Critique-GRPO enables effective weak-to-strong generalization. As shown in Table 5, Critique-GRPO (weaker refinement via critique with ground truth) achieves a +12.2% average pass@1 improvement over Qwen3-8B and outperforms R1-GRPO (65.55% *vs.* 63.75%). This demonstrates that refinements from a weaker model can significantly enhance the performance of a stronger model.

5.6 Investigation of Policy Exploration During RL Finetuning

To investigate policy exploration, we analyze two primary aspects of our RL-finetuned models: (*i*) entropy dynamics during RL fine-tuning for self-improvement using compared RL-based finetuning approaches on Qwen2.5-7B-Base and Critique-GRPO (self-critique) on Qwen3-8B (Figure 6), and (*ii*) changes in response length during fine-tuning (Figure 7).

Learning from natural language feedback helps sustain exploration. As shown in Figure 6a, the policy entropy of Critique-GRPO generally remains higher than that of R1-GRPO and R1-Dr.GRPO, suggesting more consistent exploration. The peaks in Critique-GRPO's entropy dynamics (before step 200) likely occur when its self-generated refinements deviate significantly from the initial sampled responses, leading to increased entropy and potentially beneficial distributional shifts. The subsequent decrease in entropy indicates that the model quickly internalizes these refinements, reducing the distributional deviation. This dynamic aligns with the observation that rare actions with high advantage can increase policy entropy (*i.e.,* unfamiliar but correct responses with high rewards promote *effective exploration*), whereas high-probability actions with high advantage tend to reduce entropy (Cui et al., 2025b). In contrast, R1-GRPO exhibits *entropy collapse*, where policy entropy drops sharply at the start of training and continues to decline monotonically to near zero. R1-Dr.GRPO initially exhibits higher entropy (before step 50) but rapidly drops



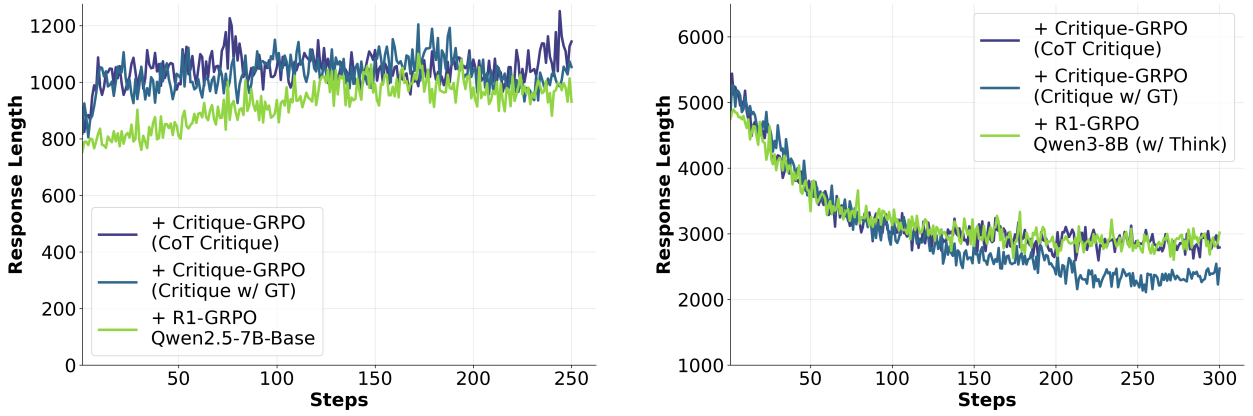
(a) Entropy dynamics for RL-based finetuning approaches over training steps on Qwen2.5-7B-Base.

(b) Entropy dynamics of self-improvement through self-critiquing and weak-to-strong generalization across training steps on Qwen3-8B.

Figure 6: Entropy dynamics during RL finetuning. A comparison of RL-based finetuning approaches for self-improvement on Qwen2.5-7B-Base (left) and self-improvement through self-critiquing with Critique-GRPO on Qwen3-8B (right).

to comparable near-zero values with R1-GRPO after step 150. Combined with the results in Table 2, the superior performance of Critique-GRPO over R1-Dr.GRPO and R1-GRPO highlights the importance of maintaining a certain level of entropy for better performance.

Learning through self-critiquing facilitates policy exploration. Figure 6b shows that Critique-GRPO (self-critique) avoids entropy collapse and maintains higher entropy than R1-GRPO. This finding aligns with the observation that increased exploration improves performance.



(a) Response length changes on Qwen2.5-7B-Base.

(b) Response length changes on Qwen3-8B.

Figure 7: Comparison of response length changes during RL finetuning on Qwen2.5-7B-Base (left) and Qwen3-8B (right).

Higher entropy does not always guarantee effective exploration. Unexpectedly, as shown in Figure 6b, Critique-GRPO (weaker refinement via critique with ground truth), shown in green, achieves higher entropy than Critique-GRPO (self-critique), shown in dark blue, yet performs worse (average pass@1: 65.55% *vs.* 68.13%). This discrepancy may be due to refinements from weaker models causing larger distributional shifts compared to self-refinements, while also being of lower quality. This suggests that the *quality* of exploration signals is more critical than the *extent* of exploration (as reflected solely by entropy).

Critique-GRPO facilitates concise reasoning. In Figure 7, Critique-GRPO achieves superior performance (Table 2) while minimally increasing response length on Qwen2.5-7B-Base (Figure 7a). This efficiency likely stems from its critique mechanism, which enables precise error identification and refinement, reducing the need for verbose reasoning. Additionally, Critique-GRPO tends to reduce response length on Qwen3-8B (Figure 7b). This trend can be attributed to the correction of Qwen3-8B's tendency toward redundant and ineffective self-reflection, as discussed in Section 5.8.

5.7 Impact of Policy Shaping on RL Finetuning

Table 6: Impact of policy shaping on the token-level probability ratios of generated refinements (as introduced in Section 4) during RL finetuning of Qwen2.5-7B-Base.

Method	Policy Shaping	Math (ID)					Science & General (OOD)			Avg.
		MATH 500	Minerva MATH	Olympiad Bench	AMC23	AIME24	Theorem QA	GPQA Diamond	MMLU Pro	
Qwen2.5-7B-Base	-	60.80	20.20	30.40	35.00	13.30	21.60	28.79	46.24	32.04
+ Critique-GRPO (Ours) (CoT Critique)	w/o	77.40	**41.00**	39.70	45.00	16.70	42.60	34.34	54.88	43.95
+ Critique-GRPO (Ours) (CoT Critique)	w/	**77.80**	36.80	**42.40**	**62.50**	**20.00**	**44.00**	**37.88**	**55.28**	**47.08**

To clarify the impact of policy shaping on the generated refinements during RL finetuning, we present the results of removing policy shaping during the RL finetuning of Qwen2.5-7B-Base in Table 6.

Policy shaping enhances learning from refinements during online RL finetuning. Critique-GRPO with policy shaping applied to the token-level probability ratios of generated refinements consistently outperforms the variant without policy shaping across nearly all tasks, improving average pass@1 scores by 3.1%.

5.8 Qualitative Analysis

Fine-Grained Analysis. We conduct a fine-grained analysis of 100 generated responses on the Minerva-MATH dataset across four key dimensions: factuality, conciseness, correctness, and logicality, using the prompt in Appendix E. Figure 8 shows that fine-tuning with Critique-GRPO on Qwen2.5-7B-Base achieves the best performance across all four dimensions as well as in average performance. Additionally, the superior performance of RL fine-tuning with Critique-GRPO over R1-GRPO in terms of factuality and correctness indicates that CoT critiques help the model effectively identify errors and improve valid exploration. The inferior performance of SFT in logicality and conciseness may be attributed to the presence of redundant and sometimes illogical self-reflective reasoning behaviors in expert demonstrations, which could negatively impact user experience.

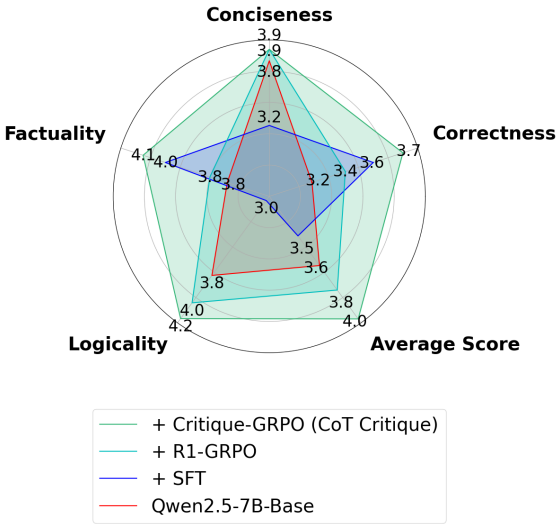


Figure 8: Qualitative analysis of generated responses on Minerva-Math across four dimensions: logicality, factuality, conciseness, correctness, and average performance.

Case Study. We present a qualitative comparison between the responses generated by the base Qwen3-8B model and the RL-finetuned Qwen3-8B model using Critique-GRPO on the MATH-500 dataset. The given question is: `Evaluate` $\sin(\arcsin 0.4 + \arcsin 0.5), \sin(\arcsin 0.5 - \arcsin 0.4)$.

