SOPHIAVL-R1: REINFORCING MLLMS REASONING WITH THINKING REWARD

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent advances have shown success in eliciting strong reasoning abilities in multimodal large language models (MLLMs) through rule-based reinforcement learning (RL) with outcome rewards. However, this paradigm typically lacks supervision over the thinking process leading to the final outcome. As a result, the model may learn sub-optimal reasoning strategies, which can hinder its generalization ability. In light of this, we propose SophiaVL-R1, as an attempt to add reward signals for the thinking process in this paradigm. To achieve this, we first train a thinking reward model that evaluates the quality of the entire thinking process. Given that the thinking reward may be unreliable for certain samples due to reward hacking, we propose the Trust-GRPO method, which assigns a trustworthiness weight to the thinking reward during training. This weight is computed based on the thinking reward comparison of responses leading to correct answers versus incorrect answers, helping to mitigate the impact of potentially unreliable thinking rewards. Moreover, we design an annealing training strategy that gradually reduces the thinking reward over time, allowing the model to rely more on the accurate rule-based outcome reward in later training stages. Experiments show that our SophiaVL-R1 surpasses a series of reasoning MLLMs on various benchmarks (*e.g.*, MathVisita, MMMU), demonstrating strong reasoning and generalization capabilities. Notably, our SophiaVL-R1-7B even outperforms LLaVA-OneVision-72B on most benchmarks, despite the latter having $10\times$ more parameters. All code, models, and datasets will be made publicly available.

1 INTRODUCTION

Recent advances have highlighted the potential of rule-based Reinforcement Learning (RL) to elicit reasoning capabilities of Large Language Models (LLMs) (Guo et al., 2025; Yu et al., 2025). In particular, DeepSeek-R1 (Guo et al., 2025) exemplifies the success of applying the GRPO (Shao et al., 2024) reinforcement learning algorithm to incentive strong reasoning with long Chain-of-Thought (CoT) in LLMs. Beyond text-based domains, this paradigm has also shown promising results in Multimodal Large Language Models (MLLMs), with representative models including R1-OneVision (Yang et al., 2025), OpenVLThinker (Deng et al., 2025b), and Video-R1 (Feng et al., 2025). The key of these methods is to utilize a rule-based function that yields accurate outcome reward signals for RL training (Guo et al., 2025; Leng et al., 2025; Deng et al., 2025b).

However, solely relying on the outcome reward usually fails to ensure the quality of the thinking process, which is critical for developing models with generalizable reasoning ability (Lightman et al., 2023). For example, models may produce correct answers through flawed thinking trajectories, as illustrated in Figure 1, rather than through systematic deduction. During GRPO training (Shao et al., 2024), the rule-based outcome reward will equally encourage these responses with correct answers, regardless of whether the underlying thinking process is sound or flawed. Therefore, the model may adopt sub-optimal or even wrong reasoning strategies that generalize poorly, leading to inferior performance. This gives rise to one intuitive thought: *Can we incorporate a reward for the thinking process during GRPO training to explicitly guide correct reasoning?*

To explore this question, we propose **SophiaVL-R1**, an MLLM that enhances reasoning by integrating model-generated thinking rewards with rule-based outcome rewards in RL training. Given that typical process reward models (PRMs) impose rigid step-wise constraints on reasoning and can be

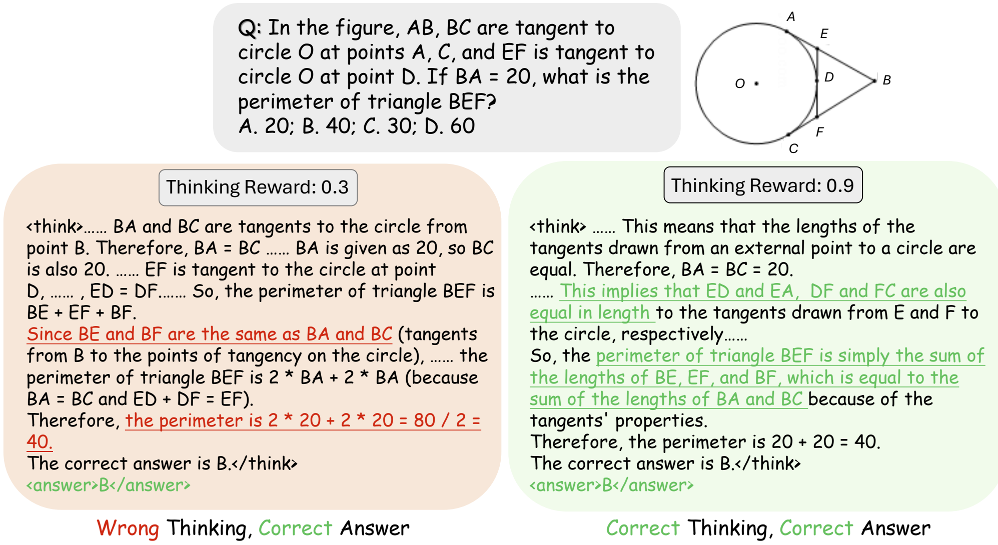


Figure 1: Examples of model responses and their corresponding thinking rewards.

overly exploited (*e.g.*, generating meaningless or repetitive steps), we measure the quality of the thinking process at a *holistic* level rather than at the *step* level. Specifically, we introduce a thinking reward model trained on annotated reasoning responses collected from GRPO training trajectories. This model evaluates intermediate reasoning quality based on criteria such as logical soundness, consistency across steps, and redundancy in the thinking process. By doing so, we provide reward signals that help the reasoning model distinguish between sound and flawed thinking processes.

Moreover, considering that the model-generated thinking rewards may be unreliable for certain cases (Ye et al., 2024; Li et al., 2025a), we propose the **Trust-GRPO** training algorithm to reduce the risks of reward hacking (Skalse et al., 2022). The core idea of Trust-GRPO is to add a trustworthiness weight to the thinking reward, which evaluates the reliability of the rewards across a group of responses to a given question. This weight is determined by comparing the thinking rewards of responses that produce correct answers with those that yield incorrect answers for the same question. A lower trustworthiness weight is assigned when high thinking rewards are abnormally given to reasoning processes that lead to incorrect answers, indicating that the reward signal may be unreliable. Unlike previous uncertainty estimation methods such as MC Dropout (Gal & Ghahramani, 2016), which usually require multiple samplings for a single response—an approach that can be computationally prohibitive for MLLMs—our method introduces no additional cost by leveraging information from the response group within GRPO. Furthermore, an annealing schedule is introduced to gradually reduce the influence of the thinking reward throughout training, allowing the model to increasingly rely on the more reliable and accurate rule-based outcome reward in later stages. In short, our proposed Trust-GRPO enables the model to receive thinking process rewards in a reliable manner, thereby guiding the exploration of favorable and generalizable reasoning strategies.

In summary, our contributions are as follows:

- We propose a thinking reward model that evaluates reasoning quality from various dimensions at a holistic level, enabling the model to distinguish between sound and flawed reasoning processes during rule-based RL training.
- We introduce the Trust-GRPO algorithm, which assigns a trustworthiness weight to thinking rewards based on their reliability. This method guides the model to explore favorable reasoning policies in a trustworthy manner without extra computational overhead.
- In the experiments, SophiaVL-R1-7B consistently outperforms existing MLLMs on diverse benchmarks (*e.g.*, MathVista, MMMU), highlighting its strong reasoning and generalization abilities. Notably, our SophiaVL-R1-7B outperforms LLaVA-OneVision-72B with $10\times$ more parameters on most benchmarks.

2 Related Work

2.1 Reward Models

Reward models (RMs) play a crucial role in guiding and shaping the behavior of models (Ouyang et al., 2022; Zhong et al., 2025). Several studies (Lightman et al., 2023; Yuan et al., 2024; Wang et al., 2025b; Zhang et al., 2025) demonstrate that process supervision—providing feedback at intermediate reasoning steps—has the potential to enhance reasoning capabilities. For example, Lightman et al. (2023) introduce powerful Process Reward Models (PRMs) with step-wise rewards, which have been applied to mathematical reasoning (Lightman et al., 2023; Wang et al., 2023). ReST-MCTS* (Zhang et al., 2024a) integrates process supervision and Monte Carlo Tree Search (MCTS) to generate per-step process rewards, enabling efficient self-training of both policy and reward models without manual annotation. Beyond the text-based domain, VisualPRM (Wang et al., 2025b) extends PRMs to the multimodal domain, achieving significant improvements in the reasoning performance of various MLLMs. Despite these advances, PRMs still face two major challenges: (1) imposing rigid step-wise constraints requires the model to strictly follow predefined reasoning steps, which can limit flexibility and generalization—particularly in general tasks (Guo et al., 2025; Cui et al., 2025); and (2) evaluating the correctness of individual steps is inherently challenging (Zhong et al., 2025), which may lead models to exploit the reward by repeating valid steps or inserting meaningless ones without making real progress. ArmoRM (Wang et al., 2024) trains a reward model to give fine-grained rewards from different perspectives in non-reasoning fields. However, it is not designed for solving the "wrong thinking, correct answer" phenomenon in R1-like training for reasoning. Therefore, in contrast to prior approaches, we aim to develop a thinking reward model that evaluates reasoning quality from multiple dimensions at a holistic level, thereby fostering a more reliable and stable training pipeline for R1-style models.

2.2 Multimodal Large Language Model Reasoning

The field of multimodal large language model reasoning aims to build human-like models capable of handling complex tasks that require understanding and reasoning across multiple modalities (Li et al., 2025c). Earlier methods typically depend on fine-grained step-level supervision or learned reward models to guide the reasoning process (Yao et al., 2024; Wang et al., 2025b; Zang et al., 2025). In contrast, DeepSeek-R1 (Guo et al., 2025) demonstrates that reinforcement learning with a rule-based reward model can effectively incentivize strong reasoning abilities without dense supervision. Following the R1 paradigm, several efforts have explored enhancing MLLM reasoning through rule-based reinforcement learning (Lai et al., 2025; Feng et al., 2025; Shen et al., 2025; Xia & Luo, 2025; Wang et al., 2025a). R1-OneVision (Yang et al., 2025) introduces a cross-modal reasoning pipeline and adopts a supervised fine-tuning followed by RL strategy to strengthen reasoning capabilities. Curr-ReFT (Wu et al., 2025) introduces a curriculum-based reinforcement learning paradigm for small-scale MLLMs, combining difficulty-aware rewards and rejection sampling to boost generalization. Video-R1 (Feng et al., 2025) proposes T-GRPO algorithm to explicitly encourage temporal reasoning in video. Despite their success on multimodal tasks, these approaches rely exclusively on outcome rewards, which often overlook the quality of intermediate reasoning steps.

3 Method

3.1 Dataset Composition

We curate a dataset SophiaVL-R1-130k, comprising 130k examples to support the training of thinking reward model (Section 3.2) and SophiaVL-R1 (Section 3.4). To overcome the scarcity of high-quality multimodal reasoning data and ensure robust model performance across a wide range of tasks, we aggregate samples from a combination of text-only and multimodal datasets, all of which are publicly available. The dataset contains both reasoning-specific tasks and general vision-language understanding tasks. We organize the data into five categories, covering diverse reasoning scenarios, as illustrated in Figure 2 (left).

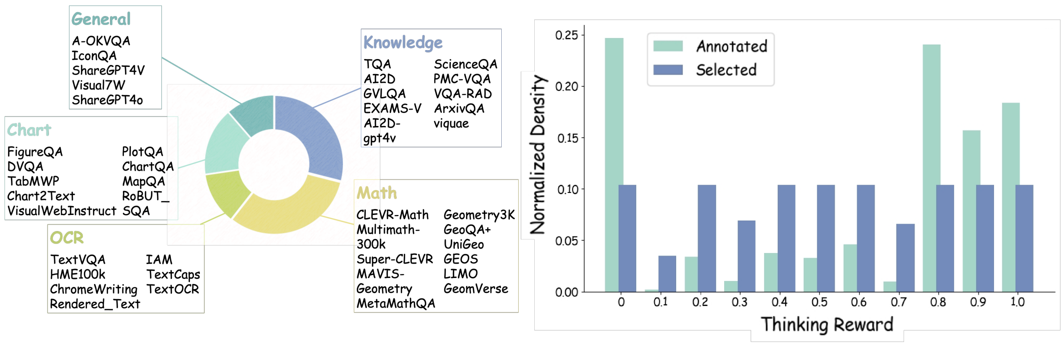


Figure 2: **Left:** Composition of our SophiaVL-R1-130k dataset from public sources. **Right:** Distribution of the SophiaVL-R1-Thinking-156k dataset used to train the thinking reward model.

3.2 Thinking Reward

To assess fine-grained reasoning quality of MLLMs' thinking process, we develop a thinking reward model that assigns a score between 0 and 1 based solely on the quality of intermediate reasoning, regardless of whether the final answer is correct.

To construct the dataset used for training the thinking reward model, we collected 470,331 (`question, response`) pairs output by Qwen2.5-VL-7B-Instruct (Bai et al., 2025) during the GRPO training on the SophiaVL-R1-130k dataset. These data contain both favorable and flawed reasoning patterns occurred in the training. Then, each response is scored by the advanced MLLM, Qwen2.5-VL-72B-Instruct (Bai et al., 2025), using the prompt in Appendix A. This results in 470,331 (`question, response, thinking reward`) tuples. The evaluation is based on five dimensions, which are identified from error patterns observed during GRPO training: Logical Soundness, Correct Reasoning, Error Identification, Language Consistency, and Redundancy. Detailed examples of error patterns are provided in Appendix B.

To ensure the quality of labels and maintain a balanced distribution across different reward levels, we apply manually designed rule-based filtering (Details in Appendix D) to remove noisy samples and perform uniform sampling to preserve distribution balance. This process results in 156,703 high-quality annotated samples. with 5,000 to 15,000 samples per interval. Each reward interval corresponds to a discrete range (*e.g.*, [0.0–0.1), [0.1–0.2), ..., [0.9–1.0]). The distribution of the full (*Annotated*) and balanced (*Selected*) datasets is shown in Figure 2 (right). We denote the resulting dataset as SophiaVL-R1-Thinking-156k, with its detailed composition reported in Appendix C.

The thinking reward model, initialized with Qwen2.5-VL-3B-Instruct (Bai et al., 2025), is then trained on this dataset using SFT, where the model is required to output a thinking reward given a question and its corresponding thinking process. Through this training, the thinking reward model learns to identify diverse reasoning errors and assign appropriate rewards accordingly, thereby playing a crucial role in GRPO training by providing feedback on reasoning quality.

3.3 Rule-based Outcome Reward

Following DeepSeek-R1 (Guo et al., 2025), we construct rule-based outcome reward functions to generate reward signals. Specifically, we design task-specific functions that assess model outputs by comparing them with ground-truth answers. Tasks are categorized based on their output formats: (1)**Numerical**: A binary reward is assigned based on an exact match between the predicted and ground-truth values; (2) **Multiple Choice**: The reward is defined based on whether the model's output matches the ground-truth choice; (3) **OCR**: The reward is computed as the negative Word Error Rate (WER), penalizing transcription inaccuracies; (4) **Free-form Text**: The reward is calculated as the average of ROUGE-1, ROUGE-2, and ROUGE-L scores, measuring n-gram and sequence-level similarity (Feng et al., 2025).

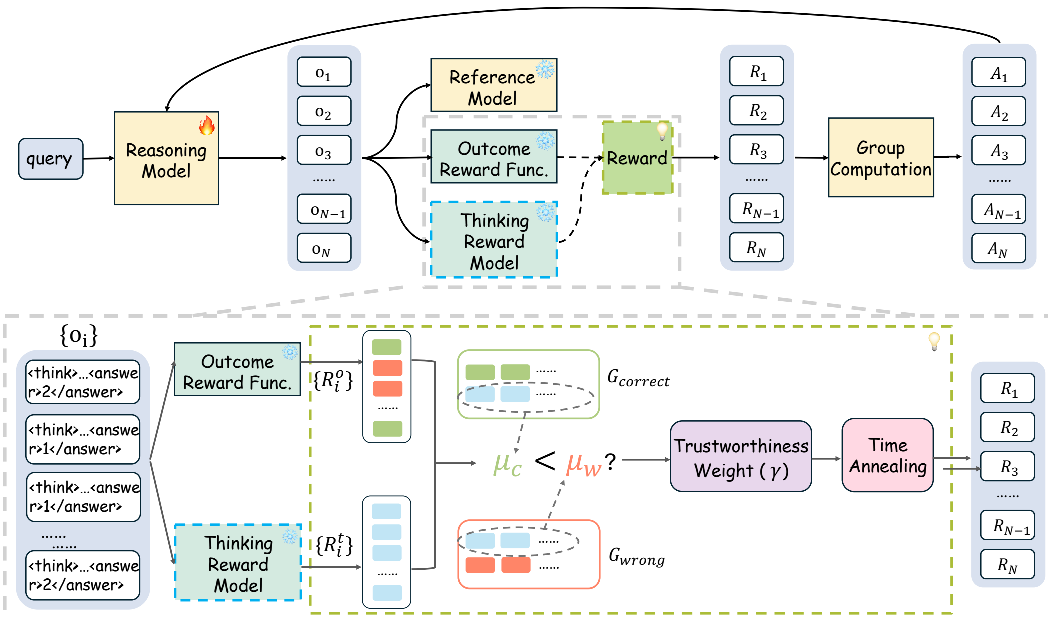


Figure 3: An illustration of our proposed Trust-GRPO.

3.4 Trustworthy Group Relative Policy Optimization (Trust-GRPO)

As discussed earlier, integrating the thinking reward into GRPO training could help the model distinguish between favorable and flawed reasoning process. Nevertheless, a direct application may result in reward hacking, given that model-generated rewards are not always trustworthy. To deal with this challenge, we introduce the Trust-GRPO algorithm, as illustrated in Figure 3.

Trust-GRPO optimizes the policy using a combination of two reward types: (1) thinking reward R^t (Section 3.2) that assigns a score between 0 and 1 based on holistic reasoning quality, and (2) outcome reward R^o (Section 3.3), derived from rule-based evaluation of outcome answer correctness. To reduce the risk of reward hacking, a trustworthiness weight γ is included to determine the influence of thinking reward R^t.

The trustworthiness is computed by contrasting the thinking reward R^t assigned to responses that arrive at correct answers with those leading to incorrect ones. When higher thinking rewards are abnormally associated with incorrect reasoning, γ will be lower, indicating the potential unreliability in the reward signal. Next, we will introduce how to derive it.

First, responses o_i to a question q are grouped into correct answer group G_{correct} and wrong answer group G_{wrong} based on their outcome rewards. Then, we calculate the average thinking reward in G_{correct} and G_{wrong} as follows:

$$\mu_c = \frac{1}{|G_{\text{correct}}|} \sum_{i \in G_{\text{correct}}} R_i^t, \quad G_{\text{correct}} = \{i \mid R_i^o \geq 0.5\}, \tag{1}$$

$$\mu_w = \frac{1}{|G_{\text{wrong}}|} \sum_{i \in G_{\text{wrong}}} R_i^t, \quad G_{\text{wrong}} = \{i \mid R_i^o < 0.5\}, \tag{2}$$

where μ_c and μ_w denote the average thinking rewards in the correct answer group and the wrong answer group, respectively. R_i^o denotes the outcome reward of response i. The trustworthiness weight γ is defined as follows:

$$\gamma = \begin{cases} 1, & \mu_c \geq \mu_w \\ e^{\mu_c - \mu_w}, & \mu_c < \mu_w \end{cases}. \tag{3}$$

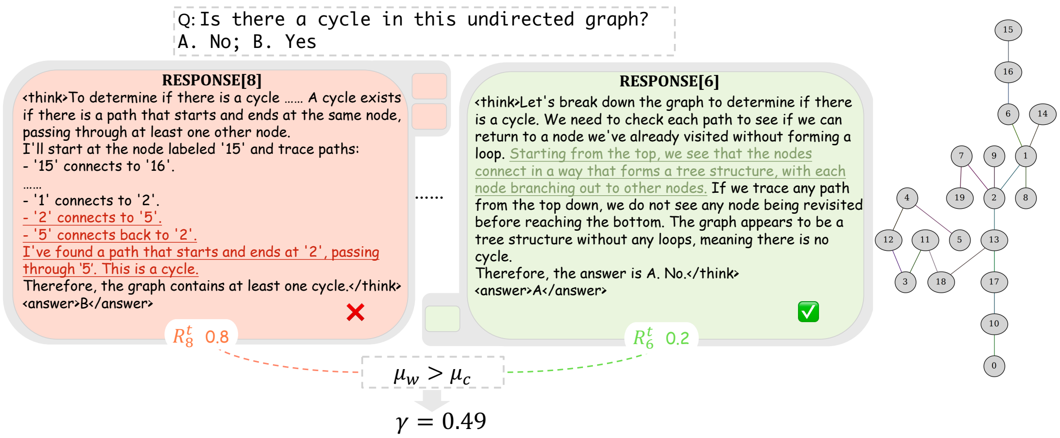


Figure 4: Example of trustworthiness weight γ. Incorrect responses (red) receive higher average thinking rewards than correct ones (green), indicating misalignment between R^t and R^o and the need for a trustworthiness-aware adjustment.

This comparison between μ_c and μ_w allows us to assess the alignment between thinking rewards and rule-based outcome rewards. A lower γ indicates a discrepancy between R^t and R^o, suggesting that the thinking reward may be unreliable for this response group and thus should be given reduced weight. γ dynamically estimates the trustworthiness of thinking rewards for each question's response group without incurring additional computational overhead. This average reward–based design is chosen for its simplicity and efficiency, which are particularly important given the high cost of training and inference of MLLMs. Additional analyses of alternative design are provided in Appendix E.1.

The i-th reward R_i incorporating the thinking reward with trustworthiness weight is defined as:

$$R_i = R_i^o + \gamma\alpha \cdot R_i^t, \tag{4}$$

where α is a hyperparameter that controls the impact of thinking reward.

We further introduce a time-based annealing strategy that gradually reduces the influence of thinking reward as training progresses. This encourages the reasoning model to rely increasingly on the more accurate rule-based outcome reward in later steps. Combining both components, the final reward R_i is defined as:

$$R_i = R_i^o + \gamma\alpha e^{-\frac{\texttt{steps}}{T}} \cdot R_i^t, \tag{5}$$

where `steps` denotes the current global training step and T is the total number of training steps, controlling the decay rate of thinking reward's influence over time. Additional analysis of the time-based annealing strategy design is provided in Appendix E.2.

The advantage A_i is computed using rewards of each response group:

$$A_i = \frac{R_i - \text{mean}(\{R_1, R_2, \cdots, R_N\})}{\text{std}(\{R_1, R_2, \cdots, R_N\})}, \tag{6}$$

Following DeepSeek-R1 (Guo et al., 2025), given a question q, GRPO samples responses $o_1, \ldots, o_N$ from the old policy π_{old}, and updates the policy π_θ by maximizing the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}\left[q \sim P(Q),\ \{o_i\}_{i=1}^N \sim \pi_{\text{old}}(O|q)\right]$$

$$\frac{1}{N}\sum_{t=1}^{N}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}A_i,\ \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}, 1-\epsilon,\ 1+\epsilon\right)A_i\right) - \beta\,\mathbb{D}_{\text{KL}}[\pi_\theta \| \pi_{\text{ref}}]\right). \tag{7}$$

By contrasting the thinking rewards of correct and incorrect responses, Trust-GRPO improves the reliability of reward signals, thereby encouraging more generalizable reasoning behavior.

Figure 4 illustrates a case where the trustworthiness weight γ helps identify potentially unreliable thinking rewards. Responses with incorrect answers are shown in red and those with correct answers in green. Despite being incorrect, the red group receives a higher average thinking reward, indicating a misalignment between R^t and R^o. This implies that the thinking reward has potential risk of unreliability, thus should be assigned less weight. More examples can be found in Appendix L.

Table 1: Comparison of models on **MathVista** and **MathVerse**. The best is **bold**, and the runner-up is underline. [1]Scientific Reasoning, [2]Textbook Question Answering, [3]Arithmetic Reasoning, [4]Math Word Problem, [5]Logical Reasoning, [6]Vision Intensive, [7]Vision Only, [8]Vision Dominant, [9]Text Dominant, [10]Text Lite.

