RealCritic: Towards Effectiveness-Driven Evaluation of Language Model Critiques

Anonymous ACL submission

Abstract

Critiques are crucial for enhancing Large Language Models (LLMs), yet evaluating this capability is notoriously difficult. Existing benchmarks often rely on *open-loop* methods, such as matching a critique's verdict to a ground-truth label, which fails to capture the practical value of the feedback. In this work, we propose a new **effectiveness-driven evaluation paradigm** that shifts the focus from judging the critique itself to measuring its impact. Our *closed-loop* methodology assesses critique quality based on a simple, yet powerful principle: a good critique should lead to a correct solution. We measure the quality of a critique by the accuracy of the final answer generated after correction based on that critique.

To instantiate this paradigm, we introduce **RealCritic**, a comprehensive benchmark built on eight challenging reasoning tasks. RealCritic is the first to systematically evaluate often-overlooked yet critical scenarios: *self-critique*, *cross-critique*, and *iterative critique*. Our experiments reveal a significant finding: while classical LLMs show comparable performance to advanced reasoning-based models like o1-mini in direct problem-solving, they lag dramatically in all critique scenarios. This shows that our approach uncovers a deeper, more fundamental dimension of model capability. We hope this new evaluation paradigm and the accompanying benchmark will steer future research towards developing more robust and genuinely self-improving LLMs.

1 Introduction

Recent advancements in Large Language Models (LLMs) (Hurst et al., 2024; Dubey et al., 2024; Mistral-AI, 2024; Qwen-Team, 2024) have demonstrated remarkable capabilities across lots of tasks, ranging from natural language understanding to problem-solving. Notably, reasoning-based LLMs have surpassed human expertise in specialized domains such as competition-level mathematical reasoning and code generation (OpenAI, 2024). However, as LLMs continue to evolve, further improving their performance has become increasingly challenging. One of the key areas of focus in this regard is the development of **critique abilities** (McAleese et al., 2024)—the capability of LLMs to provide detailed analysis, constructive feedback, and refinement suggestions for solutions generated by other models or even themselves, thereby driving continuous improvement in their outputs.

An important step toward advancing critique abilities is the establishment of high-quality benchmarks for evaluation. Without such benchmarks, it is difficult to objectively measure progress in this area. However, evaluating the quality of critiques is difficult due to the open-ended nature. That is, unlike tasks with definitive answers, critique evaluation lacks a single ground truth for making conclusive judgments about performance.

Significant efforts have been made to benchmark the critique capabilities of LLMs. For instance, CriticBench (Luo et al., 2023; Zheng et al., 2024) requires LLMs to generate Critiques accompanied by binary Verdicts (C+V) for input solutions, where the verdicts predict whether the solutions are correct or incorrect. The quality of the critiques is then evaluated by comparing these verdicts to ground truth labels, providing a measure of accuracy. CriticEval (Lan et al., 2024) conducts a comprehensive evaluation of LLMs' critique abilities, introducing two key dimensions relevant to our study: feedback and correction. The feedback dimension assesses the quality of the analysis within the critique by assigning a prediction score, while the correction dimension evaluates the quality of revisions based on the critique's analysis. Beyond dedicated benchmarks, other works such as Shepherd (Wang et al., 2023) and Auto-J (Li et al., 2023) have proposed using GPT-4 to perform comparative evaluations between different critiques. See Appendix A for a

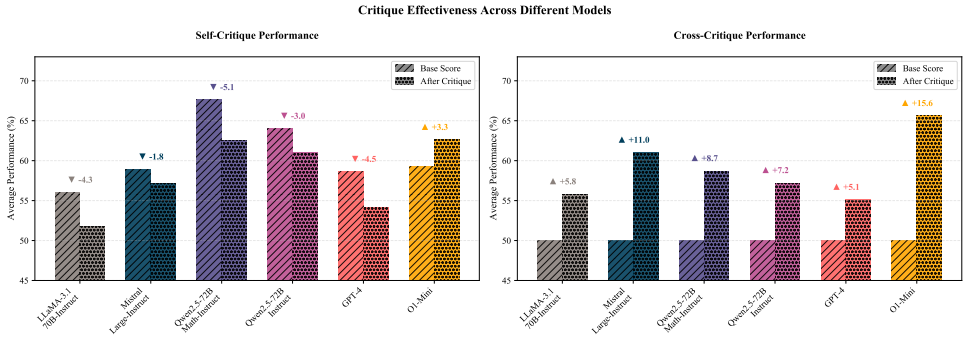

Figure 1: Benchmark results for self-critique and cross-critique abilities of representative LLMs.

review of additional work.

Despite these advancements, the key question remains: *how can we effectively measure the quality of critiques?* This challenge is supported by our preliminary study on CriticBench. Using human evaluators, we find that the benchmark suffers from a high *misjudgment* rate (about 30% of low-quality critiques are erroneously classified as high quality; see Table 2 in Section 2). Additionally, we observe that existing benchmarks often lack support for essential critique tasks, such as self-critique and iterative critique, which are believed to be crucial for reasoning-based LLMs (Wu et al., 2024a).

In this paper, we propose RealCritic, a new benchmark designed to evaluate critique abilities based on their effectiveness in driving refinement; see Table 1 for an overview. Recognizing that *a critique is high-quality if it contributes meaningfully to the self-improvement of LLMs*, we propose to measure critique quality directly based on the correction it enables. Our approach is *closed-loop*, drawing inspiration from control theory (Franklin et al., 2002), where the output (correction) is fed back into the system to assess the effectiveness of the input (critique). In contrast, previous methods for measuring critique quality are often *open-loop*, evaluating critiques in isolation without considering their downstream impact on solution refinement or treating correction abilities separately from critique abilities. Furthermore, our benchmark introduces advanced evaluation scenarios to assess LLMs' critique capabilities. Specifically, we distinguish between self-critique and cross-critique abilities and incorporate iterative critique, providing a more comprehensive framework.

We implement RealCritic using eight challenging tasks that, while grounded in verifiable answers, test a wide spectrum of reasoning abilities beyond pure computation. These tasks include not only complex **mathematical problem-solving** (e.g., GSM8K, MATH, OlympiadBench), but also **scientific reasoning** across domains like physics and biology (GPQA), **commonsense scientific analysis** (ARC), and **interdisciplinary STEM knowledge** (MMLU-STEM). This diversity ensures we evaluate critique capabilities on a robust set of logical and analytical challenges, providing a strong basis for assessing general reasoning refinement. The common thread is their verifiable ground truth, which is essential for our closed-loop evaluation, even when the critiques themselves lack a single correct answer. We evaluate several state-of-the-art models, including LLaMA-3.1-70B-Instruct (Dubey et al., 2024), Qwen2.5-72B-Instruct (Qwen-Team, 2024), Mistral-Large-Instruct-2407 (Mistral-AI, 2024), and o1-mini (OpenAI, 2024). One of our key findings is shown in Figure 1, with additional results and analyses detailed in the main text.

To summarize, our main contribution is a new effectiveness-driven paradigm for evaluating LLM critique capabilities.

- We propose a **closed-loop evaluation method** that measures critique quality by its direct impact on solution refinement. This effectiveness-driven approach overcomes the fundamental flaws of existing open-loop, verdict-based evaluations.

- We develop **RealCritic**, a new benchmark designed to implement our proposed methodology. It is the first to systematically support essential critique functions like *self-critique*, *cross-critique*, and *iterative critique*, providing a more holistic evaluation framework.

- Through experiments on RealCritic, we conduct a **comprehensive analysis** that reveals a stark gap in critique ability between classical LLMs and advanced reasoning models. This not only validates our method's ability to

Dataset	Output Format	Critique Type	Evaluation Method	Iterative Support	Test Data Size	Public Release
CriticBench (Luo et al., 2023)	C+V	Self+Cross	Verdict Matching	✗	0	✗
CriticBench (Lin et al., 2024)	C+V	Cross	Verdict Matching	✗	3,825	✓
CriticEval (Lan et al., 2024)	C+S	Cross	Score Judging	✗	3,608	✓
Shepherd (Wang et al., 2023)	C	Cross	GPT-4 Comparison	✗	352	✗
Auto-J (Li et al., 2023)	C	Cross	GPT-4 Comparison	✗	232	✓
RealCritic	C+V+Corr	Self+Cross	Correction Matching	✓	2,093	✓

Table 1: Critique Eval Methods. C = Critique, V = Verdict, S = Score, Corr = Correction. ✓= Yes, ✗ = No.

uncover deeper model traits but also provides critical insights for future model development.

2 Toward Effectiveness-Driven Evaluation

In this section, we discuss strategies for constructing an effective benchmark for evaluating critiques. Evaluating the critique capabilities of LLMs is inherently challenging due to the multifaceted and open-ended nature of the task. A high-quality critique must fulfill several criteria: it should identify flaws or weaknesses in the solution, provide constructive and actionable feedback, and maintain clarity and logical consistency. These requirements complicate the evaluation process, as each sub-part allows for a wide range of valid responses. While human evaluators may perform well in assessing critique quality, human-based evaluation methods are neither automatic nor scalable.

To address these challenges, it is crucial to leverage task-specific properties. Reasoning tasks, for instance, are particularly well-suited for this purpose, as they share a key characteristic: the final generated answer is typically a single, definitive response. This property can be utilized to facilitate the evaluation of solutions. CriticBench (Luo et al., 2023; Lin et al., 2024) leverages this property to evaluate critiques through the following process:

- Critic Generation: This benchmark requires LLMs to generate critiques that provide a detailed analysis of a solution. In addition to the analysis, the model must also produce a verdict predicting whether the input solution is correct or incorrect.

- Quality Evaluation: The benchmark assesses the quality of the critique based on the correctness of the verdict. If the verdict is correct, the critique is deemed high quality. Conversely, if the verdict is incorrect, the critique is judged as low quality.

