

AutoRubric: Rubric-Based Generative Rewards for Faithful Multimodal Reasoning

Mengzhao Jia¹, Zhihan Zhang¹, Ignacio Cases², Zheyuan Liu¹,
Meng Jiang¹, Peng Qi²

¹University of Notre Dame, ²Uniphore
mjia2@end.edu, peng.qi@uniphore.com

Abstract

Multimodal large language models (MLLMs) have rapidly advanced from perception tasks to complex multi-step reasoning, yet reinforcement learning with verifiable rewards (RLVR) often leads to spurious reasoning since only the final-answer correctness is rewarded. To address this limitation, we propose AutoRubric, a framework that integrates RLVR with process-level supervision through automatically collected rubric-based generative rewards. Our key innovation lies in a scalable self-aggregation method that distills consistent reasoning checkpoints from successful trajectories, enabling problem-specific rubric construction without human annotation or stronger teacher models. By jointly leveraging rubric-based and outcome rewards, AutoRubric achieves state-of-the-art performance on six multimodal reasoning benchmarks and substantially improves reasoning faithfulness in dedicated evaluations.

1 Introduction

Multimodal Large Language Models (MLLMs) have rapidly progressed from simple perception tasks such as visual question answering and image captioning to complex multi-step reasoning tasks (Yao et al., 2024; Liu et al., 2025c; Peng et al., 2025b). Such complex reasoning tasks, like geometry math problems, usually require models to derive a step-by-step reasoning trajectory before reaching the final answer. Reinforcement learning with verifiable rewards (RLVR), which assigns training rewards only according to the correctness of the final answer, is a popular method in optimizing MLLMs on reasoning tasks due to its simplicity and efficiency (Meng et al., 2025; Liu et al., 2025b; Xu et al., 2025). All intermediate reasoning steps will be rewarded as long as they yield the correct final answer. Unfortunately, it is prevalent for the model to learn spurious reasoning under such a rewarding paradigm: models may exploit shortcuts

or generate contradictory intermediate steps that still land on the right output, effectively “hacking” the training objective. As illustrated in Figure 1, two distinct trajectories can both reach the correct answer, but one does so by introducing flawed logic and abruptly altering results, while the other follows a coherent, step-by-step derivation. Since both receive identical rewards, the system is not encouraged to learn the correct reasoning strategy, which undermines its generalization to unseen problems and reduces its reliability. Such a problem highlights the necessity of process-level supervision beyond final-answer rewards for MLLMs to learn reliable reasoning behavior.

To incorporate process-level supervision into reasoning training, a common approach is to leverage pre-trained progress reward models (PRMs), which score intermediate reasoning steps based on their correctness (Wang et al., 2025b; Luo et al., 2025). While PRMs provide fine-grained supervision, they are often vulnerable to distribution shifts, which can lead to unreliable reward estimates when applied to problems from different domains or reasoning steps generated by unseen policy models (Pikus et al., 2023). Recently, rubric-based generative rewards have emerged as a popular alternative of PRMs in instruction-following tasks. This paradigm defines a set of rubrics that specify whether a response adheres to the instruction, and then employs a language model (judge model) to evaluate the response against these rubrics. Compared to traditional reward models, rubric-based approaches offer more robust and interpretable reward signals (Viswanathan et al., 2025a; Huang et al., 2025c). However, while rubrics in instruction-following tasks can often be directly derived from the input instruction, extending this paradigm to multimodal reasoning tasks is non-trivial, as the ground-truth reasoning trajectory is usually unknown. Consequently, designing reliable rubrics and effectively integrating them into RLVR for mul-

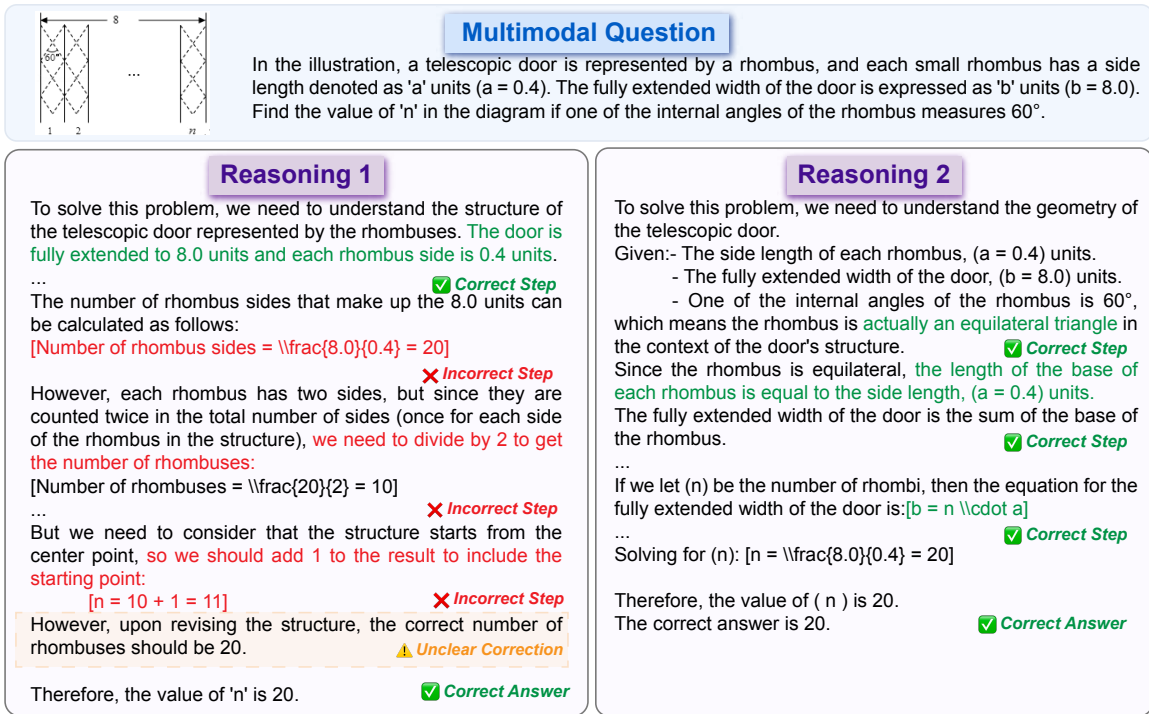


Figure 1: Illustration of a multimodal reasoning question together with two model-generated reasoning traces that both reach the correct answer. *Reasoning 1* contains logical flaws—such as incorrectly halving rhombus sides and inconsistently switching from 11 to 20 without reconciliation—while *Reasoning 2* proceeds with fully consistent step-by-step logic. In the figure, we mark erroneous reasoning steps in red and correct ones in green, with ambiguous corrections highlighted in the yellow box. Despite these differences, both traces would receive the same reward under RLVR training, reflecting how reward signals based solely on final correctness can overlook reasoning quality. Multimodal reasoning remains an open challenge.

Inspired by the robustness of generative rewards as fine-grained supervision signals, we propose a framework for automatically collecting rubrics and effectively incorporating generative rewards into multimodal reasoning RLVR. Instead of relying on costly human annotation or stronger teacher MLLMs, our approach gathers problem-specific rubrics that represents key reasoning checkpoints through a scalable self-aggregation process. Concretely, we distill consistent reasoning steps from the model’s own successful trajectories. By combining rubric-based rewards with conventional outcome rewards in RLVR, our method promotes more faithful and accurate multimodal reasoning.

With this framework, we train a model named **AutoRubric**, which demonstrates superior performance as well as faithfulness. Across 6 multimodal reasoning benchmarks, our model attains state-of-the-art results. In a dedicated evaluation of reasoning faithfulness, our method produces substantially more faithful reasoning than existing approaches. Ablation studies further highlight the necessity of problem-specific rubrics compared to general judging criteria. Moreover, detailed analysis of the training dynamics shows that our framework effectively

stabilizes training. To facilitate further research, we will release the constructed rubric dataset and code.

2 Related Work

Reinforcement Learning in MLLM Reasoning. Multimodal large language models (MLLMs) have rapidly progressed by integrating visual encoders with large language models for cross-modal understanding and reasoning. Early advances mainly relied on multimodal supervised finetuning with large-scale instruction data, such as Instruct-BLIP (Dai et al., 2023) and LLaVA (Liu et al., 2023). More recently, reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024) has emerged as a key paradigm for improving multimodal reasoning, using rule-based verification of final answers for policy optimization. Prior work largely follows two directions: (1) strengthening reasoning capability before RL by distilling multimodal chain-of-thought data from teacher models, e.g., Vision-R1 (Huang et al., 2025b) and (Wei et al., 2025); and (2) enriching supervision beyond answer correctness, such as annotated key steps in R1-VL (Zhang et al., 2025), visual perception rewards in Vision-SR1 (Xiao et al., 2025),

or reflection-based rewards in SRPO (Wan et al., 2025). However, these methods often emphasize isolated aspects of reasoning and rely heavily on costly proprietary MLLM annotations.

In contrast, AutoRubric derives problem-specific rubrics directly from multiple successful reasoning trajectories without proprietary supervision. By aggregating consistent reasoning steps and filtering spurious ones, AutoRubric provides effective process-level rewards that improve reasoning fidelity and discourage shortcut solutions.

Rubrics in RL. Since some instructions are not compatible with RLVR, and considering that traditional reward models often struggle to generalize to out-of-distribution inputs, recent literature has proposed the use of explicit rubrics to assign rewards for RL (Gunjal et al., 2025; Huang et al., 2025c). These rubrics can be either query-agnostic, focusing on general response quality, or query-specific, tailored to the nuanced requirements of a given prompt (He et al., 2025). Typically, these rubrics are integrated into a prompt for an LLM-as-a-judge to evaluate the policy model’s response. Existing methods for rubric generation include manual annotation (He et al., 2025), derivation from teacher LLM responses (Viswanathan et al., 2025b; Peng et al., 2025a), or contrastive analysis of responses with varying quality (Liu et al., 2025a). While prior work mainly applies rubrics to general instruction-following tasks where verifiable rewards are not available, our work demonstrates that rubrics can be synergized with verifiable rewards in multimodal reasoning to enhance the accuracy and faithfulness of intermediate reasoning chains.

3 Method

Our method aims to enhance the reasoning capabilities of vision-language models through a reinforcement learning framework. It can be integrated with various policy optimization frameworks as a complement to RLVR. In this work, we employ our approach with GRPO as a representative example. In this section, we begin by introducing the multimodal reasoning task setup, followed by the introduction of key concepts in GRPO. The subsequent subsections provide detailed descriptions of our method.

3.1 Preliminary

Problem Formulation In this work, we focus on using MLLMs for solving multimodal rea-

soning task. Assume we are given a collection of N multimodal reasoning instances denoted as $\mathcal{D} = \{x_i\}_{i=1}^N$. Each instance $x_i \in \mathcal{D}$ contains a visual input \mathcal{V}_i , a textual query \mathcal{Q}_i , and its labeled answer a_i . Our objective is to train a policy model that learns a function $\mathcal{F} : (\mathcal{V}_i, \mathcal{Q}_i) \mapsto a_i$.

To improve reasoning performance, the model is instructed to generate a token sequence that contains intermediate reasoning steps with the final answer:

$$s_{i,t} \sim \pi_\theta(\cdot \mid \mathcal{V}_i, \mathcal{Q}_i, s_{i,<t}), \quad t = 1, \dots, T_i,$$

where the trace $s_{i,1:T_i}$ jointly encodes the reasoning process and ends with the final answer \hat{a}_i .

Group Relative Policy Optimization. We adopt Group Relative Policy Optimization (GRPO) to optimize the policy model. GRPO is a variant of PPO that removes the need for a separately trained value function, and instead relies on relative comparisons among multiple responses sampled for the same query. This design yields a lightweight and stable training procedure.

Given a query q , we sample a group of G responses $\{o_i\}_{i=1}^G$ from the old policy $\pi_{\theta_{\text{old}}}$. Each response $o_i = (o_{i,1}, \dots, o_{i,|o_i|})$ receives a scalar reward r_i . GRPO computes a group-normalized advantage \hat{A}_i from $\{r_i\}_{i=1}^G$, which serves as the relative learning signal shared across all tokens of o_i .

The policy π_θ is then updated using a clipped importance sampling objective with KL regularization toward a fixed reference policy π_{ref} . We denote the token-level importance ratio as $\rho_{i,t}(\theta) = \pi_\theta(o_{i,t} \mid q, o_{i,<t}) / \pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})$. Full optimization details are provided in Appendix C.

3.2 Integrating LLM-As-a-Judge into RLVR

While RLVR optimizes the policy model solely based on answer correctness, this signal is often sparse and insufficient to capture the quality of intermediate reasoning. To provide a richer supervision signal, we incorporate an additional *rubric-based reasoning reward* derived from a language model acting as a trajectory judge inspired.

Rubric-guided Scoring. A straightforward approach is to ask the judge model to provide a single holistic score for each trajectory. Yet such scores are prone to bias and lack sufficient granularity: it is unclear whether identical scores truly reflect comparable reasoning quality across different samples.

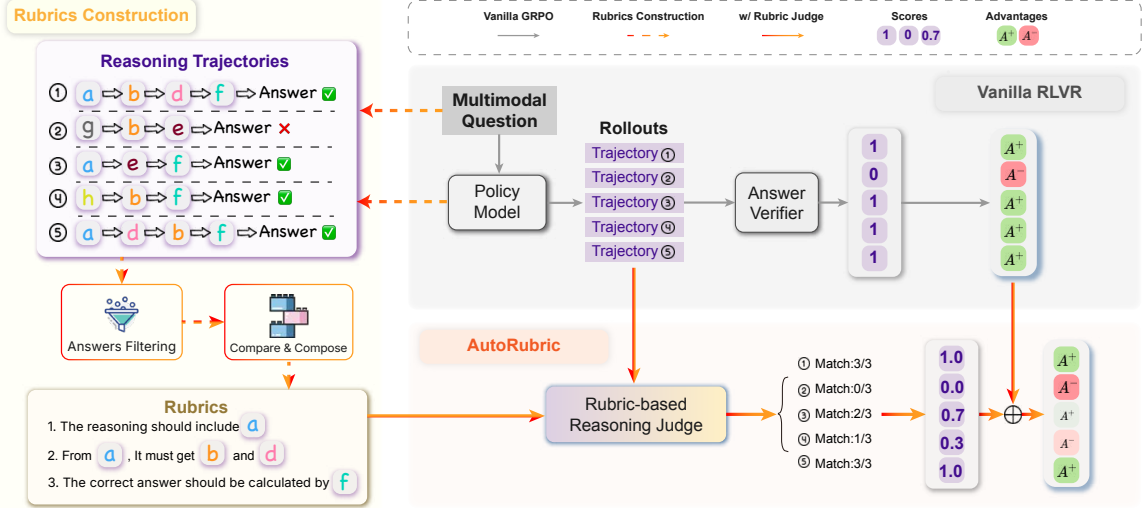


Figure 2: Our framework augments RLVR with rubric-based reasoning rewards. **Left:** Rubrics are automatically constructed by aggregating *common steps* across multiple correct reasoning trajectories, yielding problem-specific rubrics criteria without human annotation. **Right:** While vanilla RLVR cannot distinguish reasoning quality among trajectories with the same answer, rubric-based scoring enables step-level differentiation and provides richer training signals for faithful-oriented reasoning.

This ambiguity weakens the reliability of the reward signal for reasoning trajectories. To mitigate these issues, we guide the reasoning reward process with problem-specific rubrics $\mathcal{C}^x = \{c_1, \dots, c_m\}$. Each rubric item c_j specifies a key reasoning checkpoint that is expected to appear in a logically sound trajectory. Given a candidate trajectory τ , the judge model verifies whether τ satisfies each checkpoint. Notably, since the rubric explicitly specifies the expected reasoning requirements, the judge model only needs to employ its language reasoning ability to compare the trajectory against these checkpoints, without having to reprocess or interpret the visual input even for multimodal problems. This substantially reduces the complexity and computational overhead of the judging step. Let $\mathbb{1}[\tau \models c_j]$ denote an indicator function that equals 1 if c_j is satisfied, and 0 otherwise. The rubric-based reasoning reward is then computed as the fraction of satisfied checkpoints:

$$r_i^{\text{rubric}} = \frac{1}{|\mathcal{C}^x|} \sum_{j=1}^{|\mathcal{C}^x|} \mathbb{1}[\tau \models c_j]. \quad (1)$$

Combining outcome and rubric-based rewards. The rubric-based reward r_i^{rubric} is integrated with the conventional outcome reward r_i^{ans} that indicates whether the final prediction \hat{a} matches the ground truth with a weighted combination:

$$r_i = \lambda r_i^{\text{ans}} + (1 - \lambda) r_i^{\text{rubric}}, \quad (2)$$

where $\lambda \in [0, 1]$ controls the impact of the rubric-based reward. During policy optimization, the com-

bined reward r_i is assigned to each sampled trajectory, and the normalized group-relative advantages are computed following the GRPO framework. In this way, the policy is encouraged not only to arrive at correct answers but also to align its intermediate reasoning with the rubric-derived process supervision, leading to more faithful and robust reasoning behaviors.

3.3 Aggregation-based Rubric Generation

Existing approaches to acquire process supervision signals often resort to compare with manually annotated or stronger proprietary MLLMs’ reasoning trajectories. Manual annotation is prohibitively expensive. Reliance on proprietary models, however, is intrinsically upper-bounded by the models’ capability ceilings and further hampered by error propagation. Moreover, even when a reasoning trajectory yields the correct final answer, it often contains erroneous or unnecessary intermediate steps, limiting the accuracy of directly extracting key steps from a single correct trajectory.

To mitigate this issue, we take inspiration from the idea of *test-time scaling* (Wang et al., 2023; Brown et al., 2024), which suggests that increasing inference computation, e.g., sampling multiple reasoning attempts, increases the likelihood that the majority will converge to a correct solution. Analogously, we propose to *aggregate step-level consistency across the model’s own successful trajectories*. The key intuition is that if a particular step consistently appears in many correct trajec-

Table 1: Summary statistics of the rubric sets of the training samples.