Model	MathVista						MathVerse					
	AVG	SCI[1]	TQA[2]	ARI[3]	MWP[4]	LOG[5]	AVG	VI[6]	VO[7]	VD[8]	TD[9]	TL[10]
General MLLMs												
LLaVA-OneVision-7B (Li et al., 2024)	63.2	65.6	60.8	57.8	69.4	21.6	26.2	-	-	-	-	-
LLaVA-OneVision-72B (Li et al., 2024)	68.4	63.1	65.8	60.1	73.7	27.1	27.2	-	-	-	-	-
Cambrian-1-34B (Tong et al., 2024)	50.9	53.3	55.1	45.6	51.6	16.2	-	-	-	-	-	-
GPT-4V	51.8	63.1	65.8	51.8	57.5	21.6	32.8	-	-	-	-	-
Open-Source Math MLLMs												
Math-LLaVA-13B (Shi et al., 2024)	46.6	49.2	51.3	40.2	56.5	16.2	22.9	24.5	16.1	21.7	27.3	24.9
Math-PUMA-Qwen2VL-7B (Zhuang et al., 2025)	47.9	42.6	46.2	46.2	68.3	21.6	33.6	33.4	26.0	31.6	42.1	35.0
Multimath-7B (Peng et al., 2024)	50.0	-	50.0	-	61.8	-	26.9	28.1	15.0	25.9	34.8	30.8
URSA-8B (Luo et al., 2025)	59.8	58.2	63.9	53.5	75.3	21.6	45.7	**46.4**	34.6	43.9	55.3	48.3
Open-Source Reasoning MLLMs												
Curr-ReFT-7B (Deng et al., 2025a)	64.5	-	-	-	-	-	-	-	-	-	-	-
R1-OneVision-7B (Yang et al., 2025)	64.1	61.5	62.0	56.1	64.5	16.2	46.4	-	40.0	-	-	-
InternVL2.5-8B-VisualPRM (Wang et al., 2025b)	68.5	61.5	53.9	45.9	66.8	21.2	30.7	28.9	35.8	27.3	31.7	29.7
Qwen2.5-VL-7B-Instruct (Bai et al., 2025)	67.5	65.6	67.7	57.5	69.4	27.0	44.0	41.1	41.0	38.7	55.2	44.0
+GRPO	69.9	68.0	69.6	61.2	75.8	24.3	45.3	43.0	41.0	41.1	56.0	45.6
+SFT+GRPO	66.8	**72.1**	73.4	59.8	69.9	21.6	43.1	42.5	37.1	37.3	52.2	46.3
SophiaVL-R1-7B	**71.3**	70.5	**73.4**	**62.6**	**76.9**	**35.1**	**48.8**	45.4	**43.9**	**45.1**	**58.5**	**51.3**

4 EXPERIMENT

4.1 EXPERIMENT SETTINGS

Benchmarks. We evaluate our model on both multimodal mathematical reasoning and general multimodal reasoning benchmarks. For mathematical reasoning, we report detailed results on MathVista (Lu et al., 2023) and MathVerse (Zhang et al., 2024b). For general multimodal capabilities, we conduct evaluations on MMMU (Yue et al., 2024), MME (Liang et al., 2024), MMStar (Chen et al., 2024), ChartQA (Masry et al., 2022), and MMBench (Xu et al., 2023).

Implementation Details. The thinking reward model is initialized from Qwen2.5-VL-3B-Instruct and trained for 2 epochs with SFT on SophiaVL-R1-Thinking-156k using 4 NVIDIA A800 80GB GPUs. The reasoning model is initialized from Qwen2.5-VL-7B-Instruct and trained on SophiaVL-R1-130k with the Trust-GRPO algorithm. RL training is performed for 1,500 steps using a VeRL (Zheng et al., 2025; Sheng et al., 2024)-based implementation on 8 NVIDIA A800 80GB GPUs. Hyperparameters for RL training are provided in Appendix J. For evaluation, we use default prompts to generate responses. Additional evaluation details are given in Appendix K.

4.2 MAIN RESULTS

Performance on Math Reasoning Benchmarks. As shown in Table 1, SophiaVL-R1-7B achieves competitive performance on mathematical reasoning benchmarks. On the MathVista benchmark, it attains an accuracy of 71.3%, surpassing both Qwen2.5-VL-7B-Instruct models trained with GRPO and SFT+GRPO, and also outperforming the LLaVA-OneVision-72B model. Compared to the model trained by VisualPRM (Wang et al., 2025b), our model achieves significantly better performance, with an 18.1-point improvement on MathVerse (48.8 vs. 30.7), and consistently outperforms it across all sub-tasks. These results indicate that, compared to PRM-based method, our Trust-GRPO may serve as a more effective approach for providing reward signals, better guiding the model toward improved reasoning ability.

Performance on General Benchmarks. Many task-specific reasoning models, such as those optimized for mathematical problem-solving or other specialized tasks, excel within their respective

Table 2: Comparison on general ability benchmarks. The best is **bold**, and the runner-up is underline.

Model	MMMU	MME	ChartQA	MMBench	MMStar
General MLLMs					
LLaVA-OneVision-7B (Li et al., 2024)	48.8	1998.0	80.0	-	61.7
LLaVA-OneVision-72B (Li et al., 2024)	56.8	2261.0	83.7	-	<u>66.1</u>
Cambrian-1-34B (Tong et al., 2024)	49.7	1689.3	75.6	81.4	54.2
GPT-4V	56.8	1926.0	78.5	75.0	57.1
Open-Source Math MLLMs					
URSA-8B (Luo et al., 2025)	43.1	1605.7	44.4	55.5	42.3
Open-Source Reasoning MLLMs					
Curr-ReFT-7B (Deng et al., 2025a)	-	-	-	79.0	-
R1-Onevision-7B (Yang et al., 2025)	51.6	2223.3	-	75.6	59.1
InternVL2.5-8B-VisualPRM (Wang et al., 2025b)	56.2	-	60.8	83.5	63.4
Qwen2.5-VL-7B-Instruct (Bai et al., 2025)	58.7	2306.0	86.3	83.3	64.3
+GRPO	58.0	2298.2	87.2	83.4	65.6
+SFT+GRPO	<u>59.1</u>	<u>2344.1</u>	**89.2**	<u>84.6</u>	64.7
SophiaVL-R1-7B	**61.3**	**2403.8**	<u>88.5</u>	**85.4**	**66.7**

Table 3: Performance of reward models on VLRewardBench.

Model	General	Hallucination	Reasoning	Overall Accuracy	Macro Accuracy
Qwen2.5-VL-3B-Instruct	34.4	42.1	51.5	43.1	43.0
GPT-4o-mini	41.7	34.5	58.2	41.5	44.8
Qwen2-VL-72B	38.1	32.8	58.0	39.5	43.0
Our Thinking Reward Model (3B)	45.4	46.8	54.4	48.6	48.9

domains but often struggle to maintain strong performance on general multimodal benchmarks (*e.g.*, URSA-8B). Different from them, SophiaVL-R1-7B demonstrates consistently strong performance across widely recognized general ability benchmarks, as shown in Table 2, highlighting its superior generalization capability. For example, on the widely used MMMU benchmark for multi-discipline reasoning, SophiaVL-R1-7B outperforms LLaVA-OneVision-72B by 4.5 points.

4.3 Performance of Thinking Reward Model

To further evaluate the capability of our thinking reward model, we conduct experiments on VLRewardBench (Li et al., 2025b), a benchmark designed to assess multimodal reward models.

As shown in Table 3, our 3B thinking reward model achieves higher performance despite having significantly fewer parameters. In particular, it demonstrates strong performance in detecting Hallucination, indicating that it effectively distinguishes reliable from unreliable responses.

5 Ablation Study

We conduct ablation studies to examine the contributions of key components in our method. Specifically, we evaluate three variants of our SophiaVL-R1:

- **SophiaVL-R1-wo-trained-TRM**: replacing the trained thinking reward model with an untrained Qwen2.5-VL-3B-Instruct model.
- **SophiaVL-R1-wo-trust-and-annealing**: removing both the trustworthiness weighting and the annealing strategy from Trust-GRPO.

Table 4: Ablation Study.

Model	MathVista	MathVerse	MMMU	MME	ChartQA	MMBench	MMStar
Qwen2.5-VL-7B+GRPO	69.9	45.3	58.0	2298.2	87.2	83.4	65.6
SophiaVL-R1-wo-trained-TRM	68.4	47.9	57.0	2347.1	87.7	84.0	65.7
SophiaVL-R1-wo-trust-and-annealing	67.4	46.3	56.7	2366.8	86.3	82.6	65.0
SophiaVL-R1-wo-trust	70.2	47.8	60.0	2363.3	87.8	83.7	65.2
SophiaVL-R1	**71.3**	**48.8**	**61.3**	**2403.8**	**88.5**	**85.4**	**66.7**

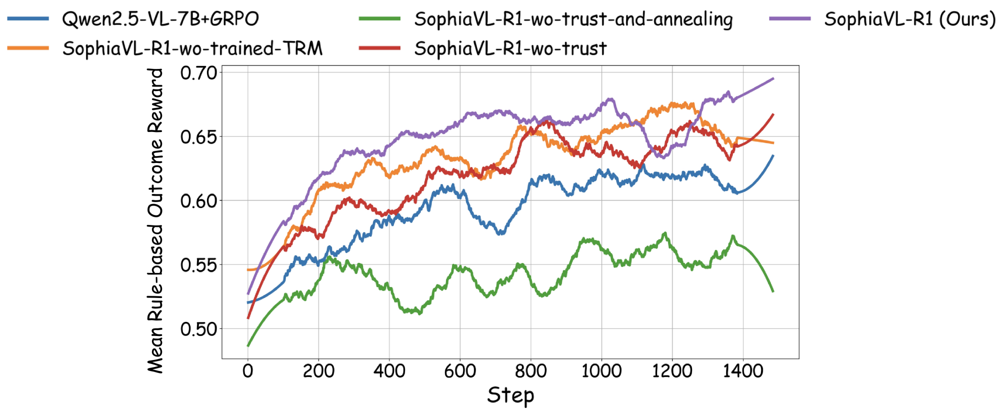


Figure 5: Training curves of mean rule-based outcome reward across different methods.

- **SophiaVL-R1-wo-trust**: removing only the trustworthiness weight while retaining the time-based annealing schedule.

Besides, we also include **Qwen2.5-VL-7B+GRPO** as a baseline, which directly uses GRPO for training Qwen2.5-VL-7B-Instruct. The results are summarized in Table 4.

Effect of the Thinking Reward Model. SophiaVL-R1-wo-trained-TRM consistently underperformances SophiaVL-R1. This highlights the effectiveness of our training pipeline and the SophiaVL-R1-Thinking-156k dataset in improving thinking reward model's ability to provide accurate and informative reward signals for reasoning optimization. What's more, SophiaVL-R1-wo-trained-TRM performs comparably to the Qwen2.5-VL-7B+GRPO. This indicates that an untrained reward model provides limited guidance. In contrast, our trained thinking reward model substantially improves the model performance, which highlights its importance in our method.

Effect of the Trustworthiness Weight γ. We observe a performance drop across all benchmarks in SophiaVL-R1-wo-trust when the trustworthiness weight is removed, compared to the full SophiaVL-R1 model. This demonstrates the effectiveness of trustworthiness weighting, which allows the model to receive thinking process rewards in a more reliable manner.

Effect of the Time-based Annealing Strategy. To assess the effect of time-based annealing, we compare SophiaVL-R1-wo-trust-and-annealing with SophiaVL-R1-wo-trust. SophiaVL-R1-wo-trust-and-annealing generally performs worse on most benchmarks. The performance drop may be due to the over-exploitation of the thinking reward, where potentially unreliable signals could interfere with the optimization of the reasoning policy. This suggests that gradually reducing the influence of the thinking reward by our proposed annealing strategy is beneficial, as it encourages reliance on the more reliable rule-based outcome reward in later training stages.

Training Curve Analysis. Figure 5 shows the mean outcome reward per training step for each method. SophiaVL-R1 achieves the highest reward and demonstrates faster improvement during training. Besides, we notice that directly combining thinking and outcome rewards (SophiaVL-R1-wo-trust-and-annealing) performs worse in training, indicating the effectiveness and necessity of our trustworthiness weighting and time-based annealing strategy. Overall, these results underscore the importance of both Trust-GRPO and the thinking reward model.

6 CONCLUSION

In this work, we propose SophiaVL-R1, a multimodal large language model trained using a novel Trust-GRPO algorithm that integrates model-generated thinking rewards with rule-based outcome rewards. To promote generalizable reasoning, we introduce a holistic-level thinking reward model that assesses the quality of reasoning processes. Furthermore, we mitigate the challenge of reward hacking by introducing a trustworthiness weighting mechanism together with a time-based annealing strategy. Experimental results across multiple benchmarks demonstrate that SophiaVL-R1 consistently outperforms existing MLLMs. Our findings highlight the value of thinking process supervision beyond final correctness and offer insights for future studies on developing reasoning models.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *arXiv preprint arXiv:2503.07065*, 2025a.

Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv preprint arXiv:2503.17352*, 2025b.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pp. 11198–11201, 2024.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Benyou Wang, and Xiangyu Yue. Video-r1: Reinforcing video reasoning in mllms. *arXiv preprint arXiv:2503.21776*, 2025.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yuxiang Lai, Jike Zhong, Ming Li, Shitian Zhao, and Xiaofeng Yang. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025.

Sicong Leng, Jing Wang, Jiaxi Li, Hao Zhang, Zhiqiang Hu, Boqiang Zhang, Hang Zhang, Yuming Jiang, Xin Li, Deli Zhao, Fan Wang, Yu Rong, Aixin Sun, and Shijian Lu. Mmr1: Advancing the frontiers of multimodal reasoning. `https://github.com/LengSicong/MMR1`, 2025.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

Dawei Li, Renliang Sun, Yue Huang, Ming Zhong, Bohan Jiang, Jiawei Han, Xiangliang Zhang, Wei Wang, and Huan Liu. Preference leakage: A contamination problem in llm-as-a-judge. *arXiv preprint arXiv:2502.01534*, 2025a.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, et al. Vl-rewardbench: A challenging benchmark for vision-language generative reward models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24657–24668, 2025b.

Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyuan Jiang, Xintong Wang, Jifang Wang, et al. Perception, reason, think, and plan: A survey on large multimodal reasoning models. *arXiv preprint arXiv:2505.04921*, 2025c.

Zijing Liang, Yanjie Xu, Yifan Hong, Penghui Shang, Qi Wang, Qiang Fu, and Ke Liu. A survey of multimodel large language models. In *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, pp. 405–409, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.

Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyao Yu, Xinzhe Ni, Zicheng Lin, Jin Zeng, and Yujiu Yang. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv preprint arXiv:2501.04686*, 2025.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Shuai Peng, Di Fu, Liangcai Gao, Xiuqin Zhong, Hongguang Fu, and Zhi Tang. Multimath: Bridging visual and mathematical reasoning for large language models. *arXiv preprint arXiv:2409.00147*, 2024.

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

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

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. *arXiv preprint arXiv:2406.12845*, 2024.

Junke Wang, Zhi Tian, Xun Wang, Xinyu Zhang, Weilin Huang, Zuxuan Wu, and Yu-Gang Jiang. Simplear: Pushing the frontier of autoregressive visual generation through pretraining, sft, and rl. *arXiv preprint arXiv:2504.11455*, 2025a.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*, 2025b.

Jinyang Wu, Mingkuan Feng, Shuai Zhang, Ruihan Jin, Feihu Che, Zengqi Wen, and Jianhua Tao. Boosting multimodal reasoning with mcts-automated structured thinking. *arXiv preprint arXiv:2502.02339*, 2025.

Xiaobo Xia and Run Luo. Gui-r1: A generalist r1-style vision-language action model for gui agents. *arXiv preprint arXiv:2504.10458*, 2025.

Cheng Xu, Xiaofeng Hou, Jiacheng Liu, Chao Li, Tianhao Huang, Xiaozhi Zhu, Mo Niu, Lingyu Sun, Peng Tang, Tongqiao Xu, et al. Mmbench: Benchmarking end-to-end multi-modal dnns and understanding their hardware-software implications. In *2023 IEEE International Symposium on Workload Characterization (IISWC)*, pp. 154–166. IEEE, 2023.

Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*, 2024.

Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv preprint arXiv:2410.02736*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9556–9567, 2024.

Yuhang Zang, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Ziyu Liu, Shengyuan Ding, Shenxi Wu, Yubo Ma, Haodong Duan, Wenwei Zhang, et al. Internlm-xcomposer2. 5-reward: A simple yet effective multi-modal reward model. *arXiv preprint arXiv:2501.12368*, 2025.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772, 2024a.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2024b.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025.

Yaowei Zheng, Junting Lu, Shenzhi Wang, Zhangchi Feng, Dongdong Kuang, and Yuwen Xiong. Easyr1: An efficient, scalable, multi-modality rl training framework. `https://github.com/hiyouga/EasyR1`, 2025.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. *arXiv preprint arXiv:2504.12328*, 2025.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 26183–26191, 2025.

A PROMPT USED FOR EVALUATING THINKING PROCESS QUALITY

Table 5: Prompt for evaluating thinking process quality.

Input	{Image}, {Question} and {Model Response}

You are an expert reasoning evaluator. I will give you a multimodal question and an answer. Your goal is to judge a reward process and give a score between 0 and 1. You should focus on whether the reasoning process is good rather than whether the final answer is correct.

Evaluation Criteria:

1. Logical Soundness	Does each step follow logically from the previous one?
2. Correct Reasoning	Are the methods and steps used appropriate and valid? Are the facts and lemmas correctly stated and applied?
3. Error Identification	Are there logical flaws, unsupported assumptions, or incorrect steps?
4. Language Consistency	Is the reasoning process conducted in a single, consistent language without mixing different languages?
5. Redundancy	Is the reasoning concise, avoiding repetition or irrelevant steps?

Provide a single score from $\{\mathbf{0, 0.1, 0.2, \ldots, 1.0}\}$ based on the reasoning quality, where:

- 0	Completely flawed reasoning.
- 1	Perfectly sound reasoning.
- Intermediate	Reflect partial correctness or minor errors (*e.g.*, 0.3 for significant flaws, 0.7 for minor errors).

Be strict, reward the good process and punish the bad one. You should only output the score without any explanation.

B ERROR PATTERNS OBSERVED IN GRPO TRAINING

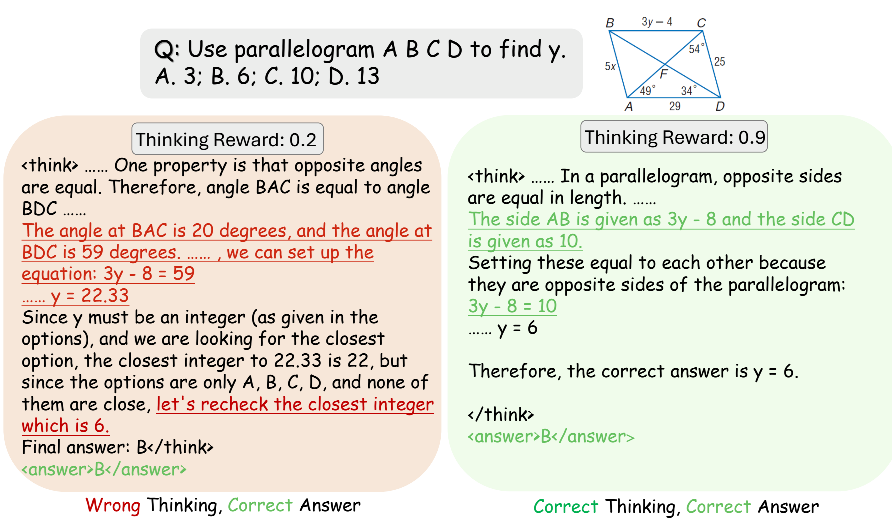


Figure 6: Examples of wrong thinking and performance of thinking reward model.

We present additional examples illustrating error patterns we observed in the reasoning process during GRPO training. Figure 6 exemplifies the **Error Identification** pattern, where the model misinterprets visual conditions by confusing edge length information with angle values, ultimately leading to incorrect reasoning. Figure 7 highlights two typical issues: the **Logical Soundness** and

Error Identification patterns. In this case, the model incorrectly extracts relationships between angles and performs faulty equation calculations.

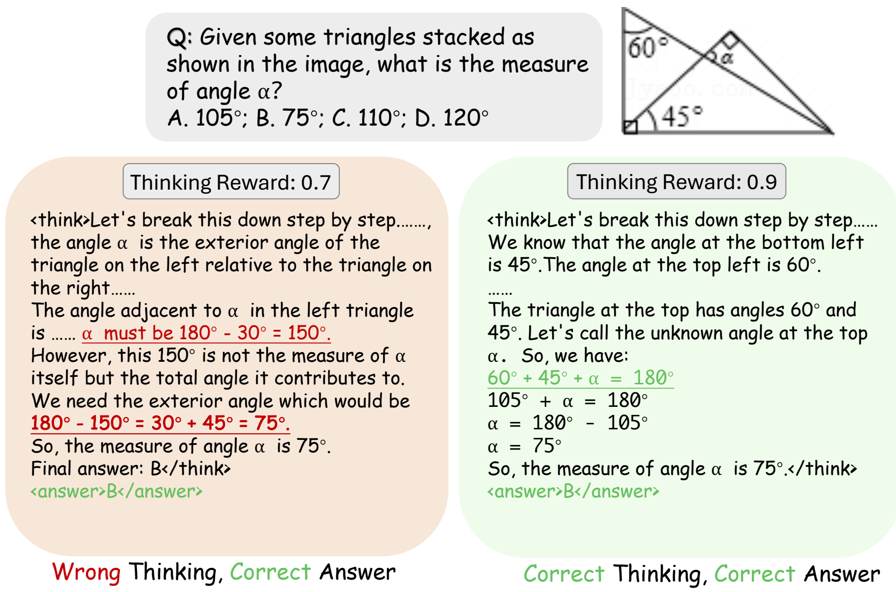


Figure 7: Examples of wrong thinking and performance of thinking reward model.

C DETAILED COMPOSITION OF SOPHIAVL-R1-THINKING-156K

Figure 8: Dataset composition and distribution of SophiaVL-R1-Thinking-156k

D SophiaVL-R1-Thinking-156k Filtering Rules for Construction

The manually designed rule-based filtering criteria described in Section 3.2 are as follows:

- Removing samples with incomplete reasoning steps, where the reasoning sequence was either cut off due to length limits or disrupted by unintended repetitive outputs of the model.
- Filtering out overly short responses (fewer than 20 words), which account for approximately 30% of the initial data. These responses mainly contain only final answer without reasoning. After filtering, about 80% of these short samples were removed.
- Discarding corrupted or meaningless responses, such as those containing random characters or mixed languages, since they provide no useful signal for training the thinking reward model.
- Applying uniform sampling to balance the distribution of reward scores across different intervals, ensuring even coverage of reasoning quality levels.

These filtering and balancing steps help maintain the quality and diversity of the SophiaVL-R1-Thinking-156k dataset used for training the thinking reward model.

E Additional Analyses on Reward Design

In this section, we provide experiments to examine two key algorithmic design choices in Trust-GRPO: the formulation of the trustworthiness weight and the annealing schedule for thinking rewards.

E.1 Average Reward-based Trustworthiness Weight Design

The trustworthiness weight γ is introduced to scale the thinking reward according to its reliability. Our design motivation is to provide a simple and efficient estimation tailored to GRPO without introducing additional computational cost, which is important given the high cost of training and inference in MLLMs.

Our design uses an average reward–based trustworthiness weight because it provides an estimation of reliability without introducing extra computation. We compared this choice with an alternative variance-based formulation to verify its justification. Specifically, for each response we sample three thinking rewards (r_1, r_2, r_3) and computed the variance of these thinking rewards. A higher variance indicates greater uncertainty, and thus a lower trustworthiness. The weight γ is defined as:

$$\gamma = \exp\left(-\frac{1}{3}\sum_{i=1}^{3}\left(r_i - \frac{1}{3}\sum_{j=1}^{3} r_j\right)^2\right).$$

Table 6 reports the results on MathVista (Math) and MMBench (General). While the variance-based approach provides an alternative measure of trustworthiness, it underperforms our original average reward–based method and incurs additional computation. These results confirm that our proposed formulation achieves a favorable balance between effectiveness and efficiency.

Table 6: Comparison between variance-based and mean reward–based(ours) trustworthiness weight.

Model	**MathVista (Math)**	**MMBench (General)**
Qwen2.5-VL-7B-Instruct	67.5	83.3
SophiaVL-R1-7B (variance)	69.1	85.1
SophiaVL-R1-7B	71.3	85.4

E.2 Decay Schedule Design of Trust-GRPO

The thinking reward provides guidance on the quality of intermediate reasoning. This signal is particularly valuable in the early stages of training, when correct reasoning does not always yield the right answer, and incorrect reasoning may occasionally arrive at the correct answer by chance. However, as training progresses, outcome rewards generally become more reliable and stable. To balance these two sources of rewards, we adopt a time-based decay schedule that gradually reduces the influence of the thinking reward. This design ensures that early updates are guided by intermediate reasoning signals, while later updates increasingly on the more reliable outcome reward.