CriticEval (Lan et al., 2024) uses a similar approach in its evaluation framework, particularly in the dimension they define as “feedback". We would like to point out that the fundamental premise of this approach is to evaluate critique quality based on the accuracy of predicting the solution’s correctness. We argue that this evaluation method may not always be effective, as an LLM might correctly predict the solution’s accuracy without providing a constructive analysis or critique. This misalignment stems from the fact that the evaluation approach does not fully capture the intuition that a critique is considered high-quality if it leads to an improved solution, rather than predicting the accuracy of the input solution. A specific example illustrating this limitation is shown in Figure 2. Furthermore, when using human evaluators to provide ground truth assessments of critique quality, we find that CriticBench suffers from a high misjudgment rate, further highlighting its limitations as shown in Table 2 (see details and more concrete examples in Appendix B).

Critique Quality	Verdict: Matched	Verdict: Unmatched	Total
High-quality	55.3	14.6	69.9
Low-quality	26.0	4.1	30.1
Total	81.3	18.7	100.0

Table 2: Human evaluation results of critique quality on CriticBench (Lin et al., 2024). All values are in percentages (%).

We believe the aforementioned fragility issue arises from the **open-loop** nature of the evaluation method employed. Specifically, the quality of the generated critique is not directly linked to its impact on subsequent refinement, leading to potential inconsistencies in evaluation. While this issue is briefly mentioned in CriticEval (Lan et al., 2024), the work does not empirically validate the severity of the issue, as we have done in our study.

In addition to the limitations in evaluation metrics discussed earlier, we find that existing benchmarks lack comprehensive modes for evaluating critique abilities. First, most benchmarks focus on critiquing input solutions that have been previously collected, operating in a cross-critique mode.

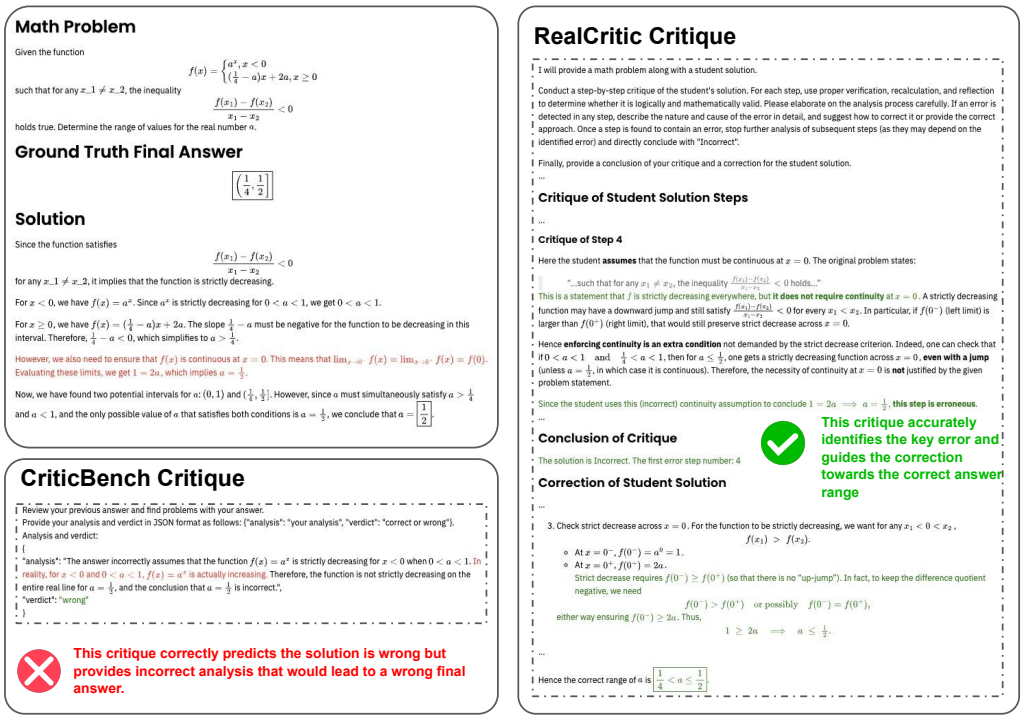

Figure 2: CriticBench (Lin et al., 2024) vs. RealCritic (ours) quality assessment comparison. CriticBench misclassifies low-quality critiques as high-quality using solution prediction accuracy. RealCritic correctly identifies quality based on effectiveness in improving solutions.

However, there is no systematic exploration of another important mode: self-critique, where input solutions are model-specific and dynamically generated. Additionally, existing benchmarks do not include an evaluation mode for multi-round iterative critique, which is essential for examining the long-horizon critique abilities of LLMs and could be particularly crucial for evaluating emergent models such as o1 (OpenAI, 2024). We plan to address these limitations in the next section by introducing a new benchmark.

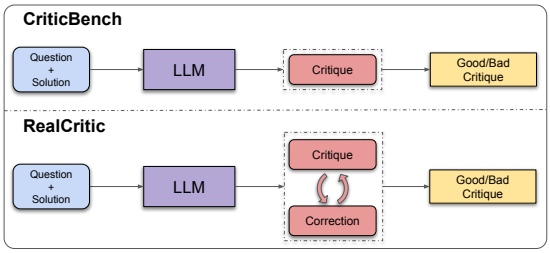

Figure 3: Comparison between the evaluation method used in CriticBench and our framework, RealCritic. Our framework operates in a closed-loop manner by assessing the quality of a critic through the quality of the new solution generated based on its feedback.

3 Our Evaluation Framework

In this section, we present our evaluation framework for benchmarking the critique abilities of LLMs in mathematical reasoning domains. Our framework is built around three key components: the closed-loop evaluation method described in Section 3.1, the spectrum of critique abilities outlined in Section 3.2, and the targeted datasets we curate for evaluation, as detailed in Section 3.3.

3.1 From Open-Loop to Closed-Loop Evaluation

Most benchmarks use open-loop evaluation, which fails to link critiques to their goal—improving solutions. We propose a closed-loop, effectiveness-driven evaluation, defining a critique's effectiveness by its ability to guide a model toward a more accurate answer. This approach is objective and measures practical utility. A key challenge is ensuring models perform genuine critiques rather than re-solving. To address this, we introduce a two-stage verification process:

- **Procedural compliance:** An automated post-check (Appendix C) filters non-compliant critiques (e.g., skipping critique-then-correct). Manual audit shows 98% adherence.
- **Quantitative evidence:** We measure semantic differences between direct CoT solutions and critique-based corrections. Table 3 shows high normalized Levenshtein distances, con-

Model	Avg. Edit Distance
LLaMA-3.1-70B-Instruct	0.679
Mistral-Large-Instruct	0.559
Qwen2.5-72B-Math-Instruct	0.440
Qwen2.5-72B-Instruct	0.559
GPT-4o	0.691
o1-mini	0.678

Table 3: Average normalized Levenshtein distance between direct CoT solutions and critique-based corrections. High scores indicate that corrections are not mere re-generations of initial solutions.

firming corrections are distinct from initial solutions.

With this verification in place, we can confidently measure critique effectiveness by evaluating the quality of the final solution. Formally, we define the performance as:

$$\text{performance} = \mathbb{E}_{x\sim\rho}[\mathbb{I}(y_{\text{correction}})],$$
$$(y_{\text{critic}}, y_{\text{correction}}) \sim \mathbb{P}_{\text{Critic-LLM}}(x, s, I_{\text{critic}}),$$

where x denotes a question drawn from the distribution ρ. The term $\mathbb{P}_{\text{Critic-LLM}}$ represents the distribution over outputs of the (critic) LLM, given an initial solution s and an instruction I_{critic} that prompts for both the critique y_{critic} and the correction $y_{\text{correction}}$. The function $\mathbb{I}$ is an indicator for the correctness of the final correction. This closed-loop, verified approach ensures we are measuring the true, practical value of a critique.

3.2 What to Critique and How to Critique?

After defining evaluation criteria, we design test cases to assess critique ability and process, emphasizing two key but often overlooked distinctions. First, self-critique (where the model evaluates its own output) is less studied than cross-critique (evaluating others' outputs), despite its importance for iterative refinement. Second, iterative critique (multi-round feedback) better tests long-horizon reasoning than single-round critiques, especially for complex tasks. We explore these in detail below.

Self-Critique Paradigm: The self-critique paradigm focuses on assessing a model's ability to identify and improve errors in its own generated solutions. Formally, the quality of critique is measured by:

$$\text{performance}_{\text{self}} = \mathbb{E}_{x\sim\rho}[\mathbb{I}(y_{\text{correction}})],$$
$$(y_{\text{critic}}, y_{\text{correction}}) \sim \mathbb{P}_{\text{Critic-LLM}}(x, s, I_{\text{critic}}),$$
$$s \sim \mathbb{P}_{\text{Critic-LLM}}(x, I_{\text{generation}}).$$

A key feature of this paradigm is that the input solutions for critique are not fixed but are instead model-specific. This paradigm evaluates the model's introspective capabilities, specifically its ability to recognize its own errors, analyze their root causes, and implement effective corrections. However, since the critic and solution generator are the same model, they inherently share the same knowledge base and potential biases, which may limit the diversity and objectivity of the critiques.

Cross-Critique Paradigm: The cross-critique paradigm evaluates a model's ability to analyze and improve answers generated by other models. This approach is widely used in benchmarks, partly because it is straightforward to implement, as input solutions for critique can be pre-collected. This paradigm is crucial for understanding how effectively a model can serve as an external evaluator. Formally, the quality of critique is measured by:

$$\text{performance}_{\text{cross}} = \mathbb{E}_{x\sim\rho}[\mathbb{I}(y_{\text{correction}})],$$
$$(y_{\text{critic}}, y_{\text{correction}}) \sim \mathbb{P}_{\text{Critic-LLM}}(x, s, I_{\text{critic}}),$$
$$s \sim \mathbb{P}_{\text{LLM}}(x, I_{\text{generation}}),$$

where $\mathbb{P}_{\text{LLM}}$ may involve any language model distribution, which is distinct from the critic's distribution and could even be a combination of multiple models. This diversity means that the critic must handle a wide range of solution styles and error types, ensuring versatile evaluation capabilities.

Iterative Critique-Correction Paradigm: Drawing on the success of o1-style models (OpenAI, 2024), it becomes evident that multi-round self-reflection and iterative refinement play a decisive role in mathematical reasoning. By enabling several cycles of critique-and-correction, we can thoroughly evaluate the model's continuous improvement across successive iterations, investigate the convergence properties underpinning its error-correction process, and analyze how self-supervision mechanisms influence performance—ultimately shedding light on the model's capacity for robust and adaptive problem solving.

Formally, we define the iterative process:

$$\text{performance}^t_{\text{iterative}} = \mathbb{E}_{x\sim\rho}[\mathbb{I}(y^t_{\text{correction}})],$$
$$(y^t_{\text{critic}}, y^t_{\text{correction}}) \sim \mathbb{P}_{\text{Critic-LLM}}(x, s, I_{\text{critic}}, y^1_{\text{critic}}, y^1_{\text{correction}}, \ldots, y^{t-1}_{\text{critic}}, y^{t-1}_{\text{correction}}),$$
$$s \sim \mathbb{P}_{\text{LLM}}(x, I_{\text{generation}}),$$

where t denotes the iteration index compared with previous notions. This formulation highlights the model's capability to refine its output through re-

peated self-critique, shedding light on its long-horizon reasoning and error-correction dynamics.

3.3 Dataset Collection

To benchmark critique abilities, we construct datasets through a two-step process: collecting questions from diverse tasks and generating solutions using multiple models.

Task Sources. We selected eight representative datasets from a pool of 20 candidates, prioritizing tasks with high reasoning complexity and verifiable ground-truth answers. Our selection is intentionally diverse to test critique capabilities across a spectrum of analytical skills. The benchmark includes five datasets focused on **mathematical reasoning**, which test multi-step logical deduction and computation, ranging from grade-school problems (GSM8K, Cobbe et al. 2021) to advanced Olympiad-level challenges (OlympiadBench, He et al. 2024). To assess broader reasoning abilities, we included three datasets focused on **scientific and STEM reasoning**: GPQA-diamond (Rein et al., 2023), which contains expert-level scientific problems; ARC-Challenge (Clark et al., 2018), which tests commonsense scientific analysis; and MMLU-STEM (Hendrycks et al., 2020a), covering interdisciplinary topics. This diverse collection allows us to evaluate whether a model's critique ability is a generalizable skill or confined to specific domains. Further details on our selection process are in Appendix D.

Solution Sources. We generate solutions using open-source models including Qwen2.5 series (Qwen-Team, 2024), LLaMA-3.1 series (Dubey et al., 2024), and DeepSeek math-specialized model (Shao et al., 2024). For incorrect solutions, we prioritize stronger models that typically provide more detailed reasoning; for correct solutions, we prioritize weaker models that often include challenging intermediate steps.

Benchmark Format. Our benchmark comprises eight datasets with balanced distributions between correct and incorrect solutions (approximately 1:1 ratio) as shown in Table **??**. The benchmark supports self-critique, cross-critique, and iterative critique protocols. Further details on datasets and solution selection strategies are provided in Appendix D.

4 Experiments

In this section, we conduct experiments using the proposed evaluation benchmark. We provide the experiment configuration below.

Models: We evaluate several state-of-the-art language models, including open-source models (LLaMA-3.1-70B-Instruct (Dubey et al., 2024), Mistral-Large-Instruct-2407 (Mistral-AI, 2024), Qwen2.5-72B-Math-Instruct, and Qwen2.5-72B-Instruct (Qwen-Team, 2024)) and proprietary models (GPT-4o (Hurst et al., 2024) and o1-mini (OpenAI, 2024)).

Implementation Details: Open-source models were deployed via the vLLM framework for efficient inference; proprietary models were accessed via their official APIs. To ensure full reproducibility, we used deterministic decoding with a temperature of 0.0. Thus, all reported results are from a single, deterministic run. This design choice removes sampling variance, meaning that statistics like error bars from multiple runs are not relevant for assessing the stability of our results. We used a maximum token limit of 2,048 and top-p of 1.0.

4.1 Results and Analysis

Our main results are presented in Table 4, Figure 1, and Figure 5. We dissect these results in the following sections, analyzing each critique capability in detail. To validate the generalizability of our findings, we conducted supplementary experiments on additional models; see Appendix E for details and analysis.

4.1.1 Self-Critique Performance Analysis

We find that self-critique performance varies significantly across models, with most showing modest changes (-5.1% to +3.3%) compared to their direct CoT solutions, indicating that error self-correction remains challenging even for advanced language models. O1-mini stands out remarkably, achieving a +3.3% average improvement over direct solutions, with exceptional gains on College Math (+24.0%), ARC (+19.4%), MATH (+10.1%), and even OlympiadBench (+6.4%), suggesting its architecture or training approach may be uniquely suited for self-reflection tasks.

Furthermore, we observe a concerning pattern across all models in self-critique performance on certain specialized tasks. Models including o1-mini consistently struggle with MMLU-STEM and GPQA tasks, showing significant performance degradation ranging from -10% to -35.6%. This pattern is particularly pronounced in Qwen2.5-72B-Math-Instruct's performance on MMLU-STEM (-35.6%) and o1-mini's performance on MMLU-

Model	Evaluation Mode	ARC	GSM8K	MATH	College Math	Minerva Math	GPQA	MMLU STEM	Olympiad Bench	Avg
LLaMA-3.1-70B-Instruct	Direct COT	88.6	93.4	65.3	40.1	33.7	32.6	63.0	31.1	56.0
	Self-Critique	87.5	91.9	59.0	37.3	33.7	22.1	53.8	28.5	51.7
	Δ*(Self-Direct)*	-1.1	-1.5	-6.3	-2.8	0.0	-10.5	-9.2	-2.6	-4.3
	Cross Critique	82.5	88.6	64.9	49.1	42.1	35.3	42.8	40.8	55.8
	Δ*(Cross-Base)*	32.5	38.6	14.9	-0.9	-7.9	-14.7	-7.2	-9.2	5.8
Mistral-Large-Instruct	Direct COT	86.7	94.5	70.1	41.6	35.6	41.1	66.8	34.5	58.9
	Self-Critique	85.2	93.4	70.9	41.6	36.8	36.3	56.5	36.0	57.1
	Δ*(Self-Direct)*	-1.5	-1.1	0.8	0.0	1.2	-4.8	-10.3	1.5	-1.8
	Cross Critique	84.8	90.8	69.0	50.9	49.4	40.5	53.8	48.7	61.0
	Δ*(Cross-Base)*	34.8	40.8	19.0	0.9	-0.6	-9.5	3.8	-1.3	11.0
Qwen2.5-72B-Math-Instruct	Direct COT	94.3	95.6	86.9	45.2	42.1	42.6	86.3	47.6	67.6
	Self-Critique	93.5	95.2	85.4	45.2	41.0	41.6	50.7	47.2	62.5
	Δ*(Self-Direct)*	-0.8	-0.4	-1.5	0.0	-1.1	-1.0	-35.6	-0.4	-5.1
	Cross Critique	72.6	86.4	73.1	52.0	50.2	43.7	38.0	53.9	58.7
	Δ*(Cross-Base)*	22.6	36.4	23.1	2.0	0.2	-6.3	-12.0	3.9	8.7
Qwen2.5-72B -Instruct	Direct COT	94.3	96.7	82.5	46.2	37.9	38.9	70.9	44.2	64.0
	Self-Critique	92.4	96.7	82.5	46.2	37.2	29.5	59.9	43.8	61.0
	Δ*(Self-Direct)*	-1.9	0.0	0.0	0.0	-0.7	-9.4	-11.0	-0.4	-2.9
	Cross Critique	84.8	88.6	71.6	51.3	47.1	26.8	41.4	46.1	57.2
	Δ*(Cross-Base)*	34.8	38.6	21.6	1.3	-2.9	-23.2	-8.6	-3.9	7.2
GPT-4o	Direct COT	92.4	94.9	76.5	41.6	33.0	34.2	59.9	36.3	58.6
	Self-Critique	89.4	93.4	74.6	40.5	30.7	24.2	42.5	37.1	54.1
	Δ*(Self-Direct)*	-3.0	-1.5	-1.9	-1.1	-2.3	-10.0	-17.4	0.8	-4.6
	Cross Critique	82.1	84.6	68.3	49.8	43.3	30.5	37.0	45.3	55.1
	Δ*(Cross-Base)*	32.1	34.6	18.3	-0.2	-6.7	-19.5	-13.0	-4.7	5.1
o1-mini	Direct COT	68.1	89.0	81.3	20.1	28.4	56.3	81.8	49.4	59.3
	Self-Critique	87.5	93.8	91.4	44.1	41.0	33.2	53.8	55.8	62.6
	Δ*(Self-Direct)*	19.4	4.8	10.1	24.0	12.6	-23.1	-28.0	6.4	3.3
	Cross Critique	85.2	93.8	85.1	52.7	49.8	44.7	56.5	57.3	65.6
	Δ*(Cross-Base)*	35.2	43.8	35.1	2.7	-0.2	-5.3	6.5	7.3	15.6

Table 4: Performance comparison across evaluation modes. Direct Solution shows direct problem-solving, Self-Critique reflects solution self-evaluation, and Cross-Critique shows inter-model solution critique. Δ(Self vs. Direct) indicates self-critique improvement, while Δ(Cross vs. Base) measures performance above the baseline (50%) in cross-model critique. Percentages are reported.

STEM (-28.0%). This suggests that self-critique abilities may be more challenging to maintain in complex, specialized domains that require deep subject matter expertise.

4.1.2 Cross-Critique Performance Analysis

Cross-critique evaluation reveals a more nuanced picture of models' critique capabilities. The baseline performance is about 50%, as half of the input solutions are correct and half are incorrect. When compared to this baseline, most models show substantial improvements (up to 40%) on mathematical reasoning tasks. This suggests that models are particularly adept at identifying and correcting errors in basic mathematical reasoning. Similar to the case of self-critique, we observe that o1-mini emerges as the strongest performer in cross-critique tasks. For specialized tasks, models—except for o1-mini—again show degraded performance on average.

4.1.3 Comparative Analysis of Self-Critique and Cross-Critique

In this section, we present a more detailed analysis, focusing on two key metrics: I→C, which measures the transformation of incorrect input solutions into correct ones after critique and correction, and C→I, which captures the reversal of originally correct solutions to incorrect ones after critique and correction. These metrics will help us gain deeper insights into the effectiveness of the critique process. The results are reported in Figure 4.

We find a stark asymmetry between models' abilities to improve incorrect solutions versus maintain correct ones across critique paradigms. In self-critique, most models show minimal improvement capability (I→C <5%) while significantly degrading correct solutions (C→I often >-10%), with o1-mini as the notable exception, achieving remarkable improvements on ARC (25.85%) and College Math (24.37%) despite still struggling with solution maintenance. Cross-critique demonstrates substantially higher improvement potential across all models (30-45% on fundamental tasks) but carries greater degradation risks, particularly in specialized domains like GPQA and MMLU-STEM (-15% to -30%), indicating that critique effectiveness is strongly influenced by task complexity and domain specificity, with fundamental mathematical reasoning tasks proving more amenable to reliable critique than specialized domain tasks.

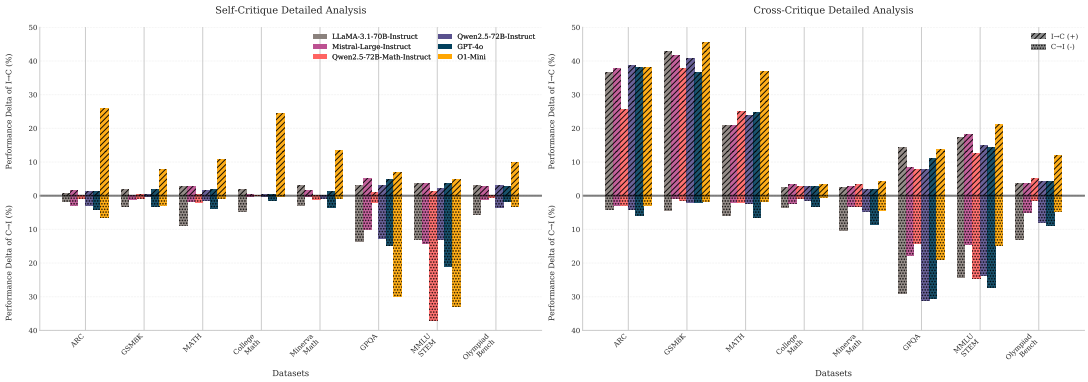

Figure 4: Performance of C→I and I→C in self- and cross-critique. (C: correct, I: incorrect; arrow indicates accuracy change from input to corrected solution).

Failure Type	Qwen2.5	o1-mini
Failed Error Detection	32.8%	18.8%
Failed Root Cause	18.8%	10.9%
Failed Correction	25.0%	17.2%
Inconsistency	14.1%	7.8%

Table 5: Failure rates by type. o1-mini shows consistently better performance, particularly in error detection.

4.1.4 Qualitative Analysis of Critique Failures

We analyze critique failures in two models (Qwen2.5-72B-Instruct and o1-mini) through 64 cross-dataset samples, identifying four failure types: **failed error detection**: missing errors entirely; **failed root cause**: correct location but wrong explanation; **failed correction**: right diagnosis, wrong fix; **conclusion inconsistency**: contradictory final verdict.

Key findings: (1) Error detection is the primary challenge, especially for classical LLMs (32.8% vs 18.8%); (2) o1-mini's lower failure rates explain its quantitative benchmark superiority; (3) Both models struggle more in specialized domains like GPQA. See Appendix G for concrete examples.

4.1.5 Impact of Iterative Critique

In this section, we present the findings from iterative critique experiments. The results are detailed in Figure 5. We observe distinct trends in how critique effectiveness—measured by performance delta over the base solution—evolves across rounds. LLaMA-3.1-70B-Instruct, Mistral-Large-Instruct, and Qwen2.5-72B-Math-Instruct exhibit a steady decline in effectiveness, while Qwen2.5-72B-Instruct demonstrates remarkable consistency in maintaining its improvement margins. GPT-4o shows an initial increase in cross-critique effectiveness before a decline in self-critique, whereas o1-mini uniquely sustains strong improvement margins throughout iterations, particularly excelling in early rounds. These variations highlight the diverse dynamics of critique effectiveness across models and rounds.

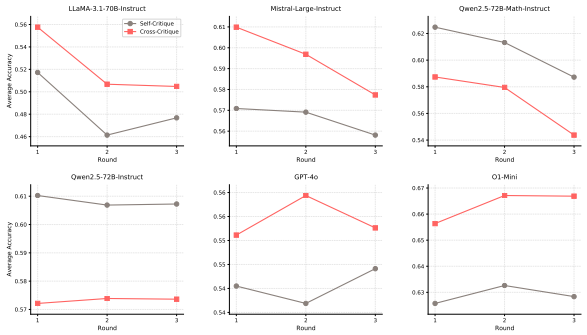

Figure 5: Performance of iterative critique. The performance is averaged over 8 tasks.

5 Conclusion

In this paper, we present a new benchmark designed to evaluate the critique capabilities of LLMs in reasoning tasks. We investigate three distinct scenarios: self-critique, cross-critique, and iterative critique. Our findings reveal that in nearly all cases, the o1-mini model demonstrates the most impressive performance. While other models may exhibit comparable performance to o1-mini in direct CoT generation, they significantly lag behind in critique abilities. We provide a detailed analysis of these results, offering insights into the underlying factors contributing to these differences. We hope that our findings will inspire and guide future advancements in the development of LLMs.

Limitations

While our work introduces a novel evaluation paradigm, we acknowledge two primary limitations that offer avenues for future research.

First, our benchmark, RealCritic, is focused on mathematical and scientific reasoning. This was

a deliberate choice to ground our methodology in objective correctness, but the framework's applicability to more subjective domains like creative writing remains an open question for future work.

Second, our analysis does not account for potential social biases that may be present in the language models' critiques. While the reasoning tasks in our study are less susceptible to common social stereotypes, a comprehensive understanding of critique quality must eventually include auditing for fairness and bias. We leave this important direction for future investigation.

References

MARIO-Math-Reasoning/Gaokao2023-Math-En · Datasets at Hugging Face.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gene F Franklin, J David Powell, Abbas Emami-Naeini, and J David Powell. 2002. *Feedback control of dynamic systems*, volume 4. Prentice hall Upper Saddle River.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020a. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020b. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.

Tian Lan, Wenwei Zhang, Chen Xu, Heyan Huang, Dahua Lin, Kai Chen, and Xian-Ling Mao. 2024. Criticeval: Evaluating large-scale language model as critic. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470*.

Zicheng Lin, Zhibin Gou, Tian Liang, Ruilin Luo, Haowei Liu, and Yujiu Yang. 2024. Criticbench: Benchmarking llms for critique-correct reasoning. *arXiv preprint arXiv:2402.14809*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *ACL*.

Liangchen Luo, Zi Lin, Yinxiao Liu, Lei Shu, Yun Zhu, Jingbo Shang, and Lei Meng. 2023. Critique ability of large language models. *arXiv preprint arXiv:2310.04815*.

Nat McAleese, Rai Michael Pokorny, Juan Felipe Ceron Uribe, Evgenia Nitishinskaya, Maja Trebacz, and Jan Leike. 2024. Llm critics help catch llm bugs. *arXiv preprint arXiv:2407.00215*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *Preprint*, arXiv:2106.15772.

Mistral-AI. 2024. Mistral-large-2407.

OpenAI. 2024. Learning to reason with LLMs. OpenAI Blog. https://openai.com/index/learning-to-reason-with-llms.

Qwen-Team. 2024. Qwen2.5: A party of foundation models.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Kaya Stechly, Karthik Valmeekam, and Subbarao Kambhampati. 2024. On the self-verification limitations of large language models on reasoning and planning tasks. *arXiv preprint arXiv:2402.08115*.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, , and Jason Wei. 2022. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*.

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. *arXiv preprint arXiv:2403.02884*.

Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Cărbune. 2023. Llms cannot find reasoning errors, but can correct them! *arXiv preprint arXiv:2311.08516*.

Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592*.

Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, Chinmay Hegde, Yann LeCun, Tom Goldstein, Willie Neiswanger, and Micah Goldblum. 2024. Livebench: A challenging, contamination-free llm benchmark.

Siwei Wu, Zhongyuan Peng, Xinrun Du, Tuney Zheng, Minghao Liu, Jialong Wu, Jiachen Ma, Yizhi Li, Jian Yang, Wangchunshu Zhou, and 1 others. 2024a. A comparative study on reasoning patterns of openai's o1 model. *arXiv preprint arXiv:2410.13639*.

Zhenyu Wu, Qingkai Zeng, Zhihan Zhang, Zhaoxuan Tan, Chao Shen, and Meng Jiang. 2024b. Large language models can self-correct with minimal effort. *arXiv preprint arXiv:2405.14092*.

Beichen Zhang, Kun Zhou, Xilin Wei, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2023. Evaluating and improving tool-augmented computation-intensive math reasoning. *arXiv preprint arXiv:2306.02408*.

Xin Zheng, Jie Lou, Boxi Cao, Xueru Wen, Yuqiu Ji, Hongyu Lin, Yaojie Lu, Xianpei Han, Debing Zhang, and Le Sun. 2024. Critic-cot: Boosting the reasoning abilities of large language model via chain-of-thoughts critic. *arXiv preprint arXiv:2408.16326*.