Response Generated by Base Qwen3-8B (Abridged)
Okay, so I need to evaluate this expression: $\sin(\arcsin 0.4 + \arcsin 0.5)$ multiplied by $\sin(\arcsin 0.5 - \arcsin 0.4)$. Hmm, that looks a bit complicated with all those inverse sine functions. Let me try to break it down step by step. First, let me recall some trigonometric identities that might help here. I remember that there are formulas for $\sin(A+B)$ and $\sin(A-B)$. Maybe I can use those? Let me write them down: $\sin(A+B) = \sin A \cos B + \cos A \sin B$, $\sin(A-B) = \sin A \cos B - \cos A \sin B$
So if I let $A = \arcsin 0.4$ and $B = \arcsin 0.5$, then the expression becomes:

$$\sin(A+B) \cdot \sin(A-B)$$

Wait, so maybe I can compute $\sin(A+B)$ and $\sin(A-B)$ separately using those identities and then multiply them together? But before that, maybe I should figure out what A and B are? Well, A is $\arcsin 0.4$, which means that $\sin A = 0.4$, and similarly $\sin B = 0.5$. But I need $\cos A$ and $\cos B$ to use the identities. Since $\sin^2\theta + \cos^2\theta = 1$, I can find $\cos A$ and $\cos B$. Let me compute $\cos A$ first. For angle A, $\sin A = 0.4$, so

$$\cos A = \sqrt{1 - \sin^2 A} = \sqrt{1 - 0.16} = \sqrt{0.84}.$$

Wait, $\sqrt{0.84}$... So $\cos B = \frac{\sqrt{3}}{2}$.
...
Okay, so now I can compute $\sin(A+B)$ and $\sin(A-B)$ using the identities.
...
So

$$\sin(A+B) = 0.2\sqrt{3} + \frac{\sqrt{21}}{10}$$

...
Okay, now moving on to $\sin(A-B)$: $\sin(A-B) = \sin A \cos B - \cos A \sin B$
...
So

$$\sin(A-B) = \frac{2\sqrt{3} - \sqrt{21}}{10}$$

Therefore, the original expression is:

$$\sin(A+B) \cdot \sin(A-B) = \left(\frac{2\sqrt{3} + \sqrt{21}}{10}\right) \cdot \left(\frac{2\sqrt{3} - \sqrt{21}}{10}\right)$$

...
That seems correct. But let me verify with another approach. Maybe using another identity? Alternatively, I remember that $\sin(A+B) \cdot \sin(A-B)$ can be expressed as $\sin^2 A - \sin^2 B$. Wait, is that correct? Wait, let me recall the identity:

$$\sin(A+B) \cdot \sin(A-B) = \sin^2 A - \sin^2 B$$

Let me check that. Yes, because:

$$\sin(A+B) \cdot \sin(A-B) = [\sin A \cos B + \cos A \sin B][\sin A \cos B - \cos A \sin B]$$

$$= (\sin A \cos B)^2 - (\cos A \sin B)^2 = \sin^2 A \cos^2 B - \cos^2 A \sin^2 B$$

...
Let me compute $\sin^2 A - \sin^2 B$:

$$\sin^2 A - \sin^2 B = (0.4)^2 - (0.5)^2 = 0.16 - 0.25 = -0.09 = \frac{-9}{100}$$

...

Response Generated by RL-Finetuned Qwen3-8B using Critique-GRPO (Abridged)
Okay, so I need to evaluate this expression: $\sin(\arcsin 0.4 + \arcsin 0.5)$ multiplied by $\sin(\arcsin 0.5 - \arcsin 0.4)$. Hmm, let me think.

First, I remember that `arcsin` is the inverse sine function, so $\arcsin 0.4$ is an angle whose sine is 0.4, and similarly for $\arcsin 0.5$. Let me denote some variables to make this easier. Let me set $\alpha = \arcsin 0.4$ and $\beta = \arcsin 0.5$. Then the expression becomes $\sin(\alpha + \beta) * \sin(\beta - \alpha)$.

Alright, so I need to compute $\sin(\alpha + \beta)$ and $\sin(\beta - \alpha)$ separately and then multiply them. Maybe I can use sine–addition formulas here. Let me recall that $\sin(A + B) = \sin A \cos B + \cos A \sin B$, and $\sin(A - B) = \sin A \cos B - \cos A \sin B$.

Since α and β are `arcsin` values, I know that $\sin \alpha = 0.4$ and $\sin \beta = 0.5$. Therefore, I need to find $\cos \alpha$ and $\cos \beta$ to use in the formulas.

For $\cos \alpha$, since $\sin^2 \alpha + \cos^2 \alpha = 1$, we have

$$\cos \alpha = \sqrt{1 - \sin^2 \alpha} = \sqrt{1 - 0.16} = \sqrt{0.84}.$$

Note that $0.84 = \frac{84}{100} = \frac{21}{25}$, so $\cos \alpha = \sqrt{\frac{21}{25}} = \frac{\sqrt{21}}{5}$.

Similarly,

$$\cos \beta = \sqrt{1 - \sin^2 \beta} = \sqrt{1 - 0.25} = \sqrt{0.75} = \sqrt{\tfrac{3}{4}} = \frac{\sqrt{3}}{2}.$$

Now, let me compute $\sin(\alpha + \beta)$. Using the formula,

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta = 0.4\left(\frac{\sqrt{3}}{2}\right) + \left(\frac{\sqrt{21}}{5}\right)(0.5).$$

Compute each term:

$$0.4 \cdot \frac{\sqrt{3}}{2} = \frac{2}{5} \cdot \frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{5}, \quad \frac{\sqrt{21}}{5} \cdot 0.5 = \frac{\sqrt{21}}{10}.$$

Therefore

$$\sin(\alpha + \beta) = \frac{\sqrt{3}}{5} + \frac{\sqrt{21}}{10} = \frac{2\sqrt{3} + \sqrt{21}}{10}.$$

Similarly, compute $\sin(\beta - \alpha)$:

$$\sin(\beta - \alpha) = \sin \beta \cos \alpha - \cos \beta \sin \alpha = 0.5\left(\frac{\sqrt{21}}{5}\right) - \frac{\sqrt{3}}{2}(0.4).$$

Thus

$$\sin(\beta - \alpha) = \frac{\sqrt{21}}{10} - \frac{2\sqrt{3}}{10} = \frac{\sqrt{21} - 2\sqrt{3}}{10}.$$

Therefore, the original expression is

$$\left(\frac{2\sqrt{3} + \sqrt{21}}{10}\right)\left(\frac{\sqrt{21} - 2\sqrt{3}}{10}\right).$$

Let $A = 2\sqrt{3} + \sqrt{21}$ and $B = \sqrt{21} - 2\sqrt{3}$; then the product is $\frac{A\,B}{100}$. Note that $A\,B = (\sqrt{21})^2 - (2\sqrt{3})^2 = 21 - 12 = 9$, so the value is $\frac{9}{100}$.
...

The base Qwen3-8B model produces an incorrect response due to an incorrect formulation of the expression as $\sin(A+B)\cdot\sin(A-B)$, as shown in the red square. In contrast, the RL-finetuned Qwen3-8B model using Critique-GRPO generates a correct response by correctly formulating the expression as $\sin(\alpha+\beta)\cdot\sin(\beta-\alpha)$, as shown in the green square. Furthermore, the base Qwen3-8B model exhibits numerous redundant and ineffective self-reflection attempts (*e.g.,* "Wait, ..." highlighted in blue), which fail to help the model identify the actual errors and answer the question correctly. This results in an excessively long response (over 6000 tokens). In contrast, the RL-finetuned Qwen3-8B model using Critique-GRPO demonstrates concise and effective reasoning. It remains on the correct path to solve the problem, exhibits valid self-reflection to validate the answer (*e.g.,* "Let me check with approximate values. ..."), and ultimately generates the correct answer. Detailed responses are provided in Appendix I.

6 Conclusion

In this paper, we first identify three key challenges faced by reinforcement learning relying solely on numerical feedback. We then demonstrate the efficacy of leveraging natural language feedback in the form of critiques to address these challenges. To this end, we propose Critique-GRPO, an online RL framework that enables LLMs to learn from both natural language and numerical feedback simultaneously for effective policy optimization. Specifically, Critique-GRPO facilitates learning from initial responses and critique-guided self-refinements while preserving exploration. Additionally, we employ a shaping function to amplify learning from correct, especially unfamiliar, refinements and penalize incorrect ones. Extensive experiments with Qwen2.5-7B-Base, Qwen2.5-Math-7B-Base, and Qwen3-8B demonstrate that Critique-GRPO consistently achieves superior performance across eight challenging reasoning tasks. Furthermore, Critique-GRPO enables efficient self-improvement through self-critiquing and weak-to-strong generalization. Future work could explore extending Critique-GRPO to multimodal reasoning tasks to strengthen connections between visual understanding and textual reasoning.

Broader Impact Statement

This research adheres to ethical guidelines prioritizing privacy, fairness, and the well-being of individuals and groups. All benchmark datasets used are solely for research purposes and were verified to contain no personally identifiable information, ensuring user privacy. Prompts for data generation were carefully designed to exclude biased or discriminatory language, and all generated data was manually reviewed to confirm the absence of offensive content or personal information. These measures ensure the ethical integrity of our work.

Limitations

While Critique-GRPO establishes a promising foundation for leveraging both natural language and numerical feedback, notable limitations remain.

Performance limitations due to failed refinements. Policy models sometimes fail to follow CoT critiques to refine their responses. We attribute this to the lack of deliberate training for self-refinement. An example of a failed refinement is provided in Appendix J. Future work could focus on improving the model's refinement capabilities or training a specialized model dedicated to refinement tasks.

Critique-GRPO requires longer training time. Critique-GRPO enables models to learn simultaneously from initial responses and self-generated refinements during online policy learning. However, generating these refinements requires an additional inference step, leading to longer training times compared to standard fine-tuning with GRPO.

The role of critique detail in refinement quality. We currently utilize three types of critiques (see Section 3), with CoT critiques demonstrating the greatest benefits for refinement. This advantage likely stems from their detailed step-by-step evaluations and concise improvement suggestions, which help models identify and correct errors in initial responses. It follows that more detailed critiques could result in higher-quality refinements. For simplicity, we use GPT-4o as the reasoning-based reward model, *not for expert knowledge distillation*. Consequently, the generated CoT critiques do not include expert demonstrations. Future work

may explore alternative reasoning-based reward models. One might assume that directly incorporating expert demonstrations into critiques would significantly improve performance. However, our experiments reveal otherwise. Upon analyzing the generated refinements, we observe that both pre-trained models (*e.g.,* Qwen2.5-7B-Base) and alignment-tuned models (*e.g.,* Qwen3-8B) tend to produce conclusive sentences and correct answers as refinements, rather than detailed step-by-step reasoning to derive the correct answer. This behavior limits the effectiveness of expert demonstrations as critiques.

Future work could investigate, in greater depth, which types of critiques provide the most significant benefits for refinement, particularly in reasoning-intensive tasks.

References

Elie Bakouch, Leandro von Werra, and Lewis Tunstall. Open-r1: a fully open reproduction of deepseek-r1. *https://huggingface.co/blog/open-r1*, 2025.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, Ilya Sutskever, and Jeff Wu. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision, 2023. URL https://arxiv.org/abs/2312.09390.

Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomek Korbak, David Lindner, Pedro Freire, Tony Tong Wang, Samuel Marks, Charbel-Raphael Segerie, Micah Carroll, Andi Peng, Phillip J.K. Christoffersen, Mehul Damani, Stewart Slocum, Usman Anwar, Anand Siththaranjan, Max Nadeau, Eric J Michaud, Jacob Pfau, Dmitrii Krasheninnikov, Xin Chen, Lauro Langosco, Peter Hase, Erdem Biyik, Anca Dragan, David Krueger, Dorsa Sadigh, and Dylan Hadfield-Menell. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=bx24KpJ4Eb. Survey Certification, Featured Certification.

Angelica Chen, Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Samuel R. Bowman, Kyunghyun Cho, and Ethan Perez. Learning from natural language feedback. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=xo3hI5MwvU.

Wenhu Chen, Ming Yin, Max Ku, Pan Lu, Yixin Wan, Xueguang Ma, Jianyu Xu, Xinyi Wang, and Tony Xia. TheoremQA: A theorem-driven question answering dataset. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7889–7901, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.489. URL https://aclanthology.org/2023.emnlp-main.489/.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025a. URL https://arxiv.org/abs/2502.01456.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025b. URL https://arxiv.org/abs/2505.22617.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang,

Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. RAFT: reward ranked finetuning for generative foundation model alignment. *Trans. Mach. Learn. Res.*, 2023, 2023. URL https://openreview.net/forum?id=m7p5O7zblY.

Mehdi Fatemi, Banafsheh Rafiee, Mingjie Tang, and Kartik Talamadupula. Concise reasoning via reinforcement learning, 2025. URL https://arxiv.org/abs/2504.05185.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. Learning from dialogue after deployment: Feed yourself, chatbot! In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3667–3684, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1358. URL https://aclanthology.org/P19-1358/.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.211. URL https://aclanthology.org/2024.acl-long.211/.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. *CoRR*, abs/2402.14008, 2024b. doi: 10.48550/ARXIV.2402.14008. URL https://doi.org/10.48550/arXiv.2402.14008.

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. *CoRR*, abs/2103.03874, 2021. URL https://arxiv.org/abs/2103.03874.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. *CoRR*, abs/2310.08491, 2023. doi: 10.48550/ARXIV.2310.08491. URL https://doi.org/10.48550/arXiv.2310.08491.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Gengyang Li, Yifeng Gao, Yuming Li, and Yunfang Wu. Thinkless: A training-free inference-efficient method for reducing reasoning redundancy. *arXiv preprint arXiv:2505.15684*, 2025.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. Numinamath. https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025a. URL https://arxiv.org/abs/2503.20783.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling, 2025b. URL https://arxiv.org/abs/2504.02495.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. Llm critics help catch llm bugs, 2024. URL https://arxiv.org/abs/2407.00215.

OpenAI. Openai o3-mini. https://openai.com/index/openai-o3-mini/, 2025.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin

Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL https://arxiv.org/abs/2412.16720.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *CoRR*, abs/2203.02155, 2022. doi: 10.48550/ARXIV.2203.02155. URL https://doi.org/10.48550/arXiv.2203.02155.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Yaswanth Chittepu, Ryan Park, Harshit Sikchi, Joey Hejna, W. Bradley Knox, Chelsea Finn, and Scott Niekum. Scaling laws for reward model overoptimization in direct alignment algorithms. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/e45caa3d5273d105b8d045e748636957-Abstract-Conference.html.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators, 2022. URL https://arxiv.org/abs/2206.05802.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 2025.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ad236edc564f3e3156e1b2feafb99a24-Abstract-Datasets_and_Benchmarks_Track.html.

Yubo Wang, Xiang Yue, and Wenhu Chen. Critique fine-tuning: Learning to critique is more effective than learning to imitate, 2025. URL https://arxiv.org/abs/2501.17703.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning, 2025. URL https://arxiv.org/abs/2505.10320.

Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8:229–256, 1992.

Zhiheng Xi, Dingwen Yang, Jixuan Huang, Jiafu Tang, Guanyu Li, Yiwen Ding, Wei He, Boyang Hong, Shihan Do, Wenyu Zhan, Xiao Wang, Rui Zheng, Tao Ji, Xiaowei Shi, Yitao Zhai, Rongxiang Weng, Jingang Wang, Xunliang Cai, Tao Gui, Zuxuan Wu, Qi Zhang, Xipeng Qiu, Xuanjing Huang, and Yu-Gang Jiang. Enhancing llm reasoning via critique models with test-time and training-time supervision, 2024. URL https://arxiv.org/abs/2411.16579.

Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance, 2025. URL https://arxiv.org/abs/2504.14945.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

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.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?, 2025. URL https://arxiv.org/abs/2504.13837.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

Xiaoying Zhang, Baolin Peng, Jianfeng Gao, and Helen Meng. Toward self-learning end-to-end task-oriented dialog systems. In Oliver Lemon, Dilek Hakkani-Tur, Junyi Jessy Li, Arash Ashrafzadeh, Daniel Hernández Garcia, Malihe Alikhani, David Vandyke, and Ondřej Dušek (eds.), *Proceedings of the 23rd Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pp. 516–530, Edinburgh, UK, September 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.sigdial-1.49. URL https://aclanthology.org/2022.sigdial-1.49/.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13348–13369, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.891. URL https://aclanthology.org/2023.findings-emnlp.891/.

Xiaoying Zhang, Baolin Peng, Ye Tian, Jingyan Zhou, Lifeng Jin, Linfeng Song, Haitao Mi, and Helen Meng. Self-alignment for factuality: Mitigating hallucinations in LLMs via self-evaluation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1946–1965, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.107. URL https://aclanthology.org/2024.acl-long.107/.

A Appendix

Contents

1 Introduction **1**

2 Related Work **3**

3 Limitations of RL with Numerical Feedback and the Promise of Natural Language Guidance **3**

3.1 Limitations of Learning with Numerical Feedback 3

3.2 Promise of Learning from Natural Language Feedback 6

4 Critique-GRPO **8**

4.1 From GRPO to Critique-GRPO 8

4.2 Online Learning with Critique-GRPO 10

5 Experiments **11**

5.1 Experimental Setup 11

5.2 Main Results 12

5.3 Investigation on Math-Centric Backbone Models 13

5.4 Self-Improvement via Self-Critiquing 14

5.5 Exploration of Weak-to-Strong Generalization 15

5.6 Investigation of Policy Exploration During RL Finetuning 15

5.7 Impact of Policy Shaping on RL Finetuning 17

5.8 Qualitative Analysis 17

6 Conclusion **20**

A Appendix **27**

B Implementation Details **29**

C Analysis of Cognitive Behaviors **30**

D Preliminary Investigation on Qwen3-8B-Base **32**

E Prompts **33**

F Leveraging Textual Critiques for Refining LLM Responses **35**

G The Critique-GRPO Algorithm **36**

H An Example of Successful Refinement using a CoT Critique **37**

I Responses Utilized in Qualitative Analysis **41**

J An Example of Failed Refinement **47**

B Implementation Details

Table 7: Default hyperparameters and training configurations used in Critique-GRPO.

Name	**Value for Qwen2.5-7B-Base or Qwen2.5-Math-7B-Base**	**Value for Qwen3-8B**	**Description**
RL Finetuning			
num_training_prompts	4k	4k	Default number of training prompts (unless specified otherwise).
training_steps	400	300	Total number of training steps.
eval_freq	20	25	Frequency of evaluations (in updates).
batch_size	128	128	Accumulated batch size during training.
lr	$1e^{-6}$	$1e^{-6}$	Learning rate.
max_prompt_length	1024	1024	Maximum input context length.
max_response_length	8192	6144	Maximum length of generated responses.
n_rollouts	7	7	Number of rollouts per prompt.
n_refinements	1	1	Number of refinements per prompt.
rewards	1 or 0	1 or 0	Scalar rewards for responses.
kl_loss_coef	0.0	0.0	Coefficient for KL divergence loss.
γ	0.1	0.1	Hyperparameter in the policy shaping function.
train_temp	1.0	1.0	Sampling temperature during rollouts.
val_temp	0.6	0.6	Sampling temperature during validation.
total_epochs	30	30	Total number of training epochs.
Evaluation			
eval_temp	0.0	0.0	Sampling temperature during evaluation.
max_tokens	8192	8192	Inference token budget during evaluation.

C Analysis of Cognitive Behaviors

To systematically investigate this question, we characterize six key cognitive behaviors that contribute to self-improving reasoning during RL fine-tuning, as follows:

- **Subgoal Setting**: Breaking down complex problems into smaller, manageable steps or subtasks. For example, "Step 1... Step 2..."
- **Summarization**: Summarizing the current state by identifying completed subtasks and determining what remains to be done. This helps guide the next steps in reasoning. For example, "Now we have obtained..., next, we need to..."
- **Verification**: Systematically checking intermediate results or computations to ensure correctness. For example, "Let's verify this result by..."
- **Backtracking**: Identifying errors or dead-ends in reasoning and explicitly revising previous methods or approaches. For example, "This approach won't work because..., let's try another method..."
- **Backward Chaining**: Reasoning from desired outcomes back to initial inputs or steps required to achieve the result. This is particularly applicable to multiple-choice questions where answer options are provided. For example, "To get 24, I could do $24 \div 2 = 12$..." (Gandhi et al., 2025)
- **Anticipation**: Anticipating potential inaccuracies or exhaustively considering multiple possibilities to solve a problem. For example, "Alternatively, this problem can be solved by..."

We analyze the reasoning (cognitive) behaviors using the prompts shown below.

When assessing the contributions of reasoning behaviors in Section 3 to successful problem-solving in RL fine-tuned models, we count each behavior appearing in the generated responses *only once*. For example, if the model produces multiple subgoals in a single response, the occurrence of "subgoal setting" is counted as one.

Prompts for Analyzing Reasoning Behaviors (1/2)

System: You are a helpful assistant.
User: The following is a chain-of-thought produced by a language model in response to a math & science problem:
Question: <Question Content>
Reasoning: <Model Reasoning>
Ground Truth: <Ground Truth Content>
Task 1: Answer Verification
Determine whether the reasoning includes any *explicit or implicit answer verification steps* — moments where the model checks intermediate computations or final results for correctness.
Example: "Let's verify this result by..."
- Report the number of distinct answer verification steps using: `<count>n</count>`. If none are found, return `<count>0</count>`. - If such behavior is present and the final answer matches the ground truth, indicate whether the behavior contributed to the correct answer using the format: `contribution: yes/no`.
Task 2: Backtracking Behavior
Determine whether the reasoning demonstrates *backtracking* — where the model identifies an error or dead end and switches to a different approach.
Example: "This approach won't work because..., let's try another method..."
- Report the number of distinct backtracking instances using: `<count>n</count>`. If none are found, return `<count>0</count>`. - If such behavior is present and the final answer matches the ground truth, indicate whether the behavior contributed to the correct answer using the format: `contribution: yes/no`.

Prompts for Analyzing Reasoning Behaviors (2/2)

Task 3: Subgoal Setting
Determine whether the reasoning includes any *explicit subgoals* — intermediate steps that break the problem into smaller, manageable parts.
Example: "First, I'll try to..., then I'll..."
- Report the number of clearly defined subgoals using: `<count>n</count>`. If none are found, return `<count>0</count>`. - If such behavior is present and the final answer matches the ground truth, indicate whether the behavior contributed to the correct answer using the format: `contribution: yes/no`.

Task 4: Backward Chaining
Determine whether the reasoning includes *backward chaining* — starting from the target result and reasoning backward to infer inputs or steps.
Example: "To get 24, I could do 24 ÷ 2 = 12..."
- Report the number of distinct backward chaining attempts using: `<count>n</count>`. If none are found, return `<count>0</count>`. - If such behavior is present and the final answer matches the ground truth, indicate whether the behavior contributed to the correct answer using the format: `contribution: yes/no`.

Task 5: Anticipation
Determine whether the reasoning includes *enumeration* or *anticipation and re-proposal* — suggesting alternative approaches or revising prior methods.
Examples: "Alternatively, this problem can be solved by...", "Let's try a different approach..."
- Report the number of such instances using: `<count>n</count>`. If none are found, return `<count>0</count>`. - If such behavior is present and the final answer matches the ground truth, indicate whether the behavior contributed to the correct answer using the format: `contribution: yes/no`.

Task 6: Summarization
Determine whether the reasoning includes *summarization* — identifying completed subtasks, summarizing progress, and determining the next steps.
Example: "Now we have obtained..., next, we need to..."
- Report the number of summarization instances using: `<count>n</count>`. If none are found, return `<count>0</count>`. - If such behavior is present and the final answer matches the ground truth, indicate whether the behavior contributed to the correct answer using the format: `contribution: yes/no`.

D Preliminary Investigation on Qwen3-8B-Base

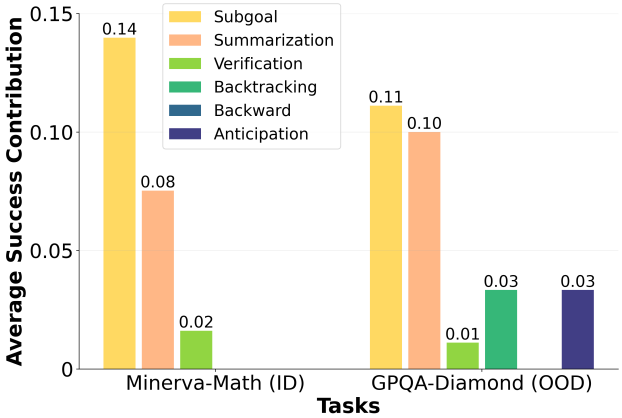


Figure 9: Average contribution of reasoning behaviors to successful completions of previously failed questions by Qwen3-8B-Base on Minerva-Math (in-distribution) and GPQA-Diamond (out-of-distribution) datasets.

We identify 50 previously unsolved problems from the Minerva-Math dataset and 15 from the GPQA-Diamond dataset for Qwen3-8B-Base. Figure 9 shows the average contribution of reasoning behaviors to successful completions of previously failed questions by Qwen3-8B-Base. Notably, self-reflection behaviors contribute minimally to successful problem-solving.

Table 8: Analysis of performance gains from critique-based self-refinement on Qwen3-8B-Base.

Method	% Failed Questions (Pass@4=0)	Critique Type	% Valid Critiques	% Valid Refinements	% Critiqued Questions	% Questions Refined
RL-finetuned Qwen3-8B-Base	17.18	Indicative Critique	100.00	3.57	100.00	11.21
		Indicative Critique w/ GT	100.00	3.93	100.00	12.23
		CoT Critique	66.08	**44.71**	98.25	**66.96**

Table 8 reveals that the best-performing RL-finetuned Qwen3-8B-Base persistently failed on 17.18% of training problems. In addition, all three types of critiques facilitate the LLM's self-refinements. These findings are consistent with the observations in Section 3.

E Prompts

Training Prompt. The following training prompt is used during all RL fine-tuning experiments:

Training Prompt

System: You are a helpful assistant.
User: `<Question Content>`
Please reason step by step and place your final answer within `\boxed`.

Prompt for Generating Chain-of-Thought Critique. We adopt a prompt inspired by (Wang et al., 2025) to enable GPT-4o (Hurst et al., 2024) to generate CoT critiques. For quality control, we retained only those model-generated critiques whose evaluative conclusions (correct/incorrect) aligned with rule-based verification. When inconsistencies occurred, we prompted the critique model to regenerate the critiques.

Prompt for Generating Chain-of-Thought Critique

System: You are a science expert. A student is trying to solve a question. Please explain briefly (step-by-step) whether the student's solution is correct or not. Finally, conclude your judgment with: "Conclusion: `correct/incorrect` [END]."
User:
Question: `<Question Content>`
Ground Truth Answer: `<Ground Truth>`
Student's Solution: `<Generated Solution>`
Critique:

Prompt for Generating Chain-of-Thought Critique with Internal Knowledge The following prompt is designed to enable an LLM to leverage its internal knowledge and evaluate the correctness of its own generated responses through step-by-step CoT critiques.

Prompt for Generating Chain-of-Thought Critique with Internal Knowledge

System: You are a science expert. A student is trying to solve a question. Please explain briefly (step-by-step) whether the student's solution is correct or not. Finally, conclude your judgment with: "Conclusion: `correct/incorrect` [END]."
User:
Question: `<Question Content>`
Student's Solution: `<Generated Solution>`
Critique:

Refinement Prompt. The following refinement prompt is used to guide the model in improving its response by incorporating the critique.

Refinement Prompt

System: You are a helpful assistant.
User: Given the following inputs:
Question: `<Question Content>`
Previous Solution: `<Generated Solution>`
Critique: `<Critique Content>`
Please re-answer by:
- Correcting potential errors identified in the critique, if they exist.
- Providing clear, step-by-step reasoning.
- Placing your final answer within `\boxed`.
Ensure the revised solution addresses all issues raised in the critique.

Future work could explore designing prompts (Zhang et al., 2023) to enable LLMs to generate high-quality CoT critiques.

Prompt for Qualitative Analysis. We employ the following prompt to conduct qualitative analysis of the generated responses using GPT-4o.

Prompt for Qualitative Analysis

System: You are a science expert. You are provided with a question, the correct ground truth answer, and a student's solution. Please conduct a fine-grained qualitative analysis of the student's solution based on the following four aspects, rated on a scale of 1-5:
1. Correctness of the Final Answer: Is the final numerical answer correct based on your calculations or the ground truth? If incorrect, provide the correct answer and explain the discrepancy.
2. Verbosity: Is the reasoning path too verbose, too concise, or appropriate? Identify areas for condensation or expansion to improve clarity.
3. Factual Accuracy: Are all formulas, conversions, and physical principles factually accurate? Highlight any errors or misleading statements.
4. Logical Coherence: Does the reasoning flow logically from one step to the next? Identify gaps in logic, missing steps, or irrelevant details that detract from the solution.
End your analysis with:
"Conclusion:" Provide ratings (1-5 scale) for each aspect.
User:
Question: `<Question Content>`
Ground Truth Answer: `<Ground Truth>`
Student's Answer: `<Answer>`
Conclusion:

F Leveraging Textual Critiques for Refining LLM Responses

We describe the process for leveraging these textual critiques to guide the refinement of LLM-generated responses:

1. **Initial Response Sampling**: Given an LLM π_θ parameterized by θ and a set of questions $\{q\}$, we sample multiple initial responses for each question $\{y_0^{(i)}\}_{i=1}^k \sim \pi_\theta(\cdot \mid q)$, where k is the number of samples.

2. **Response Evaluation and Critique Generation**: We use an evaluation function $\text{Eval}(q, y_0)$ to assess the correctness of each response y_0. The function outputs 1 if y_0 is correct and 0 otherwise. Specifically, we adopt a model-based evaluation with a reasoning-based reward model π_{RM}. The reasoning-based reward model generates a CoT critique $c_{\text{CoT}}^{(i)} \sim \pi_{RM}(\cdot \mid I_c, q, y_0^{(i)})$, where I_c is a predefined instruction (detailed in Appendix E). Based on the binary correctness label within $c_{\text{CoT}}^{(i)}$, we construct the corresponding heuristic-based critiques: an indicative critique $c_{\text{I}}^{(i)}$ (containing only the correctness label) and a critique with ground truth $c_{\text{GT}}^{(i)}$ (correctness label plus the known ground truth answer for q).

To focus on the model's ability to learn from critiques for initially incorrect solutions and to control for spontaneous self-correction, we identify persistently failed questions. A question q is classified as persistently failed if all k of its initial responses $\{y_0^{(i)}\}_{i=1}^k$ are deemed incorrect based on the labels from their respective CoT critiques. For each such incorrect response $y_0^{(j)}$ from a persistently failed question, we form a triplet $(q, y_0^{(j)}, c^{(j)})$, where $c^{(j)}$ is one of the three critique types: $c_{\text{CoT}}^{(j)}$, $c_{\text{GT}}^{(j)}$, or $c_{\text{I}}^{(j)}$.

3. **Self-Refinement Generation**: For each selected triplet $(q, y_0^{(j)}, c^{(j)})$ corresponding to an initial incorrect response, we prompt the original LLM π_θ to generate a refined response $y_{\text{refined}}^{(j)} \sim \pi_\theta(\cdot \mid I_{\text{refine}}, q, y_0^{(j)}, c^{(j)})$. This generation is conditioned on a specific refinement instruction I_{refine} (detailed in Appendix E), the original question q, the initial failed response $y_0^{(j)}$, and its associated critique $c^{(j)}$.

The full process is summarized in Algorithm 1. An example illustrating the self-refinement process, including the application of a CoT critique, is provided in Appendix H.

G The Critique-GRPO Algorithm

The Critique-GRPO algorithm is summarized in Algorithm 2.

Algorithm 2 Critique-GRPO: Online Policy Optimization Framework with Critiques

1: **Input:** Pretrained LLM policy π_{old} parameterized by θ, reward model π_{RM}, set of questions $Q = \{q\}$, refinement instruction I_{refine}, critique instruction I_c
2: **Goal:** Improve LLM policy by learning from initial responses and their refinements
3: **Step 1: Initial Response Sampling**
4: **for** each question $q \in Q$ **do**
5: Sample n initial responses from the old policy: $\{y^{(i)}\}_{i=1}^{n} \sim \pi_{\text{old}}(\cdot|q)$
6: Score the responses using the reward model π_{RM} to obtain CoT critiques:

$$\{c_{\text{CoT}}^{(i)}\}_{i=1}^{n} \sim \pi_{RM}(\cdot \mid I_c, q, y^{(i)})$$

7: Translate binary correctness labels in the critiques into scalar reward scores: $\{R^{(i)}\}_{i=1}^{n}$
8: **end for**
9: **Step 2: Critique-Guided Self-Refinement**
10: **for** each initial response $y^{(i)} \in \{y^{(i)}\}_{i=1}^{n}$ **do**
11: Generate refined responses conditioned on $(q, y^{(i)}, c_{\text{CoT}}^{(i)})$ and instruction I_{refine}:

$$y_{\text{refined}}^{(i)} \sim \pi_{\text{old}}(\cdot \mid I_{\text{refine}}, q, y^{(i)}, c_{\text{CoT}}^{(i)})$$

12: Score the refined responses using the reward model (or rule-based evaluation function):

$$\{R_{\text{refine}}^{(i)}\}_{i=1}^{n}$$

13: **end for**
14: Sample a subset of k refinements to mitigate distributional shifts: $\{y_{\text{refined}}^{(i')}\}_{i'=1}^{k} \subset \{y_{\text{refined}}^{(i)}\}_{i=1}^{n}$
15: Combine the sampled refinements with the initial responses to form a mixed group:

$$\{y^{(i)}\}_{i=1}^{n} \cup \{y_{\text{refined}}^{(i')}\}_{i'=1}^{k}$$

16: **Step 3: Online Policy Optimization**
17: Fine-tune the model on the mixed group of responses using scalar rewards with the Critique-GRPO training objective:

$$\begin{aligned}\mathcal{J}_{\text{Critique-GRPO}}(\theta) =& \mathbb{E}_{q\sim Q, \{y^{(i)}\}_{i=1}^{n}\sim\pi_{\text{old}}(\cdot|q), \{y_{\text{refined}}^{(i')}\}_{i'=1}^{k}\sim\pi_{\text{old}}(\cdot|I_{\text{refine}}, q, y^{(i')}, c_{\text{CoT}}^{(i')})} \\ & \Bigg[\underbrace{\frac{1}{n}\sum_{i=1}^{n}\sum_{t=1}^{|y^{(i)}|} r_t^{(i)}(\theta) A_t^{(i)}}_{\text{Objective for Initial Responses}} + \underbrace{\frac{1}{k}\sum_{i'=1}^{k}\sum_{t=1}^{|y_{\text{refined}}^{(i')}|} f\left(r_{\text{refined},t}^{(i')}(\theta)\right) A_t^{(i')}}_{\text{Objective for Refined Responses}} \Bigg],\end{aligned} \tag{7}$$

18: Compute the advantage for each token in a response:

$$A_t^{(i)/(i')} = R^{(i)/(i')} - \text{mean}(\{R^{(i)}\}_{i=1}^{n} \cup \{R_{\text{refined}}^{(i')}\}_{i'=1}^{k})$$

19: Calculate the probability ratios for initial and refined responses:

$$r_t^{(i)}(\theta) = \frac{\pi_\theta(y_t^{(i)}|q, y_{<t}^{(i)})}{\pi_{\text{old}}(y_t^{(i)}|q, y_{<t}^{(i)})}, \quad f(r_{\text{refined},t}^{(i')}(\theta)) = \frac{\pi_\theta(y_{\text{refined},t}^{(i')}|q, y_{\text{refined},<t}^{(i')})}{\pi_\theta(y_{\text{refined},t}^{(i')}|q, y_{\text{refined},<t}^{(i')}) + \gamma}$$

20: **Output:** Fine-tuned LLM policy π_θ

H An Example of Successful Refinement using a CoT Critique

The example below demonstrates a successful refinement using a CoT critique. This output was generated by the best-performing RL-finetuned Qwen3-8B model using GRPO algorithm in Section 3, utilizing the refinement prompt detailed in Appendix H.

Question
2. A sphere is inscribed in a cone, such that the area of the sphere and the area of the base of the cone are equal. Calculate the cosine of the angle of the axial section of the cone at the vertex which is also the apex of the cone.

CoT Critique
Let's analyze the student's solution step-by-step and identify any errors:
Step 1: Geometry Understanding The student correctly identifies the geometry of the problem, noting that the sphere is tangent to the base and sides of the cone, and that the axial section forms an isosceles triangle.
Step 2: Variable Definitions The student defines r as the radius of the sphere, R as the radius of the base of the cone, and h as the height of the cone. These definitions are appropriate.
Step 3: Relating Sphere and Cone The student attempts to relate r, R, and h using similar triangles. However, the equation derived:

$$r = \frac{Rh}{h + R}$$

is not directly applicable to the problem as it does not consider the specific condition that the areas are equal.
Step 4: Equating Areas The student correctly equates the areas of the sphere and the base of the cone:

$$4\pi r^2 = \pi R^2 \implies R = 2r$$

This step is correct.
Step 5: Substitution and Error The student substitutes $R = 2r$ into the earlier equation, leading to:

$$r = \frac{(2r)h}{h + 2r}$$

This substitution is unnecessary and results in incorrect simplification. The correct approach would involve using the inradius formula for the triangle formed by the axial section of the cone.
Step 6: Angle Calculation The student calculates $\tan\theta = 1$ and concludes $\theta = 45°$, leading to $\cos(90°) = 0$. This is incorrect because the angle calculation should be based on the relationship between h and R, derived from the inradius formula.
Step 7: Precise Calculation The student attempts to verify the angle calculation but repeats the same incorrect logic, leading to the same incorrect conclusion.
Correct Solution Approach The correct approach involves using the inradius formula for the triangle formed by the axial section of the cone: - Calculate the inradius using the area and semi-perimeter of the triangle. - Substitute $R = 2r$ and solve for h in terms of r. - Relate h and θ using trigonometric identities. - Calculate $\cos(2\theta)$ using the double angle identity.
The correct final answer is:

$$\cos(2\theta) = \frac{7}{25}.$$

Conclusion: incorrect [END]

Incorrect Initial Response

To solve the problem, let's break it down step by step.

Step 1: Geometry Understanding - **Sphere Inscribed in a Cone**: The sphere is tangent to the base and the sides of the cone. - **Axial Section**: The cross-section along the cone's central axis forms an isosceles triangle. - **Vertex Angle**: Let the vertex angle of the cone be 2θ, so each base angle is θ.

Step 2: Variable Definitions - r: Radius of the sphere. - R: Radius of the base of the cone. - h: Height of the cone.

Step 3: Relating the Sphere and the Cone The sphere is inscribed in the cone, so: - The vertical distance from the center of the sphere to the base is $h - r$. - The horizontal distance from the center of the sphere to the cone's side is r.

Using similar triangles, we relate r, R, and h:

$$\frac{r}{R} = \frac{h-r}{h} \implies rh = R(h-r) \implies r(h+R) = Rh \implies r = \frac{Rh}{h+R}.$$

Step 4: Equating Areas The area of the sphere is $4\pi r^2$, and the area of the base of the cone is πR^2. From the problem:

$$4\pi r^2 = \pi R^2 \implies 4r^2 = R^2 \implies R = 2r.$$

Step 5: Substituting $R = 2r$ Substituting $R = 2r$ into the earlier equation:

$$r = \frac{(2r)h}{h+2r} \implies r(h+2r) = 2rh \implies h + 2r = 2h \implies h = 2r.$$

Step 6: Finding the Angle θ In the axial section of the cone, we have a right triangle with: - Height $h = 2r$. - Base radius $R = 2r$.

The tangent of θ is:

$$\tan\theta = \frac{R}{h} = \frac{2r}{2r} = 1 \implies \theta = 45^\circ.$$

The cosine of the vertex angle 2θ is:

$$\cos(2\theta) = \cos(90^\circ) = 0.$$

Step 7: Verifying with Precise Calculations For the axial section, the cone forms an isosceles triangle with: - Two equal slant heights l. - Base $2R = 4r$. - Height $h = 2r$.

Using the Pythagorean theorem, the slant height l is:

$$l = \sqrt{h^2 + R^2} = \sqrt{(2r)^2 + (2r)^2} = \sqrt{8r^2} = 2r\sqrt{2}.$$

The tangent of the base angle θ is:

$$\tan\theta = \frac{R}{h} = \frac{2r}{2r} = 1 \implies \theta = 45^\circ.$$

Thus, the vertex angle is $2\theta = 90^\circ$, and:

$$\cos(90^\circ) = 0.$$

Final Answer

$$\boxed{0}$$

Correct Refined Response (1/2)
Let's revise the solution by addressing the issues identified in the critique and providing a clear, step-by-step reasoning.
Step 1: Geometry Understanding The problem involves a cone with a sphere inscribed in it, such that the sphere is tangent to both the base and the sides of the cone. Additionally, the area of the sphere equals the area of the base of the cone.
Step 2: Variable Definitions - r: Radius of the sphere. - R: Radius of the base of the cone. - h: Height of the cone.
Step 3: Relating the Sphere and the Cone The sphere is inscribed in the cone, so: - The vertical distance from the center of the sphere to the base is $h - r$. - The horizontal distance from the center of the sphere to the cone's side is r.
Using similar triangles, we relate r, R, and h:

$$\frac{r}{R} = \frac{h-r}{h} \implies r(h+R) = Rh \implies r = \frac{Rh}{h+R}.$$

Step 4: Equating Areas The area of the sphere is $4\pi r^2$, and the area of the base of the cone is πR^2. From the problem:

$$4\pi r^2 = \pi R^2 \implies 4r^2 = R^2 \implies R = 2r.$$

Step 5: Substituting $R = 2r$ Substitute $R = 2r$ into the earlier equation:

$$r = \frac{(2r)h}{h+2r} \implies r(h+2r) = 2rh \implies h + 2r = 2h \implies h = 2r.$$

Step 6: Finding the Angle θ In the axial section of the cone: - The height is $h = 2r$. - The base radius is $R = 2r$.
The tangent of θ is:

$$\tan\theta = \frac{R}{h} = \frac{2r}{2r} = 1 \implies \theta = 45^\circ.$$

The cosine of the vertex angle 2θ is:

$$\cos(2\theta) = \cos(90^\circ) = 0.$$

However, this leads to an incorrect conclusion. Let's apply the **inradius formula** for a more accurate approach.
Correct Approach: Using the Inradius Formula The axial section of the cone is an isosceles triangle with: - Height h. - Base $2R$. - Slant height $l = \sqrt{h^2 + R^2}$.
The inradius r of this triangle is given by:

$$r = \frac{A}{s},$$

where A is the area, and s is the semi-perimeter.
Area and Semi-Perimeter: The area A of the triangle is:

$$A = \frac{1}{2} \times 2R \times h = Rh.$$

Correct Refined Response (2/2)