To examine the sensitivity of Trust-GRPO to the choice of decay schedule, we compared the default exponential decay with a linear decay schedule that spans the same range of weights over the training process. The evaluation was performed on MathVista (Math) and MMBench (General), and the results are summarized in Table 7.

Table 7: Performance comparison of linear and exponential decay schedules for the thinking reward.

Model	MathVista (Math)	MMBench (General)
Qwen2.5-VL-7B-Instruct	67.5	83.3
SophiaVL-R1-7B (linear decay)	70.2	84.1
SophiaVL-R1-7B	71.3	85.4

The results indicate that both exponential and linear decay schedules improve performance relative to the instruct baseline, demonstrating that the inclusion of a decay mechanism is crucial. The exponential schedule yields slightly better performance in our experiments, but the linear schedule achieves comparable gains, suggesting that the precise functional form is less important than the principle of gradually reducing the thinking reward. More sophisticated strategies, such as learned or reward-gated schedules, may offer additional improvements and are left for future research.

F Reward Model Scaling Up Study

To explore the impact of reward model size, we further train on Qwen2.5-VL-32B-Instruct with our SophiaVL-R1-Thinking-156k dataset and use it as thinking reward model to train our SophiaVL-R1-7B. Results of reward models are shown in Table 8. Results of reasoning models are shown in Table 9.

Table 8: Performance of 32B reward models on VLRewardBench.

Model	General	Hallucination	Reasoning	Overall Accuracy	Macro Accuracy
Qwen2.5-VL-32B-Instruct	41.5	60.6	60.3	57.7	54.1
Our Thinking Reward Model (32B)	45.9	65.7	60.4	61.4	57.3

Table 9: Comparison between 7B reasoning models trained with different reward models.

Model	MathVista (Math)	MMBench (General)
SophiaVL-R1-7B (Qwen2.5-VL-3B-Instruct)	68.4	84.0
SophiaVL-R1-7B (our trained 3B RM)	71.3	85.4
SophiaVL-R1-7B (our trained 32B RM)	72.2	86.1

From these results, we observe that reward models with higher performance on VLRewardBench lead to stronger reasoning improvements in our method. Nevertheless, considering the substantial computational overhead of larger reward models, we use the 3B thinking reward model in our main training experiments to balance effectiveness and efficiency.

G Reasoning Model Scaling Up Study

We conduct expriment on Qwen2.5-VL-32B-Instruct as both reward model and reasoning model. Results are shown in Table 10.

Table 10: Performance of 32B reasoning model.

Model	MathVista (Math)	MMBench (General)
Qwen2.5-VL-32B-Instruct	72.5	86.5
Qwen2.5-VL-32B-Instruct + GRPO	73.1	86.8
SophiaVL-R1-32B	73.9	87.5

From Table 10, we observe that SophiaVL-R1-32B surpasses Qwen2.5-VL-32B-Instruct trained with standard GRPO, demonstrating that even with larger models, the proposed method is still effective.

H Analysis of Fine-Grained vs. Coarse-Grained Reward Modeling

A central design choice in our framework is the granularity of the model-based rewards used to guide policy optimization. In principle, finer-grained numerical scores may offer richer supervision than coarse correctness indicators. However, in practice, reward granularity interacts strongly with both model stability and optimization dynamics. This section provides additional analysis and empirical evidence motivating our choice of a discrete 10-choice reward scheme.

H.1 Limitations of Fully Continuous Rewards

We initially explored prompting the reward model to assign continuous real-valued scores (e.g., two-decimal numbers in [0,1]). Although this offers theoretically high resolution, we found that the reward signal became unstable and inconsistent across semantically similar responses. Small numerical fluctuations—for example, scores such as [0.33,0.32,0.29,0.27,0.33,0.32,0.27,0.27]—often reflected noise rather than meaningful quality differences. During optimization, these small variations are amplified, causing the policy to incorrectly rank responses and introducing optimization drift. This instability suggests that continuous scoring introduces more noise than useful signal when applied at this level of granularity.

H.2 Limitations of Coarse 0/1 Rewards

At the opposite extreme, coarse binary rewards provide stable but very sparse supervision. To quantify this effect, we construct a variant of our system where each reward was thresholded at 0.5, yielding a 0/1 signal. The performance results are shown in Table 11.

Table 11: Performance with 0/1 reward model and thinking reward model.

Model	MathVista (Math)	MMBench (General)
Qwen2.5-VL-7B-Instruct+Trust-GRPO+0/1-Reward Model	67.7	83.5
SophiaVL-R1-7B	71.3	85.4

The 0/1-Reward Model variant exhibits substantial degradation on both benchmarks, highlighting that binary correctness signals fail to differentiate intermediate-quality reasoning steps and thus hinder effective policy updates.

H.3 Distinguishing High-Quality and Low-Quality Reasoning

To further examine reward fidelity, we manually evaluated 60 response samples: 30 with correct final answers but flawed reasoning, and 30 with both correct answers and valid reasoning. The average reward outputs from our thinking reward model are summarized in Table 12.

Table 12: Average thinking reward for low-quality vs. high-quality reasoning cases with correct final answer.

	correct final answers but incorrect reasoning	**correct final answers and correct reasoning**
Thinking reward	0.34	0.78

We can see that the average thinking reward is 0.34 for the first group and 0.78 for the second, indicating that our thinking reward model can effectively distinguish lower-quality reasoning from higher-quality reasoning.

I Analysis of Reasoning Quality of SophiaVL-R1-7B

To provide a more fine-grained analysis of the effectiveness of our thinking reward model, we construct a variant model SophiaVL-R1-7B-wo-TRM, which is trained with untrained Qwen2.5-VL-3B-Instruct as reward model, instead of our thinking reward model. We randomly sampled 100 questions and examined the corresponding outputs produced by this variant and our SophiaVL-R1-7B. We then asked GPT-4o to identify intermediate reasoning errors (ask GPT-4o to judge whether there is error in the reasoning). The results of both models are reported in Table 13.

Table 13: Proportion of samples containing intermediate reasoning errors, as judged by GPT-4o.

	Proportion of intermediate reasoning errors(↓)
SophiaVL-R1-7B-wo-TRM	0.59
SophiaVL-R1-7B	0.42

These results highlight that our thinking reward model does correct intermediate wrong thinking process and thus improves the reasoning quality.

J Training Details

Table 14: Training hyperparameters.

Hyperparameter	**Value**
Group size	8
Batch size	8
KL divergence coefficient	0.04
Learning rate	5×10^{-7}
α	0.3
Total training steps	1500

K Evaluation Details

Most of our evaluations are conducted using VLMEvalKit (Duan et al., 2024), following the recommended Python package versions. For baseline models, performance metrics are obtained from the OpenVLM leaderboard. We adopt the default prompts for all evaluated models and modify the answer extraction function based on each model's output format. For instance, for R1-style models, we extract the content enclosed within the <answer> and </answer> tags.

For MathVista, we evaluate on the `testmini` split. For MathVerse, we report average performance over the following subsets: vision-only, vision-dominant, vision-intensive, text-dominant, and text-lite. For MMMU, we evaluate on the `mmmu_dev_val` set. For ChartQA, evaluation is conducted on the test set. For MMBench, we use the `MMBench_Dev_EN` set for evaluation.

Q: What are the key elements in this picture?

The image captures a moment on a road in Iceland, where a signpost stands tall against the backdrop of a cloudy sky and a barren landscape. The signpost, mounted on a metal pole, is painted in a striking blue color with yellow text that reads "Gullfoss". The arrow on the sign points to the right, indicating the direction towards Gullfoss, a renowned waterfall in Iceland. The sign serves as a guide for travelers, leading them towards the natural beauty of Gullfoss.

RESPONSE[6] +6

<think>Let me think about the key elements in this picture. The image shows a sign with a bright blue background and yellow text that reads "Gullfoss". The text is prominent and legible. The sign has a yellow arrow pointing to the right, indicating a direction. The background is a flat, barren landscape, likely a desert or plain, under a cloudy sky. ...... There are no other objects or signs in the frame.</think>
<answer>The key elements in this picture are the road sign with the word "Gullfoss" and a yellow arrow pointing to the right, the barren landscape background, and the cloudy sky. The sign is the most prominent feature, serving as a clear directional indicator.
</answer>

R_6^t 0.8

......

RESPONSE[8] +8

<think>Let me think about the key elements in this picture. The image prominently features a road sign. The sign is blue with yellow text and an arrow pointing to the right. The text ...... "Gullfoss". This suggests that the sign is directing towards the Gullfoss waterfall. The background ...... this sign is located in a rural area or a place of natural beauty. ...... suggesting they are part of a system designed to guide travelers.</think>
<answer>The key elements in this picture are the road sign indicating "Gullfoss", the blue color of the sign, the yellow arrow pointing to the right, the text "Gullfoss", the cloudy sky, the open landscape in the background, and the pole holding the sign. The sign's purpose is to guide travelers towards the Gullfoss waterfall in Iceland.</answer>

R_8^t 0.0

$\mu_w > \mu_c$

$\gamma = 0.45$

Figure 10: Case of a free-form problem.

L CASE STUDY OF γ

Q: James goes out to eat. He orders a steak and egg meal for $16. He is with his friend, who orders chicken fried steak for x. His friend pays for half the bill and James pays the tip along with his half of the bill. They tip 20%. How much did James pay?

RESPONSE[2]

<think>Let me think through this step by step. First, let's calculate the total bill before the tip ...... So the total bill is 16 + x ...... Let's solve this equation for x.
First, factor out (16 + x) :

$$\frac{16+x}{2} + 0.2(16+x) = 21$$

$$\frac{16+x}{2} + 0.1(16+x) = 21$$

...... So, the value of the unknown variable x is 19.</think>
<answer>19</answer>

✗

R_2^t 0.7

......

RESPONSE[5]

<think>Let me think about this step by step.
1. First, we need to calculate the total cost of the meal......
4. James pays his half of the bill plus the tip. So James' payment is $\frac{1}{2}(16+x) + 0.2(16+x)$.
5. We know that James' payment is 21. So we can set up the equation:

$$\frac{1}{2}(16+x) + 0.2(16+x) = 21$$

......
8. Solve for x: x = 30 - 16 =14</think>
<answer>14</answer>

✅

R_5^t 0.5

$\mu_w > \mu_c$

$\gamma = 0.74$

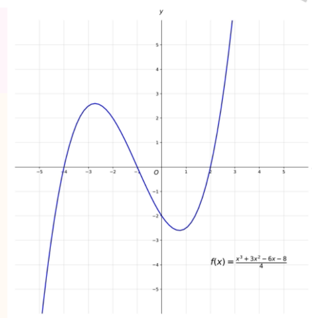
Figure 9: Case of a text-only mathematical problem.

We demonstrate a text-only mathematical problem case in Figure 9. All responses in this image corresponded to the same question displayed on the top. The ground truth answer is 14. Responses yielding incorrect answers (*e.g.*, **RESPONSE[6]**) are highlighted in red (grouped as G_{wrong}), while while those producing correct answers (*e.g.*, **RESPONSE[8]**) are highlighted in green $G_{correct}$. Notably, RESPONSE[6] receives a thinking reward of 0.7 despite a clear arithmetic simplification error, exceeding the reward assigned to RESPONSE[8]. By computing the average thinking reward of $G_{correct}$ and G_{wrong}, we obtain a trustworthiness weight of $\gamma = 0.74$. Since $\gamma < 1$, this indicates the presence of potential unreliable thinking rewards within this response group. This case demonstrates how our Trust-GRPO algorithm can adaptively identify such unreliability during training and appropriately downscale its influence by adjusting γ, thereby providing more stable and reliable reward signals for effective GRPO training.

We present a free-form reasoning example in Figure 10, with the ground-truth answer shown at the top. Notably, RESPONSE[6] identifies six key points, while RESPONSE[8] correctly identifies eight—demonstrating a more complete reasoning process. However, RESPONSE[8] receives a thinking reward of only 0.0, significantly lower than the 0.8 assigned to RESPONSE[6]. Together with the first case study and the analysis provided in the main paper, this example illustrates how the trustworthiness weight γ functions as an effective dynamic parameter for identifying unreliable thinking rewards. By adaptively down-weighting suspicious reward signals, γ helps mitigate the risk of reward hacking and contributes to more stable reinforcement learning.

M Reasoning Examples of SophiaVL-R1-7B

Question: What is the biggest zero of this function?



<think>To find the biggest zero of the function $f(x) = \frac{x^3+3x^2-6x-8}{4}$, we need to determine the values of x for which f(x) = 0. This means solving the equation: $\frac{x^3+3x^2-6x-8}{4} = 0$. Multiplying both sides by 4, we get: $x^3 + 3x^2 - 6x - 8 = 0$.
We need to find the roots of this cubic equation. By inspection or using the Rational Root Theorem, we can test possible rational roots. The possible rational roots are the factors of the constant term (-8) divided by the factors of the leading coefficient (1), which are $\pm 1, \pm 2, \pm 4, \pm 8$.
Let's test x = 2 : $2^3 + 3(2^2) - 6(2) - 8 = 8 + 12 - 12 - 8 = 0.$
So, x = 2 is a root. We can now factor x - 2 out of the cubic polynomial using synthetic division or polynomial division.
Performing synthetic division with x = 2 :

2	1	3	-6	-8
		2	10	8
	1	5	4	0

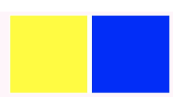
The quotient is $x^2 + 5x + 4$. We can factor this quadratic:

$$x^2 + 5x + 4 = (x + 1)(x + 4)$$

So, the roots of the quadratic are x = -1 and x = -4.
The roots of the original cubic equation are x = 2, x = -1, x = -4 .
Therefore, the biggest zero of the function is 2.</think>
<answer>2</answer>

Figure 11: An example of SophiaVL-R1-7B's reasoning output on MathVista.

Question: Will red be obtained by mixing the above two colors? Please answer yes or no.



<think>Let's break this down. The image shows two squares, one yellow and one blue. Yellow and blue are primary colors in the subtractive color model, which is used in printing and painting.
When yellow and blue are mixed, they typically produce green, not red. Red is a primary color in the additive color model, which is used in digital displays and light. In the subtractive color model, mixing yellow and blue does not produce red; it produces green.</think>
<answer>No</answer>

Figure 12: An example of SophiaVL-R1-7B's reasoning output on MME.

Which image was generated by AI on <image 1>?
A. The left one B. The right one
C. Both images D. Neither image

<think>Let me think about this. The images appear to be paintings, which are not typically generated by AI. However, the question seems to be asking which one was created by AI, which is unusual for a question about art. Given the options, it's likely that the question is asking which one is not a traditional painting, which would be the AI-generated image. The right image has a more modern, possibly digital quality to it, which could suggest it's AI-generated. The left image looks more like a traditional painting with a classic style.
Hmm, I see. The right image has a more contemporary look, which might suggest it's AI-generated. The left image looks more like a traditional painting.
Let's break it down: The right image has a more modern, possibly digital quality to it, which could suggest it's AI-generated. The left image looks more like a traditional painting.
Therefore, the right image is the one that was likely generated by AI.</think>
<answer>B. The right one.</answer>

Figure 13: An example of SophiaVL-R1-7B's reasoning output on MMMU.

What feeling is represented in this image?
A. engaged B. disordered
C. angry D. supportive

<think>Let me think about this image.
The person in the picture appears to be in a state of confusion or distress, as indicated by the chaotic lines around their head, which resemble a brain with tangled thoughts. The leaves floating around might symbolize a sense of being overwhelmed or scattered. The person's hand on their head suggests they are trying to make sense of something.
Now, let's look at the options:
A. engaged - this doesn't seem to fit as the person doesn't appear to be focused on something.
B. B. disordered - this aligns well with the chaotic lines around the head.
C. C. angry - there's no clear indication of anger, just confusion.
D. D. supportive - the image doesn't show any signs of support or encouragement.
Therefore, the most fitting option is B. disordered.</think>
<answer>B. disordered</answer>

Figure 14: An example of SophiaVL-R1-7B's reasoning output on MMStar.

N USE OF LARGE LANGUAGE MODELS (LLMS)

During the preparation of this manuscript, we use a large language model as a writing support tool. Its role is limited to refining the presentation of text, such as improving grammar, clarity, and style. The model was not involved in research ideation, methodological design, implementation, or analysis. All scientific contributions and claims are entirely the work of the author(s).