Overview	
# Training Samples	38,870
# Rubric sets	26,144
Coverage	67.26%
Avg. / Total words	80.65 / 2,107,756
Rubric Criteria Statistics	
Avg. criterion	3.47
Avg. / Max words	23.25 / 198

ories, it is likely to represent a causally essential component of the reasoning process; in contrast, steps that appear only sporadically are more likely to be spurious or unnecessary. Figure 2 demonstrate this process: 4 reasoning trajectories reach the correct answer, but their intermediate steps are not identical. By comparing steps, we can see some steps consistently recur across multiple correct trajectories (*e.g.*, Reasoning from step *a* to derive *b*, *d*, and calculating final answer with step *f*). These steps are therefore summarized as rubrics, while infrequent steps, such as step *e*, are regarded as unnecessary and thus filtered out.

Given a multimodal reasoning problem x , we first sample K reasoning trajectories $\{\tau^{(k)}\}_{k=1}^K$ from the current policy. Among them, we retain the subset $\mathcal{S} \subseteq \{\tau^{(k)}\}$ whose final answers match the verifiable ground truth. We then prompt an LLM to compare trajectories in \mathcal{S} and summarize their common steps into an ordered set of key checkpoints:

$$\mathcal{C}^x = \{c_1, c_2, \dots, c_m\},$$

where each c_i denotes a reasoning checkpoint distilled from recurring steps across correct trajectories. These checkpoints are organized into \mathcal{C}^x , a structured collection of checkpoints that encodes the essential reasoning requirements for derive the correct answer, which further serve as the problem-specific rubrics for the LLM-as-a-Judge reasoning rewarding during training.

4 Experiments

4.1 Experimental Setup

Implementation Details. In our experiments, we use Qwen2.5-VL-7B-IT (Bai et al., 2025) as the base model and train it with the verl¹ framework. We adopt GRPO (Shao et al., 2024) as the policy update algorithm during training. We train the model with *ViRL-39K* dataset proposed by Wang

et al. (2025a) for 4 epochs with a constant learning rate of 1e-6. We adopt 512 as the rollout batch size and 128 as the global policy update batch size. We set the rollout number to 8 with a sampling temperature of 1.0. For rubric-based reasoning rewards, we employ an open-sourced LLM as the judge model². The KL coefficient is fixed at 0.01. All experiments are run on a single node equipped with 8 H100 GPUs. The full set of prompts used in rubric construction and rubric-based rollouts scoring in training, is provided in the Appendix.

Benchmarks. We evaluate model performance along two dimensions. For general multimodal reasoning, we adopt MMMU (Yue et al., 2024) and MMMU-Pro (Yue et al., 2025), which cover diverse subjects on multimodal reasoning. For multimodal mathematical reasoning, we include three challenging benchmarks: MathVista (Lu et al., 2024), MATH-Vision (Wang et al., 2024), and WeMATH (Qiao et al., 2025), each designed to test different aspects of multimodal mathematical problem-solving skills.

Evaluation Metrics. To comprehensively assess both answer correctness and reasoning reliability, we adopt two distinct accuracy evaluation metrics, introduced below.

Standard Accuracy evaluates a model solely based on whether its final predicted answer matches the ground-truth answer. Formally, for each prediction $i \in \mathcal{D}$, a binary indicator $c_i \in \{0, 1\}$ is assigned, where $c_i = 1$ if the final answer is correct and $c_i = 0$ otherwise. The standard accuracy is then computed as

$$\text{Acc} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} c_i. \quad (3)$$

This evaluation protocol is widely adopted in prior work due to its simplicity and ease of comparison.

Strict Accuracy evaluates a model by jointly considering final answer correctness and its consistency with the underlying reasoning process. Unlike standard answer accuracy, it captures a critical failure mode that we frequently observe in practice—particularly for models trained with intensive RLVR—where the reasoning process implies one conclusion while the final answer states a different result. Such *reasoning-answer inconsistency* has also been reported in prior work on chain-of-thought faithfulness, indicating that

¹<https://github.com/volcengine/verl>.

²<https://huggingface.co/openai/gpt-oss-20b>.

Table 2: Performance comparison of open vision-language reasoning models on multimodal reasoning benchmarks. In each cell, we report $\frac{\text{Strict Accuracy}}{[\text{Standard Accuracy} / \text{False Positive}]}$, where Strict Accuracy and False Positive terms are defined in Sec.4.1. Larger Strict Accuracy and Smaller absolute False Positive indicates more faithful reasoning. The best results are highlighted in **bold**, while the second-best are underlined.

Models	Avg.	MathVision	MathVista	MMMU	MMMU Pro	Wemath
Qwen2.5-VL-7B (Bai et al., 2025)	45.24 49.04 / -3.80	21.60 26.20 / -4.60	66.00 68.60 / -2.60	49.80 55.00 / -5.20	33.80 37.40 / -3.60	55.00 58.00 / -3.00
MM-Eureka (Meng et al., 2025)	47.83 50.57 / -2.74	24.08 27.47 / -3.39	69.80 71.80 / -2.00	51.00 52.78 / -1.78	35.66 36.47 / -0.81	58.62 64.31 / -5.69
R1-VL (Zhang et al., 2025)	38.44 40.89 / -2.45	20.43 23.39 / -2.96	53.00 54.90 / -1.90	42.00 46.56 / -4.56	26.18 27.75 / -1.57	50.57 51.84 / -1.27
NoisyRollout (Liu et al., 2025b)	<u>50.66</u> 52.39 / -1.73	<u>26.97</u> 28.29 / -1.32	<u>71.50</u> 73.00 / -1.50	<u>53.22</u> 56.11 / -2.89	37.34 38.44 / -1.10	<u>64.25</u> 66.09 / -1.84
VLAA-Thinker (Chen et al., 2025a)	45.62 49.16 / -3.54	23.55 26.88 / -3.33	67.70 70.10 / -2.40	48.00 52.33 / -4.33	33.47 36.42 / -2.95	55.40 60.06 / -4.66
Perception-R1 (Xiao et al., 2025)	50.09 51.24 / -1.15	25.89 26.84 / -0.95	71.40 72.00 / -0.60	50.89 52.89 / -2.00	<u>38.38</u> 39.13 / -0.75	63.91 65.34 / -1.43
ThinkLite-VL (Wang et al., 2025c)	47.84 51.19 / -3.35	22.53 24.54 / -2.01	68.70 73.30 / -4.60	50.56 53.67 / -3.11	36.82 39.42 / -2.60	60.57 65.00 / -4.43
Vision-G1 (Zha et al., 2025)	48.37 53.92 / -5.55	25.82 28.75 / -2.93	70.00 76.40 / -6.40	47.67 53.78 / -6.11	34.45 39.48 / -5.03	63.91 71.21 / -7.30
VL-Rethinker (Wang et al., 2025a)	49.22 54.15 / -4.93	25.69 31.12 / -5.43	70.80 73.90 / -3.10	52.22 57.11 / -4.89	35.95 39.42 / -3.47	61.44 69.20 / -7.76
VL-Reasoner (Wang et al., 2025a)	48.56 53.38 / -4.82	25.23 29.87 / -4.64	70.60 74.80 / -4.20	52.00 56.22 / -4.22	35.66 38.96 / -3.30	59.31 67.07 / -7.76
AutoRubric	53.24 55.26 / -2.02	29.14 30.49 / -1.35	73.60 75.80 / -2.20	54.67 56.56 / -1.89	39.60 40.98 / -1.38	69.20 72.47 / -3.27

model-generated rationales may be misaligned with the actual decision process or final outputs (Huang et al., 2025a; Jiang et al., 2025; Chen et al., 2025b).

To address this issue, we introduce Strict Accuracy, a stricter metric that penalizes predictions whose final answers are not supported by their own reasoning. Specifically, an external judge is used to verify whether the conclusion implied by the reasoning matches the final boxed answer, without access to ground-truth labels. Predictions that are correct under standard accuracy but fail this consistency check are treated as false positives and excluded. Concrete qualitative examples and judge reliability evaluation are provided in Appendix E.1.

Formally, let \mathcal{D} denote the evaluation set. For each prediction $i \in \mathcal{D}$, let c_i denote the correctness indicator under standard accuracy, and let $s_i \in \{0, 1\}$ indicate whether the reasoning and final answer are consistent according to the judge. We define a reasoning-answer inconsistency indicator as $\text{Inc}_i = \mathbb{I}(s_i = 0)$, and the corresponding *Inconsistency Rate* as $\text{IncR} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \text{Inc}_i$. A *false positive* is a prediction that is correct under standard accuracy but inconsistent in reasoning, defined as $\text{FP}_i = \mathbb{I}(c_i = 1 \wedge s_i = 0)$. The *False Positive Rate* is $\text{FPR} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} \text{FP}_i$. Strict Ac-

curacy is then defined as

$$\text{StrictAcc} = \frac{1}{|\mathcal{D}|} \sum_{i \in \mathcal{D}} c_i \cdot s_i. \quad (4)$$

Baseline Methods. We compare our model with 10 MLLMs, including: Qwen2.5-VL-7B-IT (Bai et al., 2025), MM-Eureka-7B (Meng et al., 2025), R1-VL-7B (Zhang et al., 2025), NoisyRollout-7B (Liu et al., 2025b), VLAA-Thinker (Chen et al., 2025a), Perception-R1-7B (Xiao et al., 2025), ThinkLite-VL-7B (Wang et al., 2025c), Vision-G1 (Zha et al., 2025), Vision-G1 (Zha et al., 2025), VL-Rethinker-7B (Wang et al., 2025a), and VL-Reasoner-7B (Wang et al., 2025a).

Rubric Construction and Statistics. AutoRubric constructs problem-specific rubrics automatically from model-generated reasoning trajectories, enabling process-level supervision without manual annotation. Specifically, for each training sample, we generate 8 reasoning trajectories using a lightly warmed-up model and retain only those with correct final answers to derive rubrics. For problems with more than 3 correct trajectories, we feed the corresponding ones into a text-only LLM³, which extracts shared

³<https://huggingface.co/openai/gpt-oss-120b>.

steps across successful solutions and composes a structured set of rubric criteria.

Applying this procedure to the training data yields 26,144 rubric sets, corresponding to a coverage rate of 67.3%. More fine-grained statistics of rubric composition are summarized in Table 1. We further conduct a small-scale human evaluation on a random subset of 100 rubric sets, achieving an average score of 4.18 out of 5. Detailed rubrics construction process, human evaluation protocols, and additional analyses are provided in the Appendix B.1.

4.2 Experimental Results

We present the performance comparison between AutoRubric and existing state-of-the-art MLLMs across multiple benchmarks in Table 2. We observe the following results: **AutoRubric achieves the strongest performance under both Strict and Standard Accuracy.** As shown in Table 2, AutoRubric attains the highest Strict Accuracy across all benchmarks, with an average score of 53.24, outperforming the base model Qwen2.5-VL-7B by +8.00 points (45.24 \rightarrow 53.24). At the same time, it also achieves the best standard accuracy (55.26 on average), surpassing strong baselines. These results indicate that AutoRubric improves final answer correctness while simultaneously enhancing reasoning faithfulness. **Different models exhibit markedly different levels of reasoning-answer inconsistency.** Despite their relatively high standard accuracy, models such as Vision-G1 and VL-Rethinker suffer from severe inconsistency, with large average drops of -5.55 and -4.93 under Strict Accuracy, respectively. In contrast, models like Perception-R1 and NoisyRollout exhibit much smaller penalties (-1.15 and -1.73 on average), but their standard accuracy remains notably lower than AutoRubric. By comparison, AutoRubric maintains a low inconsistency (-2.02) while achieving the highest Strict Accuracy overall, demonstrating a more favorable trade-off between accuracy and faithfulness. **The prevalence of false positives varies substantially across benchmarks.** On WeMath, most models exhibit relatively large false positive rates, with drops of -7.30 for Vision-G1 and -7.76 for VL-Rethinker, likely due to the dominance of multiple-choice questions and the generally higher answer accuracy. In contrast, MathVision shows much smaller penalties for several models, such as -0.95 for Perception-R1 and -1.32 for NoisyRollout, indicating more stable alignment between reasoning

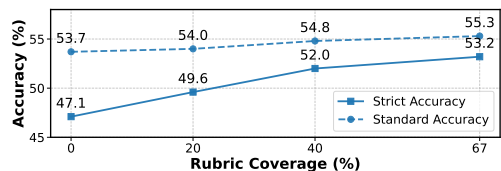


Figure 3: Effect of rubric coverage on performance. Higher rubric coverage yields consistent gains, particularly in Strict Accuracy.

Table 3: Ablation study on the design of judge rewards. **Std. Acc.** denotes the Standard Accuracy and **Strict Acc.** denotes the Strict Accuracy. The results are averaged across 5 benchmarks. AutoRubric achieves the highest Strict Acc by incorporating rubric-based judge rewards, while removing rubrics or judge rewards leads to a substantial drop in faithfulness-aware performance.

Methods	Judge	Rubrics	Std. Acc.	Strict Acc.
AutoRubric	✓	✓	55.26	53.24
w/o Rubrics	✓	×	53.11	49.43
w/o Judge Rewards	×	×	53.75	47.06

processes and final answers. These observations suggest that benchmark design should carefully account for whether models genuinely solve the underlying problem or merely arrive at the correct answer through lucky guessing, which manifests as reasoning-answer inconsistency.

4.3 Ablation Study

Effect of Reward Source In this section, we conduct ablation studies to analyze the contribution of different components in our reward design. Specifically, we compare AutoRubric with two variants: (1) *w/o Rubrics*, which employs a judge model to score reasoning trajectories but removes problem-specific rubrics, and (2) *w/o Judge Rewards*, which corresponds to Vanilla RLVR and relies solely on rule-based answer rewards without any judge-based supervision. This design allows us to separately examine the roles of judge rewards and rubric-based guidance.

As shown in Table 3, AutoRubric achieves the best performance in both Standard Accuracy and Strict Accuracy (averaged across five benchmarks), indicating improvements not only in final answer correctness but also in reasoning faithfulness. Removing rubric-based guidance (*w/o Rubrics*) or judge rewards (*w/o Judge Rewards*) leads to noticeable degradation in performance, with the gap becoming substantially more pronounced under Strict Accuracy. Notably, although *w/o Rubrics* and *w/o Judge Rewards* exhibit comparable Standard Accuracy, incorporating judge rewards—even without problem-specific rubrics—already yields a

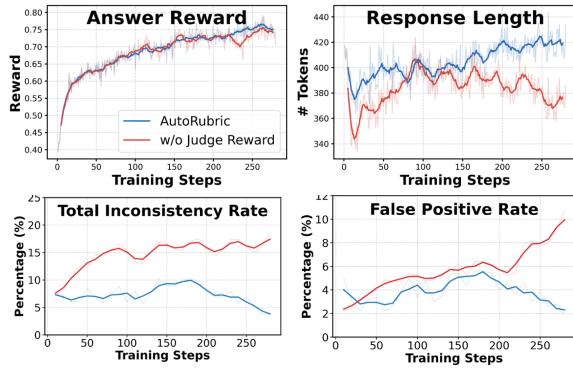


Figure 4: Top row: Comparison between AutoRubric and vanilla RLVR (*w/o* judge rewards) in training dynamics, including the evolution of answer-based training rewards and the response length of rollouts. Bottom row: Reasoning–answer inconsistency rate and false positive rate of model rollouts evaluated at different training steps.

clear improvement in Strict Accuracy, suggesting enhanced reasoning faithfulness. In comparison, AutoRubric further amplifies this gain by introducing rubric-based judge supervision, resulting in the most faithful and consistent reasoning behavior. This observation highlights that Strict Accuracy serves as a more sensitive metric for evaluating reasoning faithfulness and demonstrates that our method promotes consistent and faithful reasoning beyond merely optimizing final answers.

Effect of Rubric Coverage. We vary the proportion of training instances equipped with rubrics while keeping all other settings fixed. The full AutoRubric setup uses 67.26% rubric coverage; for 20% and 40%, we randomly subsample from this set and apply rubric-based judge rewards only to the selected instances, while the 0% setting uses answer-only supervision.

As shown in Figure 3, increasing rubric coverage consistently improves both Standard Accuracy and Strict Accuracy, with a substantially stronger effect on Strict Accuracy. Even 20% coverage yields clear gains over the 0% baseline, while higher coverage further enhances reasoning faithfulness, indicating that rubric-based supervision provides effective and sample-efficient process-level guidance.

4.4 Training Dynamics

To analyze the effect of rubric-based reasoning rewards, we visualize training dynamics and rollout-level evaluation statistics in Figure 4. The top row shows the evolution of answer-based training rewards and response length. In the early stage, AutoRubric and vanilla RLVR exhibit sim-

ilar reward trends. As training proceeds, vanilla RLVR develops pronounced oscillations, whereas AutoRubric improves in a smooth and stable manner. This divergence indicates that answer-only rewards become unreliable supervision signals at later stages, leading to unstable optimization. By contrast, rubric-based reasoning rewards provide richer and more structured feedback, preventing such collapse. The top-right panel shows that AutoRubric consistently generates longer responses than vanilla RLVR. This suggests that short rollouts that only adjust the final answer are insufficient to obtain high rewards under rubric-based supervision, thereby encouraging more extended and coherent reasoning trajectories.

The bottom row reports rollout-level reasoning faithfulness metrics. Vanilla RLVR shows a markedly higher reasoning–answer inconsistency rate that further increases during training, accompanied by a steadily rising false positive rate. This indicates a fundamental limitation of answer-only supervision: once inconsistency emerges, rewards can no longer reliably distinguish faithful reasoning from answer-correct but inconsistent trajectories. In contrast, AutoRubric consistently maintains lower inconsistency and false positive rates, demonstrating that rubric-based trajectory supervision effectively mitigates reward hacking and promotes faithful reasoning.

5 Conclusion

In this work, we identify reasoning–answer inconsistency as a critical failure mode of answer-only RLVR for multimodal reasoning, which leads to unstable training dynamics and unfaithful reasoning behaviors. To address this issue, we propose AutoRubric, a reinforcement learning framework that introduces rubric-based process supervision via an LLM-as-a-judge. By automatically inducing problem-specific rubrics from consistent reasoning trajectories, AutoRubric provides structured, trajectory-level feedback without requiring human annotation or stronger teacher models. Experiments on six multimodal reasoning benchmarks show that AutoRubric not only improves standard accuracy, but also substantially enhances reasoning faithfulness under stricter evaluation metrics. Overall, our results demonstrate that rubric-based process supervision is an effective and scalable approach for training more reliable multimodal reasoning models.

Limitations

AutoRubric relies on an LLM-as-a-judge to compute rubric-based rewards, which introduces additional computational overhead during training. In our experiments, incorporating the judge increases the training time per step by approximately 40% compared to vanilla RLVR.

Although training efficiency is an important consideration, faithful reasoning is critical in many real-world applications, such as decision support and safety-sensitive systems, where correct final answers alone are insufficient. In these settings, unfaithful or inconsistent reasoning trajectories can undermine reliability and interpretability. As shown in Table 2, AutoRubric substantially improves the consistency of reasoning trajectories in addition to accuracy, whereas Table 3 shows that RLVR without judge supervision may even exacerbate reasoning inconsistency. These results suggest that auxiliary LLM-as-a-judge signals are necessary for encouraging faithful reasoning, making the additional computation a justified trade-off.

Nevertheless, reducing this overhead remains an important direction. To balance faithfulness and efficiency, AutoRubric aggregates all rubric evaluations into a single LLM call, which is more efficient than evaluating each rubric independently. Moreover, the overhead can be further reduced through parallelization by deploying multiple judge replicas when additional GPU resources are available, enabling higher reward-computation throughput.

References

- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. [Qwen2.5-vl technical report](#). *CoRR*, abs/2502.13923.
- Bradley C. A. Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *CoRR*, abs/2407.21787.
- Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025a. [SFT or rl? an early investigation into training rl-like reasoning large vision-language models](#). *Trans. Mach. Learn. Res.*, 2025.
- Yi Chen, Yuying Ge, Rui Wang, Yixiao Ge, Junhao Cheng, Ying Shan, and Xihui Liu. 2025b. [GRPO: consistency-aware reinforcement learning for multimodal reasoning](#). *CoRR*, abs/2506.16141.
- Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. 2023. [Instructblip: Towards general-purpose vision-language models with instruction tuning](#). In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, NeurIPS 2023*.
- Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. 2025. [Rubrics as rewards: Reinforcement learning beyond verifiable domains](#). *CoRR*, abs/2507.17746.
- Yun He, Wenzhe Li, Hejia Zhang, Songlin Li, Karishma Mandyam, Sopan Khosla, Yuanhao Xiong, Nanshu Wang, Xiaoliang Peng, Beibin Li, and 1 others. 2025. [Advancedif: Rubric-based benchmarking and reinforcement learning for advancing llm instruction following](#). *arXiv preprint arXiv:2511.10507*.
- Minbin Huang, Runhui Huang, Chuanyang Zheng, Jingyao Li, Guoxuan Chen, Han Shi, and Hong Cheng. 2025a. [Answer-consistent chain-of-thought reinforcement learning for multi-modal large language models](#). *CoRR*, abs/2510.10104.
- Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025b. [Vision-r1: Incentivizing reasoning capability in multimodal large language models](#). *CoRR*, abs/2503.06749.
- Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, and 1 others. 2025c. [Reinforcement learning with rubric anchors](#). *arXiv preprint arXiv:2508.12790*.
- Enyi Jiang, Changming Xu, Nischay Singh, and Gagandeep Singh. 2025. [Misaligning reasoning with answers - A framework for assessing LLM cot robustness](#). *CoRR*, abs/2505.17406.
- Rensis Likert. 1932. A technique for the measurement of attitudes. *Archives of psychology*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. [Visual instruction tuning](#). In *Advances in Neural Information Processing Systems: Annual Conference on Neural Information Processing Systems, NeurIPS 2023*.
- Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. 2025a. [Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment](#). *arXiv preprint arXiv:2510.07743*.
- Xiangyan Liu, Jinjie Ni, Zijian Wu, Chao Du, Longxu Dou, Haonan Wang, Tianyu Pang, and Michael Qizhe Shieh. 2025b. [Noisyrollout: Reinforcing visual reasoning with data augmentation](#). *CoRR*, abs/2504.13055.

- Zhiyuan Liu, Yuting Zhang, Feng Liu, Changwang Zhang, Ying Sun, and Jun Wang. 2025c. [Othink-mr1: Stimulating multimodal generalized reasoning capabilities via dynamic reinforcement learning](#). *CoRR*, abs/2503.16081.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. [Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts](#). In *The International Conference on Learning Representations, 2024*. OpenReview.net.
- Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. [Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning](#). In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP*, pages 6774–6786. Association for Computational Linguistics.
- Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyao Yu, Xinzhe Ni, Zicheng Lin, Jin Zeng, and Yujiu Yang. 2025. [URSA: understanding and verifying chain-of-thought reasoning in multimodal mathematics](#). *CoRR*, abs/2501.04686.
- Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, and 1 others. 2025. [Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning](#). *arXiv preprint arXiv:2503.07365*.
- Hao Peng, Yunjia Qi, Xiaozhi Wang, Bin Xu, Lei Hou, and Juanzi Li. 2025a. [Verif: Verification engineering for reinforcement learning in instruction following](#). *arXiv preprint arXiv:2506.09942*.
- Yi Peng, Chris, Xiaokun Wang, Yichen Wei, Jiangbo Pei, Weijie Qiu, Ai Jian, Yunzhuo Hao, Jiachun Pan, Tianyidan Xie, Li Ge, Rongxian Zhuang, Xuchen Song, Yang Liu, and Yahui Zhou. 2025b. [Skywork R1V: pioneering multimodal reasoning with chain-of-thought](#). *CoRR*, abs/2504.05599.
- Benjamin Pikus, Will LeVine, Tony Chen, and Sean Hendryx. 2023. [A baseline analysis of reward models' ability to accurately analyze foundation models under distribution shift](#). *CoRR*, abs/2311.14743.
- Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma Gongque, Shanglin Lei, Yifan Zhang, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Xiao Zong, Yida Xu, Peiqing Yang, Zhimin Bao, Muxi Diao, Chen Li, and Honggang Zhang. 2025. [We-math: Does your large multimodal model achieve human-like mathematical reasoning?](#) In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025*, pages 20023–20070. Association for Computational Linguistics.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.
- Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. 2025a. [Checklists are better than reward models for aligning language models](#). *CoRR*, abs/2507.18624.
- Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. 2025b. [Checklists are better than reward models for aligning language models](#). *arXiv preprint arXiv:2507.18624*.
- Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, Chaofan Tao, Yangfan He, Mi Zhang, and Shen Yan. 2025. [SRPO: enhancing multimodal LLM reasoning via reflection-aware reinforcement learning](#). *CoRR*, abs/2506.01713.
- Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhui Chen. 2025a. [VI-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning](#). *CoRR*, abs/2504.08837.
- Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. [Measuring multimodal mathematical reasoning with math-vision dataset](#). In *Advances in Neural Information Processing Systems, 2024*.
- Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, Lewei Lu, Haodong Duan, Yu Qiao, Jifeng Dai, and Wenhai Wang. 2025b. [Visualprm: An effective process reward model for multimodal reasoning](#). *CoRR*, abs/2503.10291.
- Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. 2025c. [Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement](#). *CoRR*, abs/2504.07934.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Lai Wei, Yuting Li, Kaipeng Zheng, Chen Wang, Yue Wang, Linghe Kong, Lichao Sun, and Weiran Huang. 2025. [Advancing multimodal reasoning via reinforcement learning with cold start](#). *CoRR*, abs/2505.22334.

Tong Xiao, Xin Xu, Zhenya Huang, Hongyu Gao, Quan Liu, Qi Liu, and Enhong Chen. 2025. [Advancing multimodal reasoning capabilities of multimodal large language models via visual perception reward](#). *CoRR*, abs/2506.07218.

Shilin Xu, Yanwei Li, Rui Yang, Tao Zhang, Yueyi Sun, Wei Chow, Linfeng Li, Hang Song, Qi Xu, Yunhai Tong, Xiangtai Li, and Hao Fei. 2025. [Mixed-r1: Unified reward perspective for reasoning capability in multimodal large language models](#). *CoRR*, abs/2505.24164.

Huanjin Yao, Jiaying Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and Dacheng Tao. 2024. [Mulberry: Empowering MLLM with o1-like reasoning and reflection via collective monte carlo tree search](#). *CoRR*, abs/2412.18319.

Xiang Yue, Yuansheng Ni, Tianyu Zheng, Kai Zhang, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. [MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI](#). In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024*, pages 9556–9567. IEEE.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhao Chen, and Graham Neubig. 2025. [Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 15134–15186. Association for Computational Linguistics.

Yuheng Zha, Kun Zhou, Yujia Wu, Yushu Wang, Jie Feng, Zhi Xu, Shibo Hao, Zhengzhong Liu, Eric P. Xing, and Zhiting Hu. 2025. [Vision-g1: Towards general vision language reasoning with multi-domain data curation](#). *CoRR*, abs/2508.12680.

Jingyi Zhang, Jiaying Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. 2025. [R1-VL: learning to reason with multimodal large language models via step-wise group relative policy optimization](#). *CoRR*, abs/2503.12937.

A Evaluation Protocol

The benchmarks used in our evaluation consists of two types of questions: multiple-choice questions and open-ended questions. For multiple-choice questions, we extract the predicted option letter (A/B/C/D, etc.) using regular expressions. The extracted option is then directly compared against

the ground-truth label. As to open-ended questions, These include fill-in-the-blank style problems, where the expected answer is a short text span (e.g., a number, a word, or a short phrase). Since exact string matching may fail to capture semantically correct but differently phrased answers, we use Qwen3-30B-A3B-Instruct-2507⁴ as a proxy judge for evaluation. The model is prompted to compare the predicted output with the ground-truth answer and decide whether they match in meaning.

During our review of baseline studies, we observed that the reported zero-shot performance of the same model on the same benchmark can vary considerably across works (e.g., the Qwen2.5-VL-7B-IT model on MathVerse is reported as 47.9 in MM-EUREKA (Meng et al., 2025), but 46.2 in NoisyRollout (Liu et al., 2025b)). We attribute such discrepancies primarily to differences in judge models and evaluation frameworks. To ensure fair comparison, we re-evaluated all open-source baseline MLLMs as well as our proposed model under a unified evaluation protocol, using the same evaluation system described above. Notably, we strictly follow the system and instructional prompts (e.g., response format requirements) provided in the original studies in reproduction, thereby ensuring that the performance comparison tables reflect results obtained under a controlled and standardized setting.

B Supplementary Analysis of Rubrics

This section provides a detailed supplementary analysis of the rubric-based formulation adopted in AutoRubric. We first describe how rubrics are constructed and summarize their overall statistics and quality. We then present a qualitative comparison between rubric-based supervision and the key-step formulation proposed in R1-VL (Zhang et al., 2025), highlighting their differences in expressiveness and supervisory effectiveness.

B.1 Rubric Construction and Statistics

Construction. To construct problem-specific rubrics, we begin by collecting multiple reasoning trajectories for each training sample and retain only those that yield correct final answers. If no correct trajectory is obtained for a sample, no rubric is generated. To improve rubric coverage, we first train the base Qwen-2.5-VL-7B-IT model for one epoch

⁴<https://huggingface.co/Qwen/Qwen3-30B-A3B-Instruct-2507>

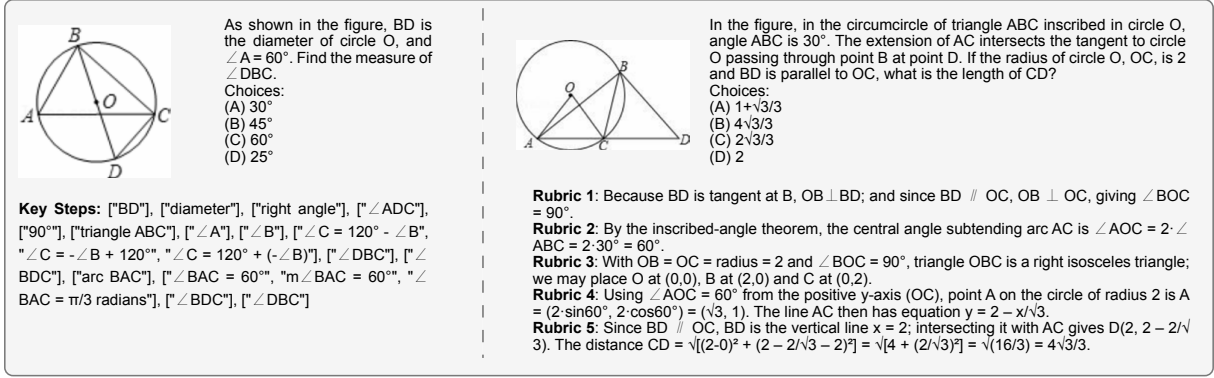


Figure 5: A comparison between (left) key steps proposed in R1-VL; and (right) rubrics constructed with AutoRubric under two similar geometry problems.

Table 4: Performance comparison with *Geometry3K* training data. AutoRubric yields consistent gains beyond the primary *ViRL-39K* training setting.

Models	Avg.	MathVision	MathVista	WeMath	MMMU	MMMU Pro
Vanilla RLVR	49.67	27.01	71.30	62.07	51.89	36.07
AutoRubric	50.99	26.84	71.60	63.68	54.67	38.15

using standard RLVR, and then use this intermediate model to generate eight reasoning trajectories per problem. For problems with more than three correct trajectories, we feed the corresponding trajectories into an open-source text-only LLM⁵. The LLM is prompted to compare these trajectories, identify their shared reasoning steps, and synthesize them into a structured set of rubric criteria. Each criterion is expressed as a complete semantic statement describing an essential aspect of correct reasoning. The full prompt used for rubric generation is provided in the Appendix.

Statistics. Beyond overall coverage, Table 1 reveals several structural properties of the generated rubrics. On average, each rubric set contains 3.47 criteria, indicating that the extracted supervision typically decomposes a solution into multiple evaluative aspects rather than a single coarse requirement. Each criterion has an average length of 23.25 words, with some criteria reaching up to 198 words, suggesting that the rubrics are semantically rich and capable of encoding detailed reasoning constraints. Across the full training set, the total rubric text amounts to over 2.1 million words, reflecting the substantial volume of process-level supervision signals introduced by our rubric construction procedure.

⁵<https://huggingface.co/openai/gpt-oss-120b>.

Human Evaluation of Rubrics Quality. To assess the quality of the automatically generated rubrics, we further conduct a small-scale human evaluation. Two graduate students jointly evaluate a randomly sampled subset of 100 rubric sets using a 5-point Likert scale (Likert, 1932) (1 = poor, 5 = excellent), based on their relevance to the problem and correctness for evaluating reasoning trajectories. The final score for each rubric set is computed as the average of the two ratings. Overall, the rubrics achieve an average score of 4.18, indicating that they are generally relevant and informative. The two annotators also exhibit strong agreement, with an average absolute rating difference of 0.82.

B.2 Comparison with Key Steps in R1-VL

To further contextualize the design of rubric-based supervision, Figure 5 provides a qualitative comparison between the key-step representation used in R1-VL (Zhang et al., 2025) and the rubric formulation adopted by AutoRubric. The figure presents two representative geometry problems. For each problem, the left column shows the concise key steps extracted following R1-VL, while the right column displays the corresponding rubric set generated by AutoRubric.

As illustrated, the key steps in R1-VL are extremely concise, often consisting of short phrases or isolated keywords. Although such representations are compact, their limited expressiveness

tends to reduce evaluation to superficial keyword matching. This makes it difficult to assess higher-level reasoning properties such as logical coherence, completeness, or whether intermediate conclusions are properly justified. Consequently, key-step supervision provides only weak and coarse-grained guidance for training reasoning-intensive models.

In contrast, AutoRubric constructs structured, criterion-based rubrics in which each criterion is formulated as a complete and semantically rich statement. These rubrics explicitly encode what constitutes correct reasoning at different stages of problem solving, offering clearer and more interpretable evaluation standards. Rather than checking for the presence of specific keywords, rubric-based evaluation focuses on whether the reasoning process satisfies meaningful semantic conditions. This richer formulation enables more informative and reliable reward signals, which are better suited for supervising complex multimodal reasoning trajectories.

C Details of Group Relative Policy Optimization

This section provides the full formulation of Group Relative Policy Optimization (GRPO) used in our experiments.

Given a query q and a group of G responses $\{o_i\}_{i=1}^G$ sampled from the old policy $\pi_{\theta_{\text{old}}}$, the GRPO objective is defined as

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\} \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left(\min \left(\rho_{i,t}(\theta) \hat{A}_i, \tilde{\rho}_{i,t}(\theta) \hat{A}_i \right) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right) \right] \quad (5)$$

where the token-level importance ratio is

$$\rho_{i,t}(\theta) = \frac{\pi_{\theta}(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad (6)$$

and the clipped ratio is

$$\tilde{\rho}_{i,t}(\theta) = \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon). \quad (7)$$

Each response o_i is assigned a scalar reward r_i . The advantage \hat{A}_i is computed via group-wise normalization:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}. \quad (8)$$

Here, ϵ is the PPO clipping parameter, β controls the strength of KL regularization, and π_{ref} denotes a fixed reference policy.

D Additional Training Results

D.1 Training on Other Datasets

To further assess the robustness of our training framework, we additionally train the model on a dataset distinct from the main training corpus. This experiment aims to evaluate whether the proposed method remains effective when applied to tasks with different data distributions and reasoning characteristics. Specifically, we adopt the Geometry3K dataset (Lu et al., 2021), a multimodal reasoning benchmark that requires models to solve geometry-related problems. The dataset comprises approximately 2.1K training samples, and we trained the model for 20 epochs, resulting in a total of 90 optimization steps.

Table 4 presents the comparison between AutoRubric and the vanilla RLVR baseline on several mathematical and general reasoning benchmarks. As shown in the table, AutoRubric consistently outperforms, achieving a notably higher average performance (+1.36 points). Notably, AutoRubric yields clear gains on MMMU and MMMU-Pro, which are designed for general reasoning beyond pure geometry. This demonstrates that the method enhances reasoning ability in broader contexts.

D.2 Data Efficiency

We compare AutoRubric and vanilla RLVR trained with 10% and 100% of the data (Table 5). Both models are trained for 15 epochs, and the subsets are randomly sampled from the full training set. Vanilla RLVR shows little improvement when scaling data from 10% to 100% (52.42 \rightarrow 52.96). One possible explanation is that the training data share highly similar distributions, causing the vanilla model to overfit and show limited generalization with more samples. In contrast, AutoRubric achieves more stable gains, improving from 53.38 to 55.28. Moreover, with only 10% of data, it already matches or surpasses the full-data RLVR on several benchmarks (e.g., WeMath, MMMU), demonstrating strong data efficiency and better utilization of limited supervision.

Table 5: Performance comparison of models trained with 10% and 100% of the training data. AutoRubric maintains consistent improvements under limited training data.

Models	Avg.	MathVision	MathVista	WeMath	MMMU	MMMU Pro
Vanilla RLVR (10%)	52.75	27.96	74.0	69.37	54.22	38.21
AutoRubric (10%)	53.38	28.06	74.5	70.75	54.67	38.90
w/o Judge Rewards (100%)	53.33	28.78	74.3	70.46	54.11	39.02
AutoRubric (100%)	55.28	31.35	75.9	71.09	57.56	40.52

E Unfaithfulness Phenomena in Reasoning

E.1 Reasoning Inconsistency Observation

Figure 6 presents representative qualitative examples from the MathVista benchmark that illustrate the reasoning–answer inconsistency phenomenon discussed in Sec. 4.1. The figure includes two problems and corresponding model outputs from VL-Rethinker and AutoRubric, respectively. In both cases, the models produce intermediate reasoning that leads to a different conclusion from the final stated answer.

For each example, the proposed strict accuracy judge first generates a detailed reasoning trace to analyze the logical steps taken by the model, and then outputs a structured judgment result that explicitly checks the consistency between the reasoning–derived conclusion and the final answer. As shown in the figure, the judge correctly identifies cases where the reasoning supports one numerical or categorical outcome, while the final answer reports a conflicting result.

These examples demonstrate that such inconsistencies occur in real model outputs on standard multimodal reasoning benchmarks, and that the proposed judge-based evaluation is effective at detecting this failure mode in a systematic and automated manner.

Judge Reliability. To assess the reliability of the strict accuracy judge, we randomly sample 100 judged predictions across five benchmarks (20 per benchmark) and manually verify the detected reasoning–answer inconsistencies. The judge achieves over 85% recall and over 90% precision, confirming that it provides a reliable signal for measuring reasoning faithfulness.

E.2 Additional Analysis of Faithfulness

Besides reasoning inconsistency check described in Sec. 4.1, we further introduce a **Reasoning Quality Check**, which serves as a stricter and more

comprehensive evaluation for reasoning trajectories. While inconsistency check focuses on identifying contradictions or unjustified answer shifts within a reasoning process, the quality assessment additionally examines three complementary dimensions: *unfounded transitions*, *calculation errors*, and *logical connectivity*. This broader evaluation captures not only internal contradictions but also the overall soundness and coherence of the reasoning trajectories, reflecting how logically valid and trustworthy a model’s reasoning process is. Specifically, at each training steps from 160 to 280 training steps, we sample 100 samples from MathVision benchmark and evaluate the reasoning trajectories generated by the two models with a strong judge model (*i.e.*, GPT-4o) under instruction shown in Fig. 12. The judge model outputs a holistic quality score between 0 and 1

The results of quality scores and inconsistency rate are demonstrated in Figure 7. Below are our key observations from this evaluation.


Vanilla RLVR shows persistently high and rising inconsistency. The inconsistency rate of Vanilla RLVR remains above 20% and increases with training, suggesting growing instability in its reasoning patterns. In contrast, AutoRubric keeps inconsistency around 10% throughout, indicating more stable and faithful reasoning.

AutoRubric maintains higher reasoning quality. AutoRubric consistently outperforms Vanilla RLVR, with quality scores that gradually improve over time. Vanilla RLVR, by comparison, exhibits flat or slightly declining quality, implying that training methods focusing solely on outcome correctness may potentially degrade the model’s reasoning quality as the number of training steps increases.

F Case Study

To clearly demonstrate the effectiveness of our method in rewarding the reasoning trajectories during training, we present a concrete case study. As illustrated in Figure 8, we illustrate a problem, and the constructed set of rubrics for it by AutoRubric.

MathVista Problem #350



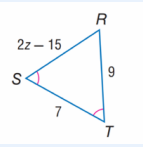
Hint: Please answer the question requiring an integer answer and provide the final value, e.g., 1, 2, 3, at the end.
Question: Move the ruler to measure the length of the line to the nearest centimeter. The line is about () centimeters long.

VL-Rethinker: To measure the length of the line using the ruler, follow these steps:
1. Identify the starting point of the line on the ruler. The line begins at the 0 cm mark.
2. Identify the ending point of the line on the ruler. The line ends between the 5 cm and 6 cm marks, closer to the 5 cm mark.
3. Determine the length by subtracting the starting point from the ending point. Since the line ends closer to the 5 cm mark, we can say it is approximately 5 cm long.
Given these observations, the line is about **4** centimeters long when rounded to the nearest centimeter.

Judge Analysis: We need to check if the reasoning conclusion matches the final answer in the box. The reasoning says "approximately 5 cm long" but the final boxed answer is 4. So mismatch.

Judge Structural Output: {"Reasoning results": "5", "boxed answer": "4", "consistent": false}

MathVista Problem #236



Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, at the end.
Question: Find z .
Choices:
(A) 7
(B) 9
(C) 12
(D) 15

AutoRubric: $\langle \text{think} \rangle$ To find the value of z , we need to use the information given in the triangle. The triangle has two angles marked with a pink arc, which indicates that the sides opposite these angles are equal. Therefore, the sides SR and ST are equal.
Given: $SR = 2z - 15$, $ST = 7$
Since $SR = ST$, we can set up the equation:
 $2z - 15 = 7$
Now, solve for z :
 $2z - 15 = 7$
 $2z = 7 + 15$
 $2z = 22$
 $z = \frac{22}{2}$
 $z = 11$
However, this value does not match any of the given choices. Let's recheck the problem statement and the choices provided. ...
The correct value of z that satisfies the equation is $z = 12$.
Therefore, the correct answer is: **C**

Judge Analysis: We need to check consistency: The reasoning states they solved $2z - 15 = 7 \Rightarrow z = 11$. Then they say "This value does not match any of the choices. Let's recheck..." The correct value of z that satisfies the equation is $z = 12$. That is inconsistent. The reasoning conclusion is $z = 11$, but final answer says $z = 12$ (option C). So inconsistent. Also the final boxed answer is "C". So we need to output JSON with 'Reasoning results' numeric conclusion from reasoning and the corresponding option? The reasoning conclusion numeric is 11. ...
Judge Structural Output: {"Reasoning results": 11, "boxed answer": "C", "consistent": false}

Figure 6: Two representative MathVista problems are shown with model outputs from VL-Rethinker and AutoRubric, where the intermediate reasoning contradicts the final answer. Our Strict Accuracy judge generates an explicit reasoning analysis followed by a structured consistency verdict, effectively detecting this failure mode.

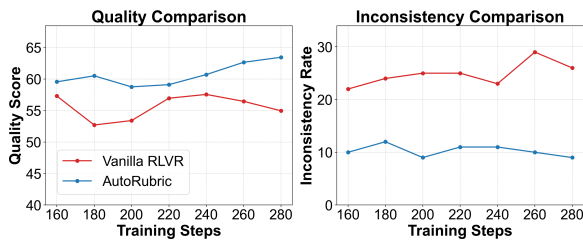


Figure 7: Comparison of Vanilla RLVR and AutoRubric on reasoning accuracy, quality, and inconsistency over training steps.

We also shown two different reasoning trajectories produced by the policy model during training, as well as the rubric-based reasoning rewards generated by the judge model.

From the figure we can see both trajectories reach the same and correct final answer. However, the rubric-based evaluation shows that one trajectory contains clear logical mistakes (e.g., define $AB = x$ and write $AC = x + CD$, conflating BC with CD), while the other does not. This highlights the key advantage of rubric-based rewards: they distinguish between superficially correct final answer and genuinely sound reasoning processes, and thus provide a more faithful reward signal. Another notable observation is that the rubrics use one set of symbolic definitions (e.g., line segment lengths denoted as a and b), while the trajectories use a different definition system (e.g., x). Despite these discrepancies, the judge model aligns the se-

mantics and provides accurate assessments. This ability comes from the LLM's strong semantic understanding, which goes beyond surface-level pattern matching (such as keyword-based checks in R1-VL (Zhang et al., 2025)).

G Reproducibility Statement

We make the following effort to ensure the reproducibility of our work. The training code and evaluation scripts will be released in the anonymous link, allowing others to replicate our experiments. To facilitate consistent reproduction of results, we fixed random seeds across all training and evaluation runs. Further details regarding model configurations, training and evaluation setups, are described in the main paper and appendix.

H Prompts

For reproducibility, we present all the prompts used in this work, including the prompt for constructing rubrics from trajectories (Figure 9), the prompt for rubric-based LLM-as-A-Judge reward (Figure 10), the prompt for strict accuracy evaluation (Figure 11) (Detailed in Section 4.1) and reasoning quality evaluation (Figure 12) (Detailed in Section E.2).

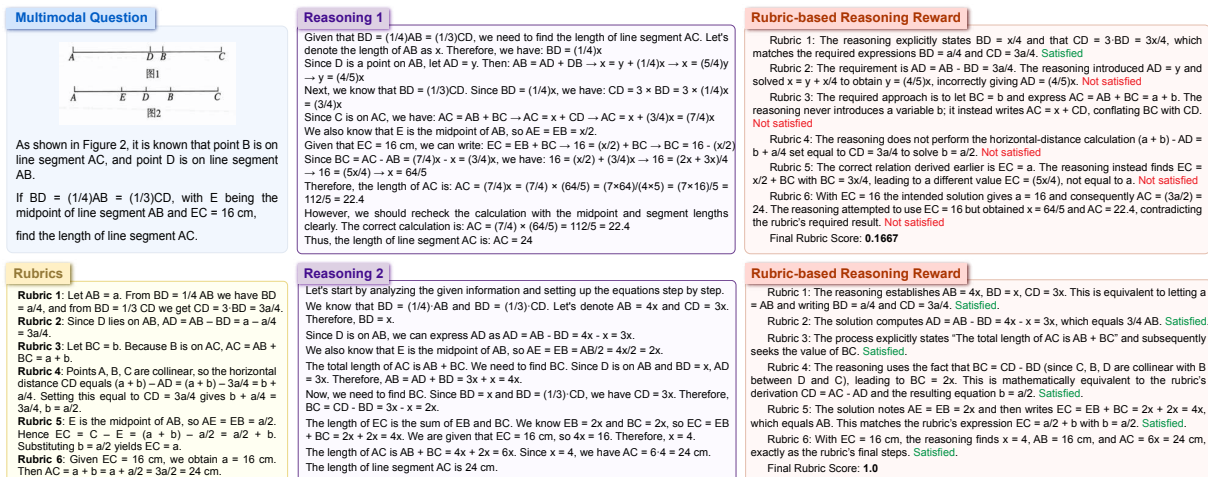


Figure 8: Illustration of a problem with the constructed rubrics, two reasoning trajectories produced by the policy model, and the rubric-based reasoning rewards assigned by the judge model.

Prompt for Constructing Rubrics from Trajectories

You are given a math question about an image and several reasoning processes. All reasoning processes arrive at the correct final answer. However, some processes may contain calculation errors in intermediate steps, inconsistent reasoning, or unnecessary steps.

CRITICAL TASK: Carefully compare multiple reasoning processes to identify which steps are ACTUALLY CORRECT. **VERIFY CALCULATIONS:** For each mathematical calculation in the reasoning processes: Check if the arithmetic is correct (e.g., 3×4 should equal 12, not 13)

- Verify algebraic manipulations are valid**
 Ensure geometric formulas are applied correctly
 If different processes have different intermediate calculations for the same step, identify which one is mathematically correct
- CROSS-VALIDATE:** When multiple processes perform the same calculation:
 If they get different intermediate results, determine which is correct
 Do NOT include incorrect calculations in the rubric, even if they appear in multiple processes
 Only include calculations that are mathematically verified to be correct
- EXTRACT ESSENTIAL CHECKPOINTS:** After verifying correctness, identify the key rubric that:
 Are mathematically correct and consistent
 Are necessary to reach the correct answer
 Include specific correct numbers, equations, and calculations
 Represent the logically sound path to the solution

IMPORTANT: The rubric must ONLY contain mathematically correct rubrics.
 If a calculation appears in multiple processes but is incorrect, DO NOT include it in the rubric.
 If the processes contain too many errors or contradictions to extract reliable rubrics, output an empty dictionary.

Please output the rubrics in JSON format as follows:

```
{
  "Rubric 1": "[content of the first necessary and CORRECT rubric]",
  "Rubric 2": "[content of the second necessary and CORRECT rubric]",
  ...
}
```

Question: (question)
 Correct Answer: (answer)
 Reasoning Processes to Analyze: (processes_text)

Return only the JSON with mathematically correct checkpoints.

Figure 9: The prompt for rubric construction.

Prompt for Rubric-based LLM-as-a-Judge Reward

You are an evaluator. Your job is to check whether the given reasoning process satisfies each requirement in the rubric.

Inputs

Problem:

{problem}

Proposed Reasoning Process:

{solution}

Evaluation Rubric:

{rubric_checkpoints}

Evaluation Procedure

For each rubric criterion:

Read the rubric criterion carefully and understand what requirement it sets.

- Recognize mathematical equivalence. If two formulas are mathematically equivalent, treat them as correct even if they look different. (E.g., " $x - y - z$ " is equivalent to " $x - (y + z)$ ")

- Be strict about the names of geometric objects. For example, $\angle OAB = 90^\circ$ is NOT the same as $\angle OAC = 90^\circ$.

Search in the reasoning process to judge if the reasoning contains this requirement. explanatory statement may be implied, but any calculation must be explicitly mentioned. If yes, mark Satisfied (1 point). If no, mark Not satisfied (0 points)."

Continue until all criteria are covered.

Calculate the total score as the fraction of rubrics satisfied.

Output Rules

For each rubric, first output a short analyse, then output the final justification in the following format:

Rubric X: [the analyse referencing the reasoning and the rubric]. [Satisfied / Not satisfied].

After all rubric criteria, output the final score as the following decimal format: Final Rubric Score: $\boxed{\{\text{score}\}}$

Figure 10: The prompt for using rubrics in LLM-as-A-Judge in training.

Prompt for Strict Accuracy Judge

You are a strict verifier. Another assistant answered a math problem and was required to first output its reasoning process, then provide the final numerical answer inside a single $\boxed{\dots}$.

Your job is ONLY to check whether the conclusion stated in the reasoning matches the final answer in the box.

If there is no final $\boxed{\dots}$ answer, treat the output as consistent and do not penalize for a missing box.

If the final boxed content is a numeric value that matches the magnitude concluded in the reasoning, allowing reasonable approximations, consider it consistent even if the reasoning includes units, directions, or symbols such as degrees that are omitted from the box.

If the final boxed content is a choice label, compare the label's associated choice text to the reasoning. It is consistent when the label corresponds to the same numeric value concluded in the reasoning.

Return a compact JSON object with the following fields:

Reasoning results: the numeric conclusion from the reasoning and the corresponding option, if any;

boxed answer: the content inside the final $\boxed{\dots}$;

consistent: true if the reasoning and final answer agree, otherwise false.

Figure 11: The prompt for judge the reasoning-answer inconsistency for calculating Strict Accuracy.

Prompt for Reasoning Quality Evaluation

You are an expert evaluator of mathematical reasoning and logical thinking. Your task is to assess the quality of the reasoning process in the given solution, focusing on identifying critical flaws that undermine the validity of the reasoning chain. You will evaluate the solution across three key dimensions to determine an overall quality score between 0 and 1, where 0 represents severely flawed reasoning and 1 represents perfectly sound reasoning. The quality assessment should consider the presence and severity of unfounded transitions or answer changes where the solution suddenly shifts to a different approach or answer without proper justification or explanation, appearing to abandon previous work without logical basis. Additionally, you should identify calculation errors including arithmetic mistakes, algebraic manipulation errors, incorrect formula applications, computational errors in numerical operations, and mistakes in unit conversions or dimensional analysis. Furthermore, evaluate the logical connectivity by examining whether each step follows logically from the previous ones, whether assumptions are clearly stated and justified, whether the reasoning maintains internal consistency throughout, and whether conclusions are properly supported by the preceding work. When evaluating, consider that minor notational inconsistencies or formatting issues should not significantly impact the score unless they create genuine ambiguity. Focus on substantive errors that affect the validity of the mathematical reasoning rather than stylistic preferences. A solution may use unconventional but valid approaches which should not be penalized if they are mathematically sound. The severity of errors should be weighted by their impact on reaching the correct conclusion, with critical errors that invalidate the entire solution having more weight than isolated minor mistakes. Remember that the presence of a single severe error such as a fundamental conceptual mistake or complete logical disconnect may warrant a very low score even if other parts are correct, while multiple minor errors that don't affect the final validity might still result in a moderate to high score.

Input/Problem:

```
```\n{input_text}\n```\n{choices_text}
```

Output/Solution:

```
```\n{output_text}\n```\n
```

Analyze this solution carefully, examining each step of the reasoning process for the three types of issues mentioned. Consider how these issues compound or interact with each other. Assess whether the reasoning maintains a clear logical thread from problem statement to conclusion. Evaluate whether calculations are performed correctly and whether any sudden shifts in approach are properly justified. After your thorough analysis, provide a quality score that reflects the overall soundness of the reasoning, where scores closer to 0 indicate severely flawed reasoning with multiple critical errors or logical disconnects, scores around 0.3-0.4 suggest significant issues that undermine the solution's validity, scores around 0.5-0.6 indicate moderate issues with some valid reasoning present, scores around 0.7-0.8 represent generally sound reasoning with minor issues, and scores above 0.8 indicate very strong reasoning with at most trivial errors. Your score should holistically capture the reliability and trustworthiness of the reasoning process as a whole.

Provide your response in the following format:

ANALYSIS: [Provide a comprehensive analysis of the reasoning, identifying specific instances of unfounded transitions, calculation errors, and logical disconnects. Quote specific parts of the solution that demonstrate these issues.]

UNFOUNDED_TRANSITIONS: [Describe any sudden changes in approach or answers without justification]

CALCULATION_ERRORS: [List specific computational or mathematical errors found]

LOGICAL_CONNECTIVITY: [Assess the overall logical flow and consistency of the reasoning]

SEVERITY_ASSESSMENT: [Explain how severe the identified issues are and their impact on the solution's validity]

QUALITY_SCORE: [A decimal number between 0 and 1 representing the overall quality of the reasoning]

Figure 12: The prompt for reasoning quality evaluation.