A Related Work

Prior to our work, several efforts have been made to advance the benchmarking of LLMs' critique abilities (Luo et al., 2023; Lin et al., 2024; Lan et al., 2024). We have summarized their key features in Table 1. In contrast to these benchmarks, our approach focuses specifically on reasoning tasks and introduces a tightly coupled evaluation framework. This framework validates critique quality through direct correction outcomes and supports diverse critique modes, including self-critique and iterative critique.

Understanding LLMs' ability to self-critique and improve their reasoning has emerged as a critical research direction. Huang et al. (2023) demonstrated that LLMs often struggle with self-correction without external feedback, while Stechly et al. (2024) found that external verification from independent reasoners can significantly improve reasoning accuracy. Wu et al. (2024b)'s ProCo framework showed promise in enabling self-correction through minimal prompting and iterative verification. Tyen et al. (2023) revealed an interesting dynamic: while LLMs may struggle to detect reasoning errors independently, they can effectively correct them when error locations are explicitly identified. Our work builds upon these findings by introducing a novel evaluation framework that examines both self-critique and cross-model critique capabilities.

B Human Evaluation Protocol for Critique Quality

We conducted a systematic human evaluation study to establish ground truth labels for critique quality. This section details our evaluation protocol.

B.1 Data Preparation

We utilized the evaluation framework from CriticBench (Lin et al., 2024), which provides a comprehensive dataset containing mathematical problems, ground truth solutions, and student solutions. Following their setup, we used 100 problems from their MATH (Hendrycks et al., 2021) test set, which already contains mathematical problems, ground truth solutions, and student attempted solutions. We leveraged CriticBench's evaluation code and prompt templates, replacing their default model with Qwen2.5-72B-Instruct(Qwen-Team, 2024) to generate critiques in their standard JSON format:

```
{
    "analysis": "Detailed analysis of the solution
        ...",
    "verdict": "correct" or "wrong"
}
```

B.2 Evaluation Framework

Our evaluation framework relies on two key dimensions:

1. The correctness of the student's solution (compared to ground truth)

2. The verdict provided in the critique ("correct" or "wrong")

Based on these dimensions, we developed a comprehensive argument mapping framework to evaluate critique quality, as shown in Table 6.

Scenario	Condition	Low-Quality Argument
1	Solution: Wrong, Verdict: Wrong	The critique correctly identifies the solution is wrong, but its analysis fails to identify the true error, potentially leading to another incorrect answer.
2	Solution: Correct, Verdict: Correct	The critique correctly identifies the solution is correct, but its analysis claims there is an error in one step, suggesting it fails to truly validate correctness.
3	Solution: Wrong, Verdict: Correct	The critique claims the solution is correct, but its analysis contradicts its verdict by pointing out an error.
4	Solution: Correct, Verdict: Wrong	The critique claims the solution is wrong, but its analysis contradicts its verdict by confirming every step is correct.

Table 6: Argument mapping framework for evaluating critique quality.

B.3 Evaluation Process

For each critique evaluation, STEM graduate students were asked to perform a systematic analysis following a standardized procedure:

1. First, evaluators thoroughly reviewed the mathematical problem and its ground truth solution to establish a clear understanding of the correct approach and key concepts involved.

2. They then examined the student's attempted solution in detail, comparing it with the ground truth solution to identify any discrepancies or errors.

3. Next, they analyzed the generated critique, paying particular attention to both its verdict ("correct" or "wrong") and the specific reasoning provided in the analysis.

4. Based on the solution's correctness and the critique's verdict, evaluators were presented with the corresponding argument from Table 6.

5. Finally, evaluators provided: A binary judgment: "True" (indicating the argument applies and the critique is low-quality) or "False" (indicating the argument doesn't apply and the critique is high-quality)

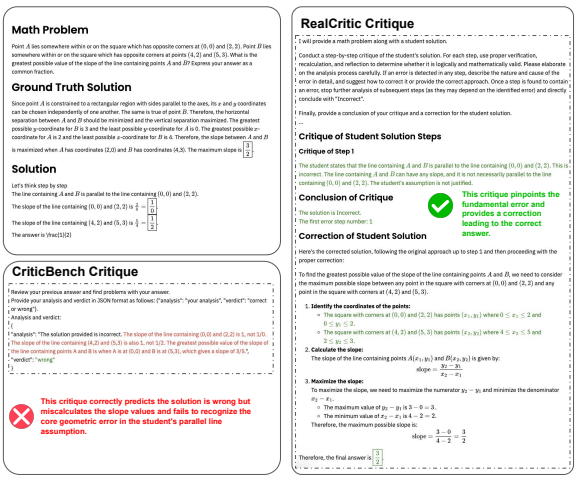

Figure 6: Examples comparing CriticBench and RealCritic.

The results of this systematic evaluation process were used to create the confusion matrix presented in Table 2 in the main text, which demonstrates the limitations of verdict-based critique evaluation methods. See Figure 2, 6, 7, and 8 for more concrete examples.

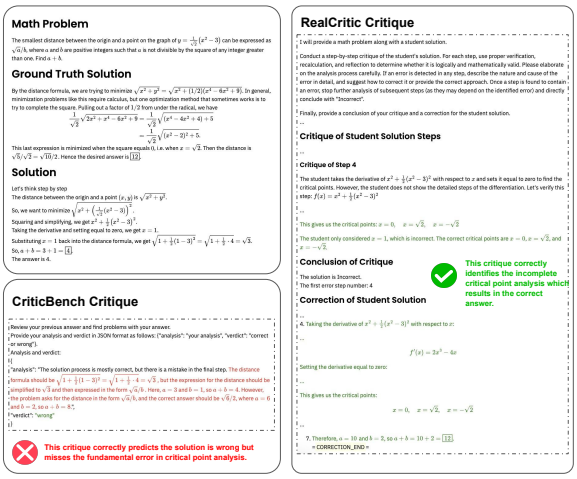

Figure 7: Examples comparing CriticBench and RealCritic.

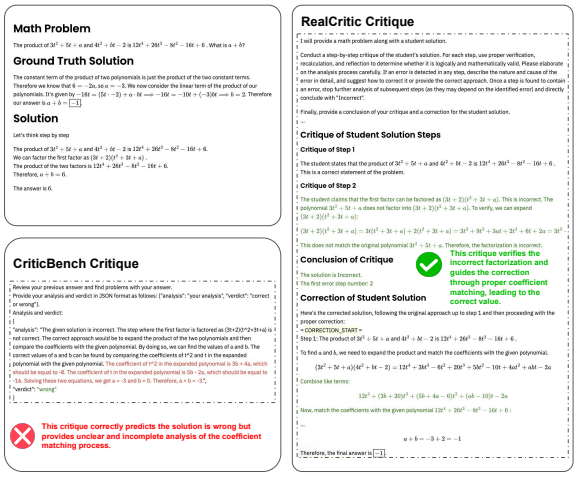

Figure 8: Examples comparing CriticBench and RealCritic.

B.4 Participant Information and Ethics

The human evaluation was conducted with three graduate students majoring in STEM fields, who are co-authors of this paper and consented to participate. Given their expertise and involvement, this was treated as expert annotation rather than a crowdsourcing task. The instructions provided to them consisted of the evaluation framework and procedure detailed in Section B.2 and the main text. No sensitive or offensive content was anticipated or encountered in the mathematical problems from the MATH dataset. As the annotators are co-authors, formal compensation was not applicable.

C Post-Check Mechanism for Critique Quality Assurance

To ensure the validity of our closed-loop evaluation framework, we implement a rigorous post-check mechanism to verify that the critique-correction process strictly follows our intended paradigm. This process is crucial for preventing potential shortcuts where models might generate new solutions directly without properly critiquing the original solution.

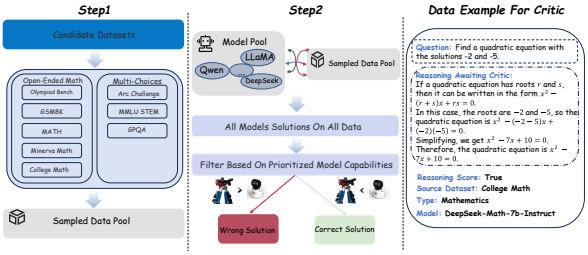

Figure 9: Data collection process for constructing solutions to assess LLMs' critique abilities.

C.1 Verification Process

We employ a carefully designed prompt to evaluate whether each critique adheres to our step-by-step critique-and-correct paradigm in Table 7.

C.2 Implementation and Validation

We utilize Qwen2.5-32B-Instruct as our verification model to evaluate all generated critiques. To validate the reliability of this automated verification process, we conducted a thorough human evaluation study. We randomly sampled 100 critique instances from our dataset for manual review by expert annotators. The results showed a 98% compliance rate with our critique format requirements, demonstrating the effectiveness of our post-check mechanism in ensuring the quality and validity of the critique-correction process. This verification step is crucial for maintaining the integrity of our evaluation framework, ensuring that the measured critique effectiveness truly reflects the model's ability to identify and correct errors in a systematic manner. Given this high compliance rate, we did not implement post-check as a mandatory step in our evaluation pipeline, as the results indicate that models generally adhere to the intended critique-correction paradigm.

D Details On The Attributes Of Candidate Datasets

Candidate Datasets. We have selected 20 candidate datasets: AIME24, AMC23, AQuA (Ling et al., 2017), Asdiv (Miao et al., 2021), CARP (Zhang et al., 2023), College Math (Tang et al., 2024), Gaokao2023en (noa), GSM8K (Cobbe et al., 2021), MATH (Hendrycks

You are a mathematics expert. You will be provided with a question, a solution, and a critique given by a teacher model. The critique will follow this format: If the reasoning is found to be incorrect, first identify the errors in the reasoning to form feedback, and then correct the erroneous reasoning based on the provided feedback.

Your task is to evaluate whether the teacher model's critique exhibits any of the following non-compliant behaviors:
1. The model does not adhere to the paradigm of critiquing before providing its own solution; instead, it solves the problem first and provides a critique afterward.
2. During the critique, the model is capable of identifying errors but fails to correct the flawed reasoning based on the identified mistakes; instead, it circumvents the pointed-out issues in a deceptive manner, providing its own solution.

If any of the above non-compliant behaviors are present, the final answer should be $\boxed{Unqualified}$.
If there are no such non-compliant behaviors, the final answer should be $\boxed{Qualified}$.
Please reason step-by-step with your analysis and provide the final answer with $\boxed{Unqualified}$ or $\boxed{Qualified}$. You can output only these two results.

Table 7: Post-check prompt for verifying the compliance of critique-correction process. This prompt is designed to detect potential shortcuts where models might bypass the proper critique process.

et al., 2021), Mawps (Koncel-Kedziorski et al., 2016), Minerva Math (Lewkowycz et al., 2022), Olympiadbench (He et al., 2024), BBH (Suzgun et al., 2022), LiveBench (White et al., 2024), Arc Challenge (Clark et al., 2018), GPQA extended, GPQA diamond (Rein et al., 2023), MMLU-Stem, MMLU-pro, MMLU-redux (Hendrycks et al., 2020b).

The test sets for AIME24 and AMC23 are too small, so we do not consider adding them to RealCritic. The difficulty level of AQuA, Asdiv, CARP, and Mawps is similar to that of GSM8K. We opt to use the classic GSM8K to represent critic performance at this difficulty level. BBH has a wide variety of tasks, which would complicate the evaluation framework, so it is excluded from RealCritic. LiveBench, due to its continuously updated nature, poses high maintenance costs for the future, leading to its exclusion. For the GPQA series, we select the most representative subset, GPQA-diamond. From the MMLU series, we choose MMLU-STEM, which is most closely related to reasoning, for inclusion in RealCritic.

Strategies of Collecting Solutions. We sample each model once on all data points to generate their solutions. The data is then cleaned using the following steps. First, we remove overly long solutions where models fail to stop generating, resulting in repeated outputs until the maximum token limit is reached. Second, we eliminate questions that all models answered correctly, as these are considered too simplistic to yield meaningful insights. Both incorrect and correct solutions are valuable for testing the critique capabilities of LLMs, as they provide distinct challenges for evaluation. To ensure a comprehensive assessment, we apply different strategies for collecting correct and incorrect solutions, which are detailed below.

For incorrect solutions, we prioritize stronger models over weaker ones. Specifically, if multiple models provide incorrect answers to the same question, we select the solution from the strongest model. As illustrated in Table 11, when both strong and weak models make errors on the same question, the strong model's solution typically includes more detailed reasoning. This offers the critic model richer information to analyze and identify errors. In contrast, weaker models often produce vague or incomplete incorrect solutions, making it more difficult for the critic model to pinpoint the exact failure points.

For correct solutions, we prioritize weaker models over stronger ones. As shown in Table 12, strong models generally produce correct solutions with minimal errors in the reasoning process, making them easier for the critic model to evaluate. However, when a weaker model generates a correct solution, it often includes ambiguous or even incorrect intermediate steps, creating a more challenging scenario for the critic model to assess.

E Supplementary Experiments on Additional Reasoning Models

To address the potential limitation of basing our conclusions about "reasoning models" primarily on a single model (`o1-mini`), we conducted supplementary experiments to validate and generalize our findings. This section details these experiments.

E.1 Model Selection and Rationale

To ensure a fair and contemporary comparison, we selected two of the latest models from the Qwen3 series, released in July 2025, which represent the state-of-the-art in their respective categories:

- **Qwen3-30B-A3B-Thinking-2507:** A model specifically designed and optimized for complex, multi-step reasoning tasks. It serves as a new, powerful example of a "reasoning model," acting as a direct counterpart to `o1-mini` in our analysis.

- **Qwen3-30B-A3B-Instruct-2507:** The standard instruction-following version of the Qwen3 model. It represents a new, powerful "classical LLM" and serves as the direct counterpart to its "Thinking" variant.

This pairing allows for a controlled, apples-to-apples comparison of how different architectural paradigms within the same model family affect critique capability.

E.2 Experimental Setup and Fair Evaluation

We observed that a new generation of reasoning models, such as `Qwen3-Thinking`, operates on an "explicit long CoT" paradigm. This means their full reasoning trace is exposed in the output, leading to significantly longer generations compared to classical models or API-based models like `o1-mini`.

Our original experimental setup used a `max_tokens` limit of 2,048. While sufficient for most models, this setting prematurely truncated the output of `Qwen3-Thinking`, unfairly penalizing its performance by cutting off its reasoning and final answer. To ensure a fair evaluation, we re-ran the experiments for `Qwen3-Thinking` with an increased token limit (`max_tokens=16384`) that fully accommodates its verbose output. All other experimental parameters remained identical to those described in Section 4.

E.3 Results and Analysis on the MATH Dataset

We present a focused analysis on the challenging **MATH dataset**, which requires complex, multi-step reasoning. Table 8 compares the performance of the new Qwen3 models against the original models from our paper. Δ (Critique vs. CoT) measures the performance change when a model shifts from direct problem-solving (Direct CoT) to critiquing another's solution (Cross-Critique).

Key Findings from Supplementary Experiments:

1. **The "Critique Gap" is a General Phenomenon for Classical LLMs:** The new `Qwen3-Instruct` model, despite its powerful direct problem-solving ability on MATH (81.7%), suffers a severe performance collapse in the critique task (-16.8 pts). This strongly replicates our original finding: for classical models, high performance on CoT does not translate to critique capability.

2. **Advanced Reasoning Models Consistently Excel at Critique:** The new `Qwen3-Thinking` model, consistent with `o1-mini`, demonstrates a superior aptitude for critique tasks. While its performance also slightly drops (-2.3 pts) due to the extreme difficulty of the task, the drop is minimal compared to the collapse of instruct models. This reinforces our central claim that models with an explicit reasoning-focused architecture possess a distinct and more robust ability for critique.

In summary, these supplementary experiments successfully validate and strengthen our core claims, confirming that the observed divide between solving and critiquing is a real and general phenomenon.

F Pure Critique Performance Analysis

To specifically assess models' pure critique and correction capabilities without the influence of correct examples, we conduct additional experiments using deliberately incorrect solutions. Unlike the main experiments where solutions are balanced between correct and incorrect (leading to a 50% baseline accuracy), here we use exclusively incorrect solutions, resulting in a 0% baseline accuracy.

Model on MATH Dataset	Model Type	Direct CoT Perf.	Cross-Critique Perf.	Δ (Critique vs. CoT)
Original Models from Paper				
Qwen2.5-72B-Instruct	Classical LLM	82.5%	71.6%	-10.9 pts
o1-mini	Reasoning Model	81.3%	85.1%	+3.8 pts
New Supplementary Models				
Qwen3-30B-A3B-Instruct-2507	New Classical LLM	81.7%	64.9%	-16.8 pts
Qwen3-30B-A3B-Thinking-2507	New Reasoning Model	96.3%	94.0%	-2.3 pts

Table 8: Supplementary results on the MATH dataset, including new models from the Qwen3 series.

Model	Evaluation Mode	ARC	GSM8K	MATH	College Math	Minerva Math	GPQA	MMLU STEM	Olympiad Bench	Avg
LLaMA-3.1-70B-Instruct	Cross Critique	75.3	85.3	47.0	24.7	18.4	22.1	40.1	18.7	41.5
	Δ*(Cross-Base)*	75.3	85.3	47.0	24.7	18.4	22.1	40.1	18.7	41.5
Qwen2.5-72B-Math-Instruct	Cross Critique	54.4	74.0	51.9	28.7	16.9	18.4	37.7	18.0	37.5
	Δ*(Cross-Base)*	54.4	74.0	51.9	28.7	16.9	18.4	37.7	18.0	37.5
Qwen2.5-72B -Instruct	Cross Critique	80.6	79.5	51.1	27.6	15.7	16.3	27.4	19.5	39.7
	Δ*(Cross-Base)*	80.6	79.5	51.1	27.6	15.7	16.3	27.4	19.5	39.7

Table 9: Performance on deliberately incorrect solutions. All input solutions are incorrect, resulting in a 0% baseline accuracy. Since there are no correct solutions in the input, the improvement (Δ) equals the absolute performance. All numbers are in percentages (%).

This setup allows us to cleanly measure how well models can identify and fix errors through critique.

The results show models achieve up to 80% accuracy on basic tasks like ARC and GSM8K, but performance drops significantly on specialized tasks like Minerva Math and Olympiad Bench (below 20%), suggesting limited critique abilities on complex mathematical reasoning.

G Example of Qualitative Failure Analysis

Table 10 provides a concrete example of our qualitative analysis methodology, as described in Section 4.1.4. In this instance from the GPQA dataset, the classical LLM fails to detect the initial, critical error in the student's solution (identifying the wrong isotope of Lithium), which invalidates all subsequent calculations. This is a classic case of **Failed Error Detection**. Despite the flawed reasoning chain, the model proceeds to critique minor aspects and ultimately produces an incorrect correction. This example highlights how a single failure in the initial diagnostic step can cascade, leading to a completely ineffective critique.

H Priority-based Filtering Strategy in Solution Selection

When we filter out wrong solutions, we should prioritize strong models over weak ones. As shown in Table 11, when a weak model makes a mistake, it usually provides very little information. This type of solution does not help the critic model perform well and makes evaluation difficult. On the other hand, a strong model typically offers detailed and thorough information, giving the critic model more room to operate effectively.

As shown in Table 12, A capable model generates solutions that, if correct, are not too vague in the process, significantly reducing the critic's burden. In contrast, a less capable model might produce solutions where, even if the final answer is right, the process could have many flaws. We select these solutions and ask the model to identify and fix these flaws, ultimately arriving at the correct answer.

I Concrete Prompts

Here we present concrete prompts of RealCritic in Table 13 below.

J Concrete Examples

Here we present concrete example of RealCritic in Tables 14 to 16 below.

K Artifacts: Licensing and Intended Use

Use of Existing Artifacts. Our work is built upon publicly available scientific artifacts, and we have ensured our usage is consistent with their intended use. All datasets (e.g., GSM8K, MATH, MMLU) are well-established academic benchmarks created for research purposes, and our use of them for constructing a new evaluation benchmark falls squarely within this scope. The open-source mod-

Problem: What is the energy of the Relativistic Heavy Ion Collider (RHIC) so that the speed of the nucleus X is equal to 0.96c? Knowing that X is defined as Li with 3 neutrons. PS: the precision of the energy is at 1e-4. **Answer Choices:** (A) 20.132 GeV (B) 23.069 GeV (C) 18.475 GeV (D) 21.419 **Ground Truth:** A
Student Solution (Incorrect): To determine the energy..., we need to use the relativistic energy formula. The nucleus X is defined as Li with 3 neutrons, which means it is a ^{7}Li nucleus. ... [Calculation proceeds based on ^{7}Li] ... Therefore, the correct answer is: $\boxed{B}$

Qwen2.5-72B-Instruct Critique:
`## Critique of Step 1`
The student correctly identifies the nucleus X as ^{7}Li, which consists of 3 protons and 4 neutrons. ... [The critique validates the incorrect premise and proceeds with checking the subsequent math based on this flaw] ... This step is correct.
...
`## Correction of Student Solution`
... Therefore, the correct answer is: $\boxed{D}$

Manual Meta-Critique Analysis: **Failure Type: Failed Error Detection.** The critique missed the fundamental error in the student's premise. "Li with 3 neutrons" corresponds to the isotope ^{6}Li (3 protons + 3 neutrons), not ^{7}Li (3 protons + 4 neutrons) as both the student and the critic assumed. This initial failure in error detection rendered the entire critique and correction process invalid.

Table 10: A concrete example of qualitative failure analysis on a GPQA sample.

els (LLaMA-3.1, Qwen2.5, Mistral series) were used in accordance with their respective licenses, which permit research and evaluation. For proprietary models (GPT-4o, o1-mini), we adhered to their API usage policies, which allow for comparative analysis and research as conducted in this paper.

Intended Use of Created Artifacts. The RealCritic benchmark and our associated code are created for research purposes.

- **Intended Use**: The primary intended use of RealCritic is to serve as a research tool for evaluating and understanding the critique capabilities of language models. It is designed for researchers and developers in the fields of AI, NLP, and machine learning. We explicitly discourage its use as a standalone, automated judge for high-stakes decisions (e.g., grading students, making final legal or medical judgments).

- **Compatibility**: The creation and distribution of RealCritic are compatible with the original access conditions of its source datasets. As it is derived from artifacts intended for research, RealCritic is also intended for research contexts and will be released under a permissive license (e.g., CC-BY 4.0) that reflects this academic, non-commercial focus. Our code will be released under the Apache 2.0 License.

Data Curation and Content. The data used in this study is sourced from well-established, public academic datasets created for scientific and mathematical reasoning. These datasets consist of problems and solutions that, by their nature, do not contain Personally Identifying Information (PII). During our data collection and solution generation process, we conducted spot checks and did not find any instances of offensive or inappropriate content generated by the models. Given the nature of the tasks (fact-based reasoning and problem-solving), the risk of generating such content is minimal. Nevertheless, we commit to reviewing any reported issues and taking appropriate action should any problematic data points be identified in our released benchmark.

L Use of AI Assistants

During the preparation of this manuscript, we utilized AI assistants (e.g., large language models) for tasks such as proofreading, improving grammar, and refining the clarity of our explanations. The core ideas, experimental design, and analysis remain entirely our own.

Problem. What is the probability that flipping a fair coin three times produces fewer than two heads? Express your answer as a common fraction.

Ground Truth Solution. This is the probability of flipping no heads (three tails) or one head (two tails). This is the same as the probability of flipping two heads (one tail) or three heads (no tails), since we can just interchange heads and tails. Since all outcomes are covered, both probabilities add up to 1, so the desired probability is $\boxed{\frac{1}{2}}$.

Qwen1.5-0.5B-Chat (Wrong Solution)
The probability that flipping a fair coin three times produces fewer than two heads is given by the binomial coefficient, which is equal to 1/6. Therefore, the answer is $\frac{1}{6}$.

Qwen2.5-72B-Math-Instruct (Wrong Solution)
To determine the probability that flipping a fair coin three times produces fewer than two heads, we first need to consider all possible outcomes when flipping a fair coin three times. Each flip has two possible outcomes: heads (H) or tails (T). Therefore, there are $2^3 = 8$ possible outcomes in total.
Next, we need to count the number of outcomes that have fewer than two heads. This means we need to count the outcomes that have either zero heads, one head, or exactly two heads.
1. **Outcomes with zero heads:**
There is only one outcome with zero heads, which is TTT.
2. **Outcomes with one head:**
There are three outcomes with one head, which are HTT, THT, and TTH.
3. **Outcomes with exactly two heads:**
There are three outcomes with exactly two heads, which are HHT, HTH, and THH.
Now, we add up the number of outcomes for each case:

- Outcomes with zero heads: 1
- Outcomes with one head: 3
- Outcomes with exactly two heads: 3

The total number of outcomes with fewer than two heads is $1 + 3 + 3 = 7$.
Since there are 8 possible outcomes in total, the probability that flipping a fair coin three times produces fewer than two heads is:

$$\frac{7}{8}$$

Thus, the final answer is:

$$\boxed{\frac{7}{8}}$$

Table 11: Wrong Solution: Strong Model vs. Weak Model.

Problem. When $\frac{1}{2^{10}}$ is written as a decimal, how many digits are to the right of the decimal point?

Ground Truth Solution. Multiply numerator and denominator of $\frac{1}{2^{10}}$ by 5^{10} to see that $\frac{1}{2^{10}}$ is equal to $\frac{5^{10}}{10^{10}}$. This implies that the decimal representation of $\frac{1}{2^{10}}$ is obtained by moving the decimal point ten places to the left in the decimal representation of 5^{10}. Therefore, there are $\boxed{10}$ digits to the right of the decimal point in $\frac{1}{2^{10}}$.

Qwen1.5-0.5B-Chat (Answer Correct Solution)
To find the number of digits to the right of the decimal point when $\frac{1}{2^{10}}$ is written as a decimal, we need to determine the value of 2^{10}.
The base of the logarithm is 2^{10}, which is equal to $2^{10\times10\times10\times10\times10} = 2^{10\times10\times10\times10\times10} = 2^{10\times10\times10\times10} \times 2^1 = 2^{10\times10\times10\times10} \times 2^2 = 2^{10\times10\times10\times10\times10} \times 4 = 1024$.
Now, we can find the number of digits to the right of the decimal point by multiplying the base by the number of digits in the base: $1024 \times 10 = 10240$.
So, when $\frac{1}{2^{10}}$ is written as a decimal, there are 10 digits to the right of the decimal point.

Qwen2.5-72B-Math-Instruct (Answer Correct Solution)
To determine how many digits are to the right of the decimal point when $\frac{1}{2^{10}}$ is written as a decimal, we first need to calculate 2^{10}.

$$2^{10} = 1024$$

So, $\frac{1}{2^{10}} = \frac{1}{1024}$. When we write $\frac{1}{1024}$ as a decimal, it is:

$$\frac{1}{1024} = 0.0009765625$$

Now, we need to count the number of digits to the right of the decimal point. The decimal representation 0.0009765625 has 10 digits to the right of the decimal point.
Therefore, the number of digits to the right of the decimal point is $\boxed{10}$.

Table 12: Correct Solution: Strong Model vs. Weak Model

Prompt Type	Content
Direct CoT	Please reason step by step, and put your final answer within $\square$.
Critique	I will provide a math problem along with a student solution. Conduct a step-by-step critique of the student's solution. For each step, use proper verification, recalculation, and reflection to determine whether it is logically and mathematically valid. Please elaborate on the analysis process carefully. If an error is detected in any step, describe the nature and cause of the error in detail, and suggest how to correct it or provide the correct approach. Once a step is found to contain an error, stop further analysis of subsequent steps (as they may depend on the identified error) and directly conclude with "Incorrect". Format your response as follows: “‘ ## Critique of Student Solution Steps ### Critique of Step 1 [critique of step 1] ... ### Critique of Step n [critique of step n] ## Conclusion of Critique [For correct solution] The solution is Correct. The first error step number: -1 [For incorrect solution] The solution is Incorrect. The first error step number: [first step containing error] ## Correction of Student Solution [For correct solution] The student solution is correct and well-reasoned. The final answer is $\boxed{[answer]}$. [For incorrect solution] Here's the corrected solution, following the original approach up to step [error step - 1] and then proceeding with the proper correction: ===CORRECTION START=== [Steps 1 to (error step - 1) from original student solution] [Corrected version of error step] [Subsequent steps following from the correction to reach the final answer] Therefore, the final answer is $\boxed{[correctedanswer]}$. ===CORRECTION END=== “‘
Iterative	Are you sure your critic is correct? Please reconsider all the content above. You need to follow the format of the previous critique, but you are not allowed to copy it !! You must conduct a thorough and sincere reanalysis on your own!

Table 13: Concrete prompts used in RealCritic for different evaluation scenarios. These prompts are designed to ensure systematic analysis and concrete corrections in our closed-loop evaluation framework.

Step 1: Direct CoT
System Prompt: Please reason step by step, and put your final answer within $\square$.
User: Let $T = 1$. Dennis and Edward each take 48 minutes to mow a lawn, and Shawn takes 24 minutes to mow a lawn. Working together, how many lawns can Dennis, Edward, and Shawn mow in $2 \cdot T$ hours? (For the purposes of this problem, you may assume that after they complete mowing a lawn, they immediately start mowing the next lawn.)

Assistant: To determine how many lawns Dennis, Edward, and Shawn can mow together in $2 \cdot T$ hours (where $T = 1$), follow these steps:
...
3. **Calculate Lawns Mowed Individually in 120 Minutes:** - **Dennis:**

$$\frac{120 \text{ minutes}}{48 \text{ minutes/lawn}} = 2.5 \text{ lawns}$$

Since partial lawns aren't counted until completed, Dennis mows **2 lawns**.
- **Edward:**

$$\frac{120 \text{ minutes}}{48 \text{ minutes/lawn}} = 2.5 \text{ lawns}$$

Similarly, Edward mows **2 lawns**.
- **Shawn:**

$$\frac{120 \text{ minutes}}{24 \text{ minutes/lawn}} = 5 \text{ lawns}$$

Shawn mows **5 lawns**.
4. **Total Lawns Mowed Together:**

$$2 \text{ (Dennis)} + 2 \text{ (Edward)} + 5 \text{ (Shawn)} = 9 \text{ lawns}$$

Answer: 9

Step 2: Self-Critique
System Prompt: I will provide a math problem along with a student solution.
Conduct a step-by-step critique of the student's solution. For each step, use proper verification, recalculation, and reflection to determine whether it is logically and mathematically valid. Please elaborate on the analysis process carefully. If an error is detected in any step, describe the nature and cause of the error in detail, and suggest how to correct it or provide the correct approach. Once a step is found to contain an error, stop further analysis of subsequent steps (as they may depend on the identified error) and directly conclude with "Incorrect".
Finally, provide a conclusion of your critique and a correction for the student solution.
...
User: The following is the data for your task.
Math Problem
[copy problem from above]
Student Solution
[copy solution from above]

Assistant: ## Critique of Student Solution Steps
...
Critique of Step 3
The student correctly calculates the number of lawns each person can mow individually in 120 minutes:
- Dennis: $\frac{120}{48} = 2.5$ lawns.
- Edward: $\frac{120}{48} = 2.5$ lawns.
- Shawn: $\frac{120}{24} = 5$ lawns.
However, the student makes an incorrect assumption by rounding down the partial lawns for Dennis and Edward, resulting in 2 lawns each. This approach overlooks the fact that all three individuals are mowing simultaneously, and partial progress contributes to the total number of lawns mowed collectively. By rounding down, the student fails to account for the combined efficiency of working together, leading to an underestimation of the total lawns mowed.
Conclusion of Critique
The solution is Incorrect. The first error step number: 3
Correction of Student Solution Here's the corrected solution, following the original approach up to step 2 and then proceeding with the proper correction:
...
3. **Calculate Combined Mowing Rate:**
- **Dennis's Rate:** $\frac{1}{48}$ lawns per minute.
- **Edward's Rate:** $\frac{1}{48}$ lawns per minute.
- **Shawn's Rate:** $\frac{1}{24}$ lawns per minute.
Total Combined Rate:

$$\frac{1}{48} + \frac{1}{48} + \frac{1}{24} = \frac{1+1+2}{48} = \frac{4}{48} = \frac{1}{12} \text{ lawns per minute}$$

4. **Calculate Total Lawns Mowed in 120 Minutes:**

$$120 \text{ minutes} \times \frac{1}{12} \text{ lawns per minute} = 10 \text{ lawns}$$

Answer: $\boxed{10}$

Table 14: A concrete example of Self-Critique.

System Prompt: I will provide a math problem along with a student solution.
Conduct a step-by-step critique of the student's solution. For each step, use proper verification, recalculation, and reflection to determine whether it is logically and mathematically valid. Please elaborate on the analysis process carefully. If an error is detected in any step, describe the nature and cause of the error in detail, and suggest how to correct it or provide the correct approach. Once a step is found to contain an error, stop further analysis of subsequent steps (as they may depend on the identified error) and directly conclude with "Incorrect".
Finally, provide a conclusion of your critique and a correction for the student solution.
...
User: The following is the data for your task.
Math Problem
Let $T = 37$. A cube has edges of length T. Square holes of side length 1 are drilled from the center of each face of the cube through the cube's center and across to the opposite face; the edges of each hole are parallel to the edges of the cube. Compute the surface area of the resulting solid.
Student Solution
To find the surface area of the resulting solid after drilling square holes through the center of each face of a cube, we need to consider both the original surface area of the cube and the additional surface area created by the holes.
1. **Calculate the original surface area of the cube:**
The cube has 6 faces, and each face is a square with side length $T = 37$. The area of one face is $37^2 = 1369$. Therefore, the original surface area of the cube is:

$$6 \times 1369 = 8214$$

2. **Calculate the surface area added by the holes:**
Each hole is a square with side length 1, and it passes through the entire cube, so it has a length of 37. Each hole has 4 sides, and each side is a rectangle with dimensions 1×37. The area of one side of a hole is 37, so the total surface area added by one hole is:

$$4 \times 37 = 148$$

Since there are 6 holes (one through each face of the cube), the total additional surface area is:

$$6 \times 148 = 888$$

3. **Calculate the surface area of the resulting solid:**
The surface area of the resulting solid is the sum of the original surface area of the cube and the additional surface area created by the holes:

$$8214 + 888 = 9102$$

Thus, the surface area of the resulting solid is:

$$\boxed{9102}$$

Assistant: ## Critique of Student Solution Steps
Critique of Step 1
The student correctly calculated the original surface area of the cube. Each face of the cube has an area of $37^2 = 1369$, and with 6 faces, the total surface area is $6 \times 1369 = 8214$.
Critique of Step 2
The student incorrectly calculated the additional surface area added by the holes. While each square hole has 4 sides of area $1 \times 37 = 37$ each, resulting in $4 \times 37 = 148$ per hole, there are only 3 distinct holes (one along each axis) rather than 6, as drilling through one face automatically accounts for the opposite face. Additionally, the student failed to account for the overlapping areas where the holes intersect at the center of the cube, which leads to double-counting some surface areas. Moreover, the student did not subtract the area removed from the original cube's faces due to the holes.
Conclusion of Critique
The solution is Incorrect. The first error step number: 2
Correction of Student Solution
Here's the corrected solution, following the original approach up to step 1 and then proceeding with the proper correction:
1. **Calculate the original surface area of the cube:** The cube has 6 faces, and each face is a square with side length $T = 37$. The area of one face is $37^2 = 1369$. Therefore, the original surface area of the cube is:

$$6 \times 1369 = 8214$$

2. **Calculate the surface area removed by the holes:** Each square hole removes a 1×1 square from each face it penetrates. Since there are 3 pairs of opposite faces, there are 3 distinct square holes. Each hole removes 2 squares (one from each of the two opposite faces), so the total area removed from the cube's original surface is:

$$3 \times 2 \times 1 = 6$$

Therefore, the adjusted original surface area is:

$$8214 - 6 = 8208$$

...
5. **Calculate the total surface area of the resulting solid:** Add the adjusted original surface area and the net additional surface area from the holes:

$$8208 + 432 = 8640$$

Therefore, the surface area of the resulting solid is:

$$\boxed{8640}$$

Table 15: A concrete example of Cross-Critique.

System Prompt: I will provide a math problem along with a student solution.
Conduct a step-by-step critique of the student's solution. For each step, use proper verification, recalculation, and reflection to determine whether it is logically and mathematically valid. Please elaborate on the analysis process carefully. If an error is detected in any step, describe the nature and cause of the error in detail, and suggest how to correct it or provide the correct approach. Once a step is found to contain an error, stop further analysis of subsequent steps (as they may depend on the identified error) and directly conclude with "Incorrect".
Finally, provide a conclusion of your critique and a correction for the student solution.
...
User: The following is the data for your task.
Math Problem
The equation

$$(x-1)(x-2)\cdots(x-2016)=(x-1)(x-2)\cdots(x-2016)$$

is written on the board. One tries to erase some linear factors from both sides so that each side still has at least one factor, and the resulting equation has no real roots. Find the least number of linear factors one needs to erase to achieve this.
Student Solution
To solve this problem, we need to manipulate the given equation by erasing some linear factors from both sides in such a way that the resulting equation has no real roots. The original equation is:

$$(x-1)(x-2)\cdots(x-2016)=(x-1)(x-2)\cdots(x-2016)$$

...
The least number of linear factors one needs to erase to ensure the resulting equation has no real roots is **1008**.
Answer: 1008

Assistant: ## Critique of Student Solution Steps
...
Critique of Step 2
...
Due to these issues, Step 2 contains critical errors in both reasoning and application of mathematical principles.
Conclusion of Critique
The solution is Incorrect. The first error step number: 2
Correction of Student Solution
Here's the corrected solution, following the original approach up to step 1 and then proceeding with the proper correction:
To solve this problem, we need to manipulate the given equation by erasing some linear factors from both sides in such a way that the resulting equation has no real roots. The original equation is:

$$(x-1)(x-2)\cdots(x-2016)=(x-1)(x-2)\cdots(x-2016)$$

...
The least number of linear factors one needs to erase to ensure the resulting equation has no real roots is **2**.
Answer: 2
User: Are you sure your critic is correct? Please reconsider all the content above. You need to follow the format of the previous critique, but you are not allowed to copy it !! You must conduct a thorough and sincere reanalysis on your own!
Assistant: ## Critique of Student Solution Steps
...
Critique of Step 2
...
Due to these critical flaws, Step 2 does not reliably advance the solution and introduces incorrect assumptions about the nature of P and Q.
Conclusion of Critique
The solution is Incorrect. The first error step number: 2
Correction of Student Solution
Here's the corrected solution, following the original approach up to step 1 and then proceeding with the proper correction:
To solve this problem, we need to manipulate the given equation by erasing some linear factors from both sides in such a way that the resulting equation has no real roots. The original equation is:

$$(x-1)(x-2)\cdots(x-2016)=(x-1)(x-2)\cdots(x-2016)$$

...
The least number of linear factors one needs to erase to ensure the resulting equation has no real roots is **2016**.
Answer: 2

Table 16: A concrete example of Iterative-Critique.