The semi-perimeter s is:

$$s = \frac{2l + 2R}{2} = l + R = \sqrt{h^2 + R^2} + R.$$

Inradius: Substitute A and s:

$$r = \frac{Rh}{\sqrt{h^2 + R^2} + R}.$$

Given $R = 2r$ and $h = 2r$, substitute into the equation:

$$r = \frac{(2r)(2r)}{\sqrt{(2r)^2 + (2r)^2} + 2r} = \frac{4r^2}{\sqrt{8r^2} + 2r} = \frac{4r^2}{2r\sqrt{2} + 2r}.$$

Simplify:

$$r = \frac{4r}{2(\sqrt{2} + 1)} = \frac{2r}{\sqrt{2} + 1}.$$

Rationalizing the Denominator:

$$r = \frac{2r}{\sqrt{2} + 1} \cdot \frac{\sqrt{2} - 1}{\sqrt{2} - 1} = \frac{2r(\sqrt{2} - 1)}{2 - 1} = 2r(\sqrt{2} - 1).$$

Step 7: Find the Final Angle Given $h = 2r$ and $R = 2r$, the slant height l is:

$$l = \sqrt{h^2 + R^2} = \sqrt{(2r)^2 + (2r)^2} = \sqrt{8r^2} = 2r\sqrt{2}.$$

The tangent of θ is:

$$\tan\theta = \frac{R}{h} = \frac{2r}{2r} = 1 \implies \theta = 45^\circ.$$

The cosine of the vertex angle 2θ is:

$$\cos(2\theta) = \frac{1 - \tan^2\theta}{1 + \tan^2\theta} = \frac{1 - \left(\frac{3}{4}\right)^2}{1 + \left(\frac{3}{4}\right)^2} = \frac{\frac{7}{16}}{\frac{25}{16}} = \frac{7}{25}.$$

Final Answer:

$$\boxed{\frac{7}{25}}$$

I Responses Utilized in Qualitative Analysis

Question
Evaluate

$$\sin(\arcsin 0.4 + \arcsin 0.5) \cdot \sin(\arcsin 0.5 - \arcsin 0.4).$$

Response Generated by Qwen3-8B (1/4)
Okay, so I need to evaluate this expression: $\sin(\arcsin 0.4 + \arcsin 0.5)$ multiplied by $\sin(\arcsin 0.5 - \arcsin 0.4)$. Hmm, that looks a bit complicated with all those inverse sine functions. Let me try to break it down step by step. First, let me recall some trigonometric identities that might help here. I remember that there are formulas for $\sin(A+B)$ and $\sin(A-B)$. Maybe I can use those? Let me write them down:

$$\sin(A+B) = \sin A \cos B + \cos A \sin B$$

$$\sin(A-B) = \sin A \cos B - \cos A \sin B$$

So if I let $A = \arcsin 0.4$ and $B = \arcsin 0.5$, then the expression becomes:

$$\sin(A+B) \cdot \sin(A-B)$$

Wait, so maybe I can compute $\sin(A+B)$ and $\sin(A-B)$ separately using those identities and then multiply them together? But before that, maybe I should figure out what A and B are? Well, A is $\arcsin 0.4$, which means that $\sin A = 0.4$, and similarly $\sin B = 0.5$. But I need $\cos A$ and $\cos B$ to use the identities. Since $\sin^2\theta + \cos^2\theta = 1$, I can find $\cos A$ and $\cos B$. Let me compute $\cos A$ first. For angle A, $\sin A = 0.4$, so

$$\cos A = \sqrt{1 - \sin^2 A} = \sqrt{1 - 0.16} = \sqrt{0.84}.$$

Wait, $\sqrt{0.84}$... Let me calculate that. 0.84 is $\frac{84}{100}$, so $\sqrt{0.84} = \frac{\sqrt{84}}{10}$. Simplify $\sqrt{84}$... $\sqrt{84} = \sqrt{4 \cdot 21} = 2 \cdot \sqrt{21}$. So

$$\sqrt{0.84} = \frac{\sqrt{84}}{10} = \frac{2 \cdot \sqrt{21}}{10} = \frac{\sqrt{21}}{5}.$$

Wait, let me check that again. $\sqrt{84}$ is $\sqrt{4 \cdot 21}$ which is $2 \cdot \sqrt{21}$, so yes, $\sqrt{84}/10 = 2 \cdot \sqrt{21}/10 = \sqrt{21}/5$. So $\cos A = \sqrt{21}/5$. Wait, but $\sqrt{21}$ is approximately 4.583, so $\sqrt{21}/5$ is about 0.9166. Let me check if that's correct. Since $\sin A$ is 0.4, then $\cos A$ should be $\sqrt{1 - 0.16} = \sqrt{0.84} \approx 0.9165$, which matches. So that's correct. Similarly, for angle B, $\sin B = 0.5$, so $\cos B = \sqrt{1 - 0.25} = \sqrt{0.75} = \frac{\sqrt{3}}{2} \approx 0.8660$. That's right because $\sin\left(\frac{\pi}{6}\right) = 0.5$, so $\cos\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2}$. So $\cos B = \frac{\sqrt{3}}{2}$. Okay, so now I can compute $\sin(A+B)$ and $\sin(A-B)$ using the identities. Starting with $\sin(A+B)$:

$$\sin(A+B) = \sin A \cos B + \cos A \sin B$$

Plugging in the values:

$$\sin A = 0.4, \quad \cos B = \frac{\sqrt{3}}{2}, \quad \cos A = \frac{\sqrt{21}}{5}, \quad \sin B = 0.5$$

So:

$$\sin(A+B) = 0.4\left(\frac{\sqrt{3}}{2}\right) + \left(\frac{\sqrt{21}}{5}\right)(0.5)$$

Response Generated by Qwen3-8B (2/4)
Let me compute each term: First term:

$$0.4\left(\frac{\sqrt{3}}{2}\right) = \left(\frac{0.4}{2}\right)\sqrt{3} = 0.2\sqrt{3}$$

Second term:

$$\left(\frac{\sqrt{21}}{5}\right)(0.5) = \left(\frac{\sqrt{21}}{5}\right)\left(\frac{1}{2}\right) = \frac{\sqrt{21}}{10}$$

So

$$\sin(A+B) = 0.2\sqrt{3} + \frac{\sqrt{21}}{10}$$

Hmm, maybe I can write 0.2 as $\frac{1}{5}$? Let me check: 0.2 is $\frac{1}{5}$, so

$$\frac{1}{5}\sqrt{3} + \frac{\sqrt{21}}{10}.$$

To combine these terms, maybe get a common denominator? Yes, common denominator is 10:

$$\frac{2\sqrt{3}}{10} + \frac{\sqrt{21}}{10} = \frac{2\sqrt{3}+\sqrt{21}}{10}$$

So

$$\sin(A+B) = \frac{2\sqrt{3}+\sqrt{21}}{10}$$

Okay, now moving on to $\sin(A-B)$:

$$\sin(A-B) = \sin A\cos B - \cos A\sin B$$

Which is similar to the previous one, but with a minus sign. So plugging in the same values:

$$\sin A\cos B - \cos A\sin B = 0.4\left(\frac{\sqrt{3}}{2}\right) - \left(\frac{\sqrt{21}}{5}\right)(0.5)$$

Which is the same as: First term:

$$0.2\sqrt{3} - \frac{\sqrt{21}}{10}$$

Again, converting 0.2 to $\frac{1}{5}$:

$$\frac{1}{5}\sqrt{3} - \frac{\sqrt{21}}{10} = \frac{2\sqrt{3}-\sqrt{21}}{10}$$

So

$$\sin(A-B) = \frac{2\sqrt{3}-\sqrt{21}}{10}$$

Therefore, the original expression is:

$$\sin(A+B)\cdot\sin(A-B) = \left(\frac{2\sqrt{3}+\sqrt{21}}{10}\right)\cdot\left(\frac{2\sqrt{3}-\sqrt{21}}{10}\right)$$

Response Generated by Qwen3-8B (3/4)

Hmm, this looks like a difference of squares. Because $(a+b)(a-b) = a^2 - b^2$. Let me check: Let me denote $a = 2\sqrt{3}$ and $b = \sqrt{21}$. Then the product is $(a+b)(a-b) = a^2 - b^2$. So compute $a^2 - b^2$:

$$a^2 = (2\sqrt{3})^2 = 4 \cdot 3 = 12$$

$$b^2 = (\sqrt{21})^2 = 21$$

$$\text{Therefore, } a^2 - b^2 = 12 - 21 = -9$$

$$\text{Therefore, the product is } \frac{-9}{10 \cdot 10} = \frac{-9}{100}$$

Wait, so the entire expression simplifies to $\frac{-9}{100}$? But let me check my steps again to make sure I didn't make a mistake. First, $\sin(A+B)$ and $\sin(A-B)$ were calculated as $\frac{2\sqrt{3}+\sqrt{21}}{10}$ and $\frac{2\sqrt{3}-\sqrt{21}}{10}$. Multiplying them gives:

$$\frac{(2\sqrt{3})^2 - (\sqrt{21})^2}{100} = \frac{12-21}{100} = \frac{-9}{100}$$

That seems correct. But let me verify with another approach. Maybe using another identity? Alternatively, I remember that $\sin(A+B) \cdot \sin(A-B)$ can be expressed as $\sin^2 A - \sin^2 B$. Wait, is that correct? Wait, let me recall the identity:

$$\sin(A+B) \cdot \sin(A-B) = \sin^2 A - \sin^2 B$$

Let me check that. Yes, because:

$$\sin(A+B) \cdot \sin(A-B) = [\sin A \cos B + \cos A \sin B][\sin A \cos B - \cos A \sin B]$$

$$= (\sin A \cos B)^2 - (\cos A \sin B)^2 = \sin^2 A \cos^2 B - \cos^2 A \sin^2 B$$

But that's not exactly $\sin^2 A - \sin^2 B$. Wait, unless there's another identity. Alternatively, maybe using product-to-sum formulas? Alternatively, let me compute $\sin^2 A - \sin^2 B$. Let me see:

$$\sin^2 A - \sin^2 B = (\sin A - \sin B)(\sin A + \sin B). \text{ Not sure if that helps.}$$

But let me check if $\sin(A+B) \cdot \sin(A-B)$ equals $\sin^2 A - \sin^2 B$. Let me compute $\sin^2 A - \sin^2 B$:

$$\sin^2 A - \sin^2 B = (0.4)^2 - (0.5)^2 = 0.16 - 0.25 = -0.09 = \frac{-9}{100}$$

Which is exactly the same as before! So that's a shortcut. Therefore:

$$\sin(A+B) \cdot \sin(A-B) = \sin^2 A - \sin^2 B = \frac{-9}{100}$$

Therefore, the answer is $\frac{-9}{100}$. So that confirms the previous result. Therefore, the value of the original expression is $\frac{-9}{100}$. But let me check once more with actual numbers to make sure. Let me compute $\arcsin(0.4)$ and $\arcsin(0.5)$ numerically, then compute the sines of their sum and difference, multiply them, and see if it's approximately -0.09. First, $\arcsin(0.5)$ is $\pi/6 \approx 0.5236$ radians. $\arcsin(0.4)$... Let me calculate that. Let me recall that $\sin(0.4115\,\text{radians}) \approx 0.4$. Let me check with calculator: Well, $\sin(0.4115) \approx \sin(0.4115) \approx 0.4$.

Let me check with calculator steps: But since I don't have a calculator here, maybe approximate. Alternatively, use known values. For example, $\sin(\pi/6) = 0.5$, $\sin(\pi/12) \approx 0.2588$, so 0.4 is between $\pi/12$ and $\pi/6$. Let me approximate $\arcsin(0.4)$.

Response Generated by Qwen3-8B (4/4)

Let me use Taylor series or something? Maybe not. Alternatively, just take that $\arcsin(0.4) \approx 0.4115$ radians (as I thought before). Let me take that as approximate value. So $A \approx 0.4115$ radians, $B = \pi/6 \approx 0.5236$ radians. Compute $A + B \approx 0.4115 + 0.5236 \approx 0.9351$ radians. Compute $\sin(0.9351)$. Let me recall that $\sin(\pi/3) \approx 0.8660$, $\pi/3 \approx 1.0472$ radians. So 0.9351 is less than $\pi/3$. Let me compute $\sin(0.9351)$. Let me use calculator-like approximation.

Alternatively, use the value we found earlier: $\frac{2\sqrt{3}+\sqrt{21}}{10}$. Let me compute that numerically. First, $\sqrt{3} \approx 1.732$, $\sqrt{21} \approx 4.5837$. So $2\sqrt{3} \approx 3.464$, $\sqrt{21} \approx 4.5837$. Then $2\sqrt{3} + \sqrt{21} \approx 3.464 + 4.5837 \approx 8.0477$. Divide by 10: ≈ 0.80477. So $\sin(A + B) \approx 0.8048$. Similarly, $\sin(A - B) = \frac{2\sqrt{3}-\sqrt{21}}{10} \approx \frac{3.464-4.5837}{10} \approx \frac{-1.1197}{10} \approx -0.11197$. Therefore, multiplying them: $0.8048 \cdot (-0.11197) \approx -0.090$. Which is -0.09, which is $-\frac{9}{100}$. So that matches. Therefore, the exact value is $-\frac{9}{100}$, and the approximate decimal is -0.09. So that seems correct. Therefore, the answer is $-\frac{9}{100}$.

But let me check once more if there's any mistake in the sign. Wait, when I calculated $\sin(A - B)$, I had $\frac{2\sqrt{3}-\sqrt{21}}{10}$. Since $\sqrt{21} \approx 4.583$ and $2\sqrt{3} \approx 3.464$, so $2\sqrt{3} - \sqrt{21}$ is negative, so $\sin(A - B)$ is negative. Then $\sin(A + B)$ is positive, so multiplying them gives negative. Which matches with $-\frac{9}{100}$. So that's correct. Alternatively, if I think about angles: A is $\arcsin(0.4)$, which is in the first quadrant, B is $\arcsin(0.5)$ which is $\pi/6$, also first quadrant. So $A + B$ is the sum of two first quadrant angles. Depending on their measures, $A + B$ could be in the first or second quadrant. Similarly, $A - B$ could be positive or negative. Let me check if $A - B$ is positive or negative. Since $\arcsin(0.4) \approx 0.4115$ radians, $\arcsin(0.5) \approx 0.5236$ radians. So $A - B \approx 0.4115 - 0.5236 \approx -0.1121$ radians. So a negative angle, which is equivalent to a positive angle by adding 2π, but sine is an odd function, so $\sin(-x) = -\sin x$. Therefore, $\sin(A - B) = -\sin(B - A)$. So the negative value makes sense. Therefore, multiplying $\sin(A + B)$ which is positive with $\sin(A - B)$ which is negative gives a negative result. So $-\frac{9}{100}$ is correct. Therefore, I think the answer is $-\frac{9}{100}$. Let me check once more with another identity.

Alternatively, using complex numbers or other methods? Maybe not necessary. Since two different approaches gave me the same answer, and numerical approximation also matches, I think it's safe. But just to make sure, let me compute $\sin^2 A - \sin^2 B$:

$$\sin^2 A = (0.4)^2 = 0.16$$

Response Generated by RL-Finetuned Qwen3-8B using Critique-GRPO (1/2)
Okay, so I need to evaluate this expression: $\sin(\arcsin 0.4 + \arcsin 0.5)$ multiplied by $\sin(\arcsin 0.5 - \arcsin 0.4)$. Hmm, let me think.

First, I remember that `arcsin` is the inverse sine function, so $\arcsin 0.4$ is an angle whose sine is 0.4, and similarly for $\arcsin 0.5$. Let me denote some variables to make this easier. Let me set $\alpha = \arcsin 0.4$ and $\beta = \arcsin 0.5$. Then the expression becomes $\sin(\alpha + \beta) * \sin(\beta - \alpha)$.

Alright, so I need to compute $\sin(\alpha + \beta)$ and $\sin(\beta - \alpha)$ separately and then multiply them. Maybe I can use sine–addition formulas here. Let me recall that $\sin(A + B) = \sin A \cos B + \cos A \sin B$, and $\sin(A - B) = \sin A \cos B - \cos A \sin B$.

Since α and β are `arcsin` values, I know that $\sin \alpha = 0.4$ and $\sin \beta = 0.5$. Therefore, I need to find $\cos \alpha$ and $\cos \beta$ to use in the formulas.

For $\cos \alpha$, since $\sin^2 \alpha + \cos^2 \alpha = 1$, we have

$$\cos \alpha = \sqrt{1 - \sin^2 \alpha} = \sqrt{1 - 0.16} = \sqrt{0.84}.$$

Note that $0.84 = \frac{84}{100} = \frac{21}{25}$, so $\cos \alpha = \sqrt{\frac{21}{25}} = \frac{\sqrt{21}}{5}$.

Similarly,

$$\cos \beta = \sqrt{1 - \sin^2 \beta} = \sqrt{1 - 0.25} = \sqrt{0.75} = \sqrt{\tfrac{3}{4}} = \frac{\sqrt{3}}{2}.$$

Now, let me compute $\sin(\alpha + \beta)$. Using the formula,

$$\sin(\alpha + \beta) = \sin \alpha \cos \beta + \cos \alpha \sin \beta = 0.4\left(\frac{\sqrt{3}}{2}\right) + \left(\frac{\sqrt{21}}{5}\right)(0.5).$$

Compute each term:

$$0.4 \cdot \frac{\sqrt{3}}{2} = \frac{2}{5} \cdot \frac{\sqrt{3}}{2} = \frac{\sqrt{3}}{5}, \quad \frac{\sqrt{21}}{5} \cdot 0.5 = \frac{\sqrt{21}}{10}.$$

Therefore

$$\sin(\alpha + \beta) = \frac{\sqrt{3}}{5} + \frac{\sqrt{21}}{10} = \frac{2\sqrt{3} + \sqrt{21}}{10}.$$

Similarly, compute $\sin(\beta - \alpha)$:

$$\sin(\beta - \alpha) = \sin \beta \cos \alpha - \cos \beta \sin \alpha = 0.5\left(\frac{\sqrt{21}}{5}\right) - \frac{\sqrt{3}}{2}(0.4).$$

Thus

$$\sin(\beta - \alpha) = \frac{\sqrt{21}}{10} - \frac{2\sqrt{3}}{10} = \frac{\sqrt{21} - 2\sqrt{3}}{10}.$$

Therefore, the original expression is

$$\left(\frac{2\sqrt{3} + \sqrt{21}}{10}\right)\left(\frac{\sqrt{21} - 2\sqrt{3}}{10}\right).$$

Let $A = 2\sqrt{3} + \sqrt{21}$ and $B = \sqrt{21} - 2\sqrt{3}$; then the product is $\frac{A\,B}{100}$. Note that $A\,B = (\sqrt{21})^2 - (2\sqrt{3})^2 = 21 - 12 = 9$, so the value is $\dfrac{9}{100}$.

Response Generated by RL-Finetuned Qwen3-8B using Critique-GRPO (2/2)

Therefore, the value of the original expression is $\frac{9}{100}$, which is 0.09.

Let me check with approximate values. Take $\arcsin 0.5 = \frac{\pi}{6} \approx 0.5236\,\text{rad}$ and $\arcsin 0.4 \approx 0.4115\,\text{rad}$. Then $\alpha + \beta \approx 0.9351\,\text{rad}$, so $\sin(\alpha+\beta) \approx 0.805$; also $\beta - \alpha \approx 0.1121\,\text{rad}$, so $\sin(\beta-\alpha) \approx 0.1118$. Their product $0.805 \times 0.1118 \approx 0.09$, confirming $\frac{9}{100}$.

Final Answer

$$\boxed{\frac{9}{100}}$$

To evaluate the expression

$$\sin(\arcsin 0.4 + \arcsin 0.5) \cdot \sin(\arcsin 0.5 - \arcsin 0.4),$$

we begin by introducing a substitution to simplify the notation. Let

$$\alpha = \arcsin 0.4, \qquad \beta = \arcsin 0.5,$$

so that $\sin\alpha = 0.4$ and $\sin\beta = 0.5$.

We rewrite the expression as $\sin(\alpha+\beta)\,\sin(\beta-\alpha)$.

Step 1: Use the Sine Addition and Subtraction Formulas

Recall

$$\sin(\alpha+\beta) = \sin\alpha\,\cos\beta + \cos\alpha\,\sin\beta, \qquad \sin(\beta-\alpha) = \sin\beta\,\cos\alpha - \cos\beta\,\sin\alpha.$$

Using

$$\cos\alpha = \frac{\sqrt{21}}{5}, \qquad \cos\beta = \frac{\sqrt{3}}{2},$$

we compute

$$\sin(\alpha+\beta) = \frac{2\sqrt{3}+\sqrt{21}}{10}, \quad \sin(\beta-\alpha) = \frac{\sqrt{21}-2\sqrt{3}}{10}.$$

Step 2: Multiply the Two Sine Values

$$\left(\frac{2\sqrt{3}+\sqrt{21}}{10}\right)\left(\frac{\sqrt{21}-2\sqrt{3}}{10}\right) = \frac{(\sqrt{21})^2-(2\sqrt{3})^2}{100} = \frac{9}{100}.$$

Final Answer

$$\boxed{\frac{9}{100}}$$

J An Example of Failed Refinement

The following example demonstrates a failed refinement using a CoT critique, where RL-finetuned Qwen3-8B using Critique-GRPO fails to follow the critique to refine its responses.

Question
Example 15: Let a, b, c be positive real numbers, and $abc + a + c = b$, find the maximum value of

$$p = \frac{2}{a^2+1} - \frac{2}{b^2+1} + \frac{3}{c^2+1}.$$

CoT Critique
The student's solution contains several errors and misunderstandings. Let's go through the mistakes step-by-step:
1. **Misinterpretation of the Constraint:** - The student correctly rearranges the constraint $abc+a+c = b$ to express b in terms of a and c, but then incorrectly assumes $b = \tan(\alpha+\gamma)$ without proper justification. The trigonometric substitution should be used more carefully, ensuring that the constraint is satisfied throughout.
2. **Incorrect Use of Trigonometric Identities:** - The student attempts to use trigonometric identities to simplify the expression for p, but the application is incorrect. The identities used do not lead to a valid simplification of the expression.
3. **Incorrect Derivative Calculation:** - The student attempts to find the maximum by taking derivatives, but the calculations are incorrect. The derivative should be taken with respect to the correct variables, and the critical points should be found accurately.
4. **Incorrect Evaluation of p:** - The student evaluates p at specific points, but the calculations are inconsistent and do not match the constraint. The values chosen for a, b, and c do not satisfy the constraint $abc + a + c = b$.
5. **Incorrect Final Answer:** - The student concludes with an incorrect maximum value of $p = \frac{105}{32}$, which is not supported by the calculations. The correct maximum value, as derived from a proper analysis, should be $\frac{10}{3}$.
Conclusion: wrong [END]

Incorrect Initial Response (1/3)
Okay, so I need to find the maximum value of this expression:

$$p = \frac{2}{a^2+1} - \frac{2}{b^2+1} + \frac{3}{c^2+1},$$

given that a, b, c are positive real numbers satisfying $abc + a + c = b$. Hmm, let me start by understanding the constraint equation better.
The equation is:

$$abc + a + c = b.$$

Maybe I can rearrange this to express one variable in terms of the others. Let me try to solve for b. Let's see:

$$abc + a + c = b \implies b - abc = a + c \implies b(1 - ac) = a + c \implies b = \frac{a+c}{1-ac}.$$

Wait, but since a, b, c are positive real numbers, the denominator $1 - ac$ must be positive because the numerator $a + c$ is positive. Therefore:

$$1 - ac > 0 \implies ac < 1.$$

So, the product of a and c must be less than 1. That's an important condition.
—
Now, the expression for p involves a, b, c. Since b is expressed in terms of a and c, perhaps I can substitute that into p and then try to maximize with respect to a and c. Let me try that.
Substituting $b = \frac{a+c}{1-ac}$ into p, we get:

$$p = \frac{2}{a^2+1} - \frac{2}{\left(\frac{a+c}{1-ac}\right)^2 + 1} + \frac{3}{c^2+1}.$$

This looks complicated. Maybe there's a substitution or some symmetry here. Let me check if there's a way to simplify the expression for b. Alternatively, maybe using substitution variables.
—
Trigonometric Substitution
Let's consider substituting variables to simplify the constraint. For example, sometimes in equations with products, substitutions like $x = a, y = c$ are helpful, but I'm not sure in this case. Alternatively, maybe trigonometric substitution? Since we have terms like $a^2 + 1$, which reminds me of $\tan\theta$ substitutions. For example:
- Let $a = \tan\alpha$, $c = \tan\gamma$, where $\alpha, \gamma \in (0, \pi/2)$. - Then:

$$a^2 + 1 = \sec^2\alpha \implies \frac{1}{a^2+1} = \cos^2\alpha,$$

and similarly for c.
Let me try that. Substituting $a = \tan\alpha$ and $c = \tan\gamma$, and using the constraint $abc + a + c = b$, we get:

$$\tan\alpha \cdot b \cdot \tan\gamma + \tan\alpha + \tan\gamma = b.$$

Solving for b:

$$b(\tan\alpha\tan\gamma - 1) = \tan\alpha + \tan\gamma \implies b = \frac{\tan\alpha + \tan\gamma}{\tan\alpha\tan\gamma - 1}.$$

Wait, but earlier we had:

$$b = \frac{a+c}{1-ac}.$$

Incorrect Initial Response (2/3) Since $ac = \tan\alpha\tan\gamma$, so $1 - ac = 1 - \tan\alpha\tan\gamma$. Therefore:

$$b = \frac{\tan\alpha + \tan\gamma}{1 - \tan\alpha\tan\gamma}.$$

—
Simplifying Further
Notice that:

$$\frac{\tan\alpha + \tan\gamma}{1 - \tan\alpha\tan\gamma} = \tan(\alpha + \gamma).$$

Thus:

$$b = \tan(\alpha + \gamma).$$

Therefore, since $b > 0$, $\tan(\alpha + \gamma) > 0$. Since $\alpha, \gamma \in (0, \pi/2)$, $\alpha + \gamma \in (0, \pi/2)$. This gives:

$$\alpha + \gamma < \frac{\pi}{2}.$$

—
Substituting Back into p
Using the substitution:

$$p = \frac{2}{a^2 + 1} - \frac{2}{b^2 + 1} + \frac{3}{c^2 + 1}.$$

Since $a = \tan\alpha$, we have:

$$\frac{1}{a^2 + 1} = \cos^2\alpha, \quad \frac{1}{c^2 + 1} = \cos^2\gamma, \quad \frac{1}{b^2 + 1} = \cos^2(\alpha + \gamma).$$

Thus:

$$p = 2\cos^2\alpha - 2\cos^2(\alpha + \gamma) + 3\cos^2\gamma.$$

—
Reducing the Problem
The problem now reduces to maximizing:

$$p = 2\cos^2\alpha - 2\cos^2(\alpha + \gamma) + 3\cos^2\gamma,$$

subject to $\alpha, \gamma > 0$ and $\alpha + \gamma < \pi/2$.
Let $\theta = \alpha + \gamma$. Then:

$$p = 2\cos^2\alpha - 2\cos^2\theta + 3\cos^2(\theta - \alpha).$$

—
Symmetry Assumption
Assume $\alpha = \gamma$, which implies $\theta = 2\alpha$. Then:

$$p = 2\cos^2\alpha - 2\cos^2(2\alpha) + 3\cos^2\alpha = 5\cos^2\alpha - 2\cos^2(2\alpha).$$

Using the identity $\cos(2\alpha) = 2\cos^2\alpha - 1$, we get:

$$\cos^2(2\alpha) = (2\cos^2\alpha - 1)^2.$$

Let $x = \cos^2\alpha$, so $x \in \left(\frac{1}{2}, 1\right)$. Substituting:

$$p = 5x - 2(2x - 1)^2 = 5x - 2(4x^2 - 4x + 1) = -8x^2 + 13x - 2.$$

Incorrect Initial Response (3/3)
This is a quadratic equation, and its maximum occurs at:

$$x = \frac{-b}{2a} = \frac{13}{16}.$$

Substituting $x = \frac{13}{16}$ back into p:

$$p = -8\left(\frac{13}{16}\right)^2 + 13\left(\frac{13}{16}\right) - 2 = \frac{105}{32}.$$

—
Final Answer

$$\boxed{\frac{105}{32}}$$

Incorrect Refined Response (1/2)
The given problem asks us to find the **maximum value** of the expression:

$$p = \frac{2}{a^2+1} - \frac{2}{b^2+1} + \frac{3}{c^2+1}$$

subject to the constraint:

$$abc + a + c = b$$

where a, b, c are **positive real numbers**.
—
Step 1: Analyze the Constraint
From the constraint:

$$abc + a + c = b$$

we can rearrange to isolate b:

$$b(1 - ac) = a + c \quad \Rightarrow \quad b = \frac{a+c}{1-ac}.$$

Since $a, b, c > 0$, the denominator $1 - ac$ must be positive, so:

$$ac < 1$$

This condition will be important in limiting the domain of possible values for a and c.
—
Step 2: Trigonometric Substitution
To simplify the expression for p, we use the following substitutions:
- Let $a = \tan\alpha$ - Let $c = \tan\gamma$
Then, since $a^2 + 1 = \sec^2\alpha$, we have:

$$\frac{1}{a^2+1} = \cos^2\alpha, \quad \frac{1}{c^2+1} = \cos^2\gamma.$$

Also, from the tangent addition formula:

$$b = \frac{a+c}{1-ac} = \tan(\alpha+\gamma).$$

Therefore:

$$\frac{1}{b^2+1} = \cos^2(\alpha+\gamma).$$

Substituting into p, we get:

$$p = 2\cos^2\alpha - 2\cos^2(\alpha+\gamma) + 3\cos^2\gamma.$$

Let $\theta = \alpha + \gamma$, and express $\gamma = \theta - \alpha$. Then:

$$p = 2\cos^2\alpha - 2\cos^2\theta + 3\cos^2(\theta - \alpha).$$

—

Incorrect Refined Response (2/2)
Step 3: Assume Symmetry for Simplification
To simplify further, assume $\alpha = \gamma$, which implies $\theta = 2\alpha$. Then:

$$p = 2\cos^2\alpha - 2\cos^2(2\alpha) + 3\cos^2\alpha = 5\cos^2\alpha - 2\cos^2(2\alpha).$$

Using the double-angle identity $\cos(2\alpha) = 2\cos^2\alpha - 1$, we get:

$$\cos^2(2\alpha) = (2\cos^2\alpha - 1)^2.$$

Let $x = \cos^2\alpha$, so $x \in \left(\frac{1}{2}, 1\right)$. Then:

$$p = 5x - 2(2x-1)^2 = -8x^2 + 13x - 2.$$

This is a quadratic in x, opening downward, so its maximum occurs at:

$$x = \frac{-b}{2a} = \frac{13}{16}.$$

Substituting back:

$$p = -8\left(\frac{13}{16}\right)^2 + 13\left(\frac{13}{16}\right) - 2 = \frac{105}{32}.$$

—
Final Answer

$$\boxed{\frac{105}{32}}$$