Learning Consistency in Reward Modeling for Multi-Modal Reasoning

Chen Li[1,2], Bolin Ni[2], Boxin Zhang[2,3], Jinnian Zhang[2], Ke Ye[1], Houwen Peng[2], Han Hu[2], and Nanning Zheng[1⋆]

[1] State Key Laboratory of Human-Machine Hybrid Augmented Intelligence, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, China
edward82@stu.xjtu.edu.cn, nnzheng@mail.xjtu.edu.cn
[2] Tencent, Beijing, China
[3] School of Physics, Peking University, Beijing, China

Abstract. Reliable reward system is essential for reinforcement learning in multi-modal reasoning, yet existing methods face key limitations: preference-based models often misjudge correctness on complex tasks, while rule-based approaches struggle with semantically equivalent answers and therefore restrict the scale of usable RL data. To address these challenges, we propose Consistency Reward Models (CRM), which assess how well a candidate response aligns semantically and mathematically with ground truth. Our model produces a continuous reward derived from the logits associated with "Consistent" vs. "Inconsistent", enabling more informative and stable optimization signals than binary rewards during RL. By design, CRM compares answers in text space, which our ablation shows is both sufficient and more accurate for consistency judgment. We also find that incorporating Chain-of-Thought (CoT) reasoning improves the reliability of these consistency assessments. As a judging model, it achieves an 77.8% win rate over VLMEvalKit, demonstrating stronger judging accuracy. When used for RL, the approach enables a 7B policy model to attain state-of-the-art performance on MathVista (75.2%) and MathVision (30.6%), outperforming rule-based systems, while substantially increasing the scale of RL-usable data. The method further generalizes to OCR, MMMU and open tasks and supports stable RL training up to 72B model. These results show consistency reward models offer a more scalable and robust foundation for multi-modal reasoning RL.

Keywords: Reward Modeling · Multi-Modal Reasoning · Reinforcement Learning

⋆ Corresponding author

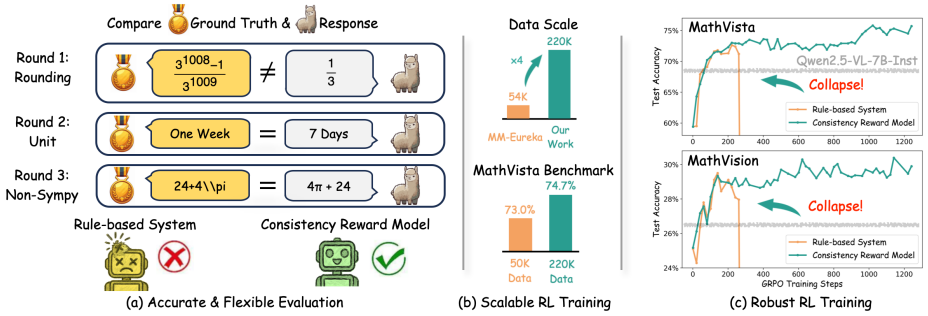

Fig. 1: We propose *Consistency Reward Models*. The model provides accurate reward signals and eliminates the need to consider the verifiability of formatted answers in RL data, significantly expanding the RL training data. With accurate rewards and a large amount of data, robust RL training is achieved, showing significant advantages over rule-based methods.

1 Introduction

Large-scale reasoning models, such as DeepSeek-R1 [20], OpenAI-o3 [45], and Gemini2.5-Pro [18], have recently demonstrated remarkable progress toward general reasoning ability. A central factor behind these advances is reinforcement learning (RL), which strengthens models' reasoning behaviors by optimizing against reward signals. However, the effectiveness of RL critically depends on the design of the reward system, and simple or misaligned rewards often lead to reward hacking [8,66], where models exploit flaws in the reward function without genuinely improving correctness or reasoning quality.

Existing reward modeling approaches face limitations in multimodal reasoning. Preference-based models [11], commonly used in RLHF [4, 46, 56], struggle in complex reasoning settings [55] because the task reduces to predicting absolute correctness of long and intricate responses—something small reward models are not equipped to do. Meanwhile, rule-based reward systems provide symbolic precision through tools such as SymPy [40] or sandbox environments [49, 61]. These systems have been demonstrated higher accuracy in various tasks, including logic [68], mathematics [22, 73], and coding [32]. But they fail to capture semantic and mathematical equivalence (e.g., "one week" vs. "7 days") and enforce strict formatting. This rigidity causes many valid responses to be incorrectly penalized and prevents RL from scaling to more diverse data. Although more optimized rules have been developed in [21, 37], they still fail to accommodate all practical scenarios. These challenges motivate the need for a reward model that is both semantically aware and robust enough for large-scale RL.

To address these issues, we introduce Consistency Reward Models (CRM), a simple yet effective framework that evaluates whether a candidate response is semantically and mathematically consistent with a ground-truth answer. The key intuition is that a reward model does not need to solve the problem itself; it

only needs to judge the equivalence between the model's response and the reference, which greatly simplifies the modeling task, especially for long or multi-step reasoning. We formulate this judging process as a generative procedure in which the model produces a brief Chain-of-Thought (CoT) to compare the two answers and then emits a special token indicating "Consistent" or "Inconsistent." From the logits of these judgment tokens, we derive both hard rewards and continuous soft rewards. Unlike binary decisions, the continuous reward captures the degree of semantic consistency—for example, responses that mix correct and incorrect elements naturally receive intermediate scores between consistent and inconsistent outputs. This finer-grained signal allows RL to distinguish different error modes and promotes smoother optimization. We also find that the lightweight CoT comparison is particularly helpful for soft rewards, where capturing subtle distinctions is important for producing reliable and stable training signals, while still not requiring the reward model to solve the underlying reasoning task.

To train such reward models, we construct a large-scale consistency dataset containing paired questions, ground-truth answers, and diverse model-generated responses from 13 mainstream LLMs and VLMs. For each question-response pair, we collect high-quality consistency judgments using a combination of GPT-4o [24] and Doubao-1.5-vision [52] with CoT, retaining only samples where all judges agree. The resulting dataset contains 860K verified annotations and forms an important contribution to the community, enabling reliable training of consistency reward models and other judge architectures.

Through comprehensive experiments, we show that consistency reward models substantially improve both static judging accuracy and downstream RL performance. A 7B reward model achieves an 77.8% win rate over VLMEvalKit [15], while using the reward model in RL enables a 7B policy model to achieve state-of-the-art results across MathVista [35], MathVision [62], and MathVerse [76]. The approach scales effectively to models with up to 72B parameters and generalizes to OCR and MMMU [74], demonstrating its robustness and versatility. Overall, consistency reward modeling provides a scalable and semantically grounded reward foundation for multimodal reasoning RL. While the reward model itself judges answer consistency in text space—a design validated by our ablation showing text-only input outperforms image-conditioned variants—it directly enables VLM policy improvements across diverse vision tasks.

The contributions can be summarized as follows:

- **We propose consistency reward models**, a semantically driven reward modeling framework for multimodal reasoning that overcomes the limitations of preference-based and rule-based approaches. By comparing the candidate response with the ground truth through CoT-guided analysis, and deriving both hard and continuous rewards from judgment-token logits, our method provides accurate, flexible, and robust reward signals.
- **We construct a high-quality consistency judgment dataset** comprising 860K samples, each containing a question, ground-truth answer, multiple model-generated responses, and verified consistency annotations collected us-

ing GPT-4o and Doubao-1.5 with CoT. This dataset offers a valuable resource for training reward models and other LLM-as-judge systems.

– **We demonstrate strong empirical performance across evaluation and RL settings**. Our 7B consistency reward model achieves a 77.8% win rate over VLMEvalKit, exceeding existing baselines. When applied to RL, consistency rewards enable a 7B policy model to reach state-of-the-art performance on MathVista and MathVision, and support stable RL training up to 72B models. The approach also generalizes well to OCR and MMMU tasks.

2 Consistency Reward Models

2.1 Overview of Reward Modeling

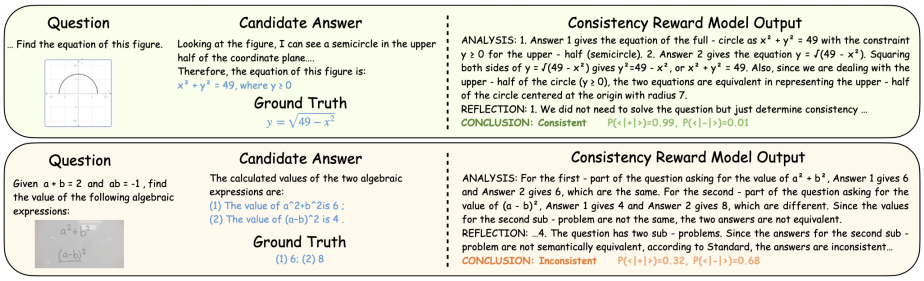

Fig. 2: Illustration of our consistency reward model's output. Through CoT reasoning, our model can achieve detailed comparisons of mathematical equivalence. It ultimately provide conclusions of consistency or inconsistency and the continuous reward score.

Recent advances in large-scale reinforcement learning herald a new era for MLLMs, enabling them to solve complex reasoning tasks. A key factor behind this success is accurate reward modeling, which involves assigning a reward score to the candidate response to a question, where this score can either be a continuous scalar or a predefined tier. Formally, reward modeling can be expressed as:

$$R(q, r) = f(q, s, \theta), \tag{1}$$

where q represents the given question, s the candidate response, and θ the parameters of the reward model.

A widely adopted approach to reward modeling is the preference reward model, which utilizes the Bradley-Terry framework to model the preference between two responses, s_1 and s_2, for the same question q. The probability of preferring s_1 over s_2 is defined as:

$$P(s_1 \succ s_2 \mid q) = \sigma\big(R(q, s_1) - R(q, s_2)\big). \tag{2}$$

However, this model shows limitations, particularly in tasks that require strong reasoning capabilities. The main reasons are twofold: 1) the discriminative scalar

head design hinders the model's ability to engage in CoT reasoning, and 2) the model struggles to accurately assess whether a candidate response is correct, particularly in complex tasks.

To address these limitations, recent research has turned to the rule-based reward model. This model evaluates the correctness of a response by comparing it to the ground-truth answer using a series of symbolic computation tools, thus offering a high degree of precision in correctness assessment. Nevertheless, it also comes with obvious challenge: fixed rules can result in errors, particularly when different phrasings are used to express semantically equivalent answers.

2.2 Learning Consistency for Reward Modeling

Consistency Reward Model. Different from preference learning and fixed rules, we propose a simple yet effective approach: learning the consistency between the candidate answer and the ground truth answer for reward modeling. Specifically, for a given question, we compare the semantic equivalence of the candidate answer and the ground-truth answer to assess the correctness of the candidate answer. Formally, this can be expressed as:

$$R(q, s, g^*) = f(q, s, g^*, \theta), \tag{3}$$

where g^* represents the ground-truth answer.

Compared to the preference reward model, the consistency reward model relaxes the task of "directly judging the quality of the candidate answer" to simply "comparing whether the candidate answer is semantically equivalent to the ground-truth answer", significantly simplifying the task. In some cases, the model can even determine correctness without fully understanding the question, which greatly enriches the application scenarios of the reward model, especially in the hard reasoning scenario.

In contrast to the rule-based reward function, the consistency reward model leverages the semantic understanding capabilities of pre-trained language models, allowing for more flexible handling of diverse tasks without requiring strict formatting. This approach avoids the false negatives that can arise from purely rule-based reward functions.

We formulate the consistency reward model as a generative model and introduce two special judgment tokens, <|+|> and <|-|>, which represent "Consistent" and "Inconsistent", respectively. After producing a Chain-of-Thought (CoT), the model emits one of these tokens, and the probabilities associated with them are used to determine the reward. Let $\mathbf{P}(\cdot)$ denote the model-predicted probability (drived from the token logits) for each judgment token. The reward can be obtained in two forms:

$$r_{\text{hard}} = \begin{cases} 1, & P(\text{<|+|>}) > P(\text{<|-|>}), \\ -1, & \text{otherwise}, \end{cases} \tag{4}$$

$$r_{\text{soft}} = P(\text{<|+|>}) - P(\text{<|-|>}). \tag{5}$$

While soft rewards provide a richer training signal, using them directly in GRPO can introduce instability. Because GRPO normalizes rewards within each group to compute the advantage A, the sign of the advantage may contradict the underlying judgment of the reward model. To prevent updates that oppose the reward model's intent, we apply a reject-sampling rule: samples whose advantage has an opposite sign to their hard reward are excluded from optimization,

$$\mathcal{D}_{\text{train}} = \{(q, s) \in \mathcal{D} \mid \text{sign}(A(q, s)) = \text{sign}(r_{\text{hard}}(q, s))\}\,. \tag{6}$$

This filtering ensures that policy updates remain aligned with the model's consistency judgment, leading to more stable RL training under soft rewards. The overall judgment process is illustrated in Figure 2.

CoT Reasoning. Through chain-of-thought reasoning, it gradually compares and analyzes the candidate answer and the ground-truth answer for consistency, ultimately producing a conclusion. The model completes the comparison in three steps: 1) *Analysis*: Analyze the semantic consistency between the candidate answer and the ground-truth answer in the context of the question; 2) *Reflection*: Step by step, verify and reflect on whether the judgment is correct and if all principles have been adhered to; 3) *Conclusion*: Output <|+|> or <|-|>, which represent "Consistent" and "Inconsistent", respectively. We find that the use of CoT can improve the accuracy of the hard reward and significantly enhance the semantic information captured by the soft reward, thereby reducing the risk of being hacked. The design principles of our CoT are as follows:

1) *Judging Over Solving*: It should judge the correctness of the response, rather than solving the problem itself.
2) *Mathematical Equivalence*: Utilize mathematical theorems and expression simplification techniques to compare.
3) *Multiple Sub-questions*: Judge each sub-question.
4) *Contextual Focus*: Consider the question's context.
5) *Tail Focus*: The overlong responses will be truncated.
6) *Image Absence*: Enable the model to handle judgments in multi-modal questions, using only text input.
7) *Multilingual*: The input may be in multiple languages, but the output must be in English.

Training Data Pipeline. The training dataset is constructed in three steps as shown in Figure 3. To train a reliable consistency reward model, we construct a large-scale dataset tailored specifically for semantic and mathematical consistency judgment. Our design philosophy is guided by three principles:

- the questions should cover the diversity and difficulty of reasoning tasks;
- the responses should cover a wide spectrum of correctness, formats, and reasoning styles to train the model to distinguish fine-grained consistency levels;
- the annotations must be reliable enough to serve as supervision for both hard and continuous rewards.

1) **Question Collection**: We collect questions from both text-only and multi-modal data. The text data includes subsets of the Numina-Math dataset [28],

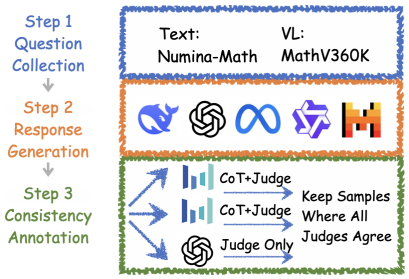

Fig. 3: Illustration of our consistency reward model training data pipeline.

excluding 'synthetic math' and 'metamath'. Additionally, we gather 340k multi-modal data from the MathV360K dataset [53].
2) **Response Generation**: We select 13 mainstream models for response generation, including Qwen families, Mistral-Large [41], GPT-4o [24], DeepSeek-V3 [31], DeepSeek-R1-Distill series [20] and QvQ-70B-preview [59].
3) **Consistency Annotation**: GPT-4o is first utilized to perform direct consistency judgments. Subsequently, we employ the Doubao-1.5-vision-pro-32k-25011 [52] model with CoT reasoning to generate two independent consistency judgments. Only those samples where all three judges agree are kept. This process results in a high-quality dataset for consistency annotations.

The dataset is balanced by a 1:1 ratio between consistent and inconsistent labels, yielding a final training corpus of 860K carefully curated data. We conduct further analysis of training dataset in Supplementary Material.

3 Experiments

3.1 Evaluation Protocols

In this work, we comprehensively evaluate the effectiveness of our method in two scenarios: *1) static judging performance*; and *2) behavior in online RL training*.

Static judging performance evaluates the reward model's capability as a judge itself. This is the most common evaluation approach in existing works, which assess the ability of distinguishing a pair of candidate responses. In this evaluation, we use the **hard reward** version of consistency reward model by default.

A more challenging scenario involves deploying the reward model in online dynamic RL training, where it must handle more diverse real-time responses. In such cases, the impact of reward hacking becomes amplified, placing higher demands on the reward model's capabilities. By employing both evaluation settings, we can obtain a comprehensive assessment of the reward model's capabilities and its performance in downstream RL training.

3.2 Implementation Details

Hyperparameters of Consistency Reward Model. We train the 7B and 1.5B consistency reward model based on Qwen2.5-Instruct [70]. During training,

we set the maximum sequence length to 4096 tokens, and use a cosine schedule with a maximum learning rate of 2e-5, training for a total of 3 epochs. During generation, we set the temperature to 1.0 and top-k to 20, ensuring that the model's generation maintains diversity while also ensuring generation quality. We utilize FSDP [77] and vllm [26] to accelerate.

RL Training Strategy. We use GRPO to train VLMs. In hard reward settings, we discard samples with an advantage of 0, as these samples do not contribute to the loss. This strategy has also been validated as effective in other works [36,39]. In soft reward settings, we discard samples whose advantage has an opposite sign to the hard reward. We serve CRM with vLLM. By overlapping CRM inference with old/ref-logprob and policy forward passes, the overhead is only about 6% over rule-based mothods.

Hyperparameters of RL Training. We use Qwen-2.5-VL series as our base models. We set the rollout batch size to 512, and 10 responses are generated per sample with a temperature of 1.0. The training batch size is set to 1280. We set a constant learning rate of 1e-6.

3.3 Evaluating CRM's Judging Performance

CRM's Judging Accuracy We first constructed a consistency judgment validation set with 1,000 samples, collected in the same pipeline as the training set but with isolated data. On this validation set, we evaluated the capabilities of our CRM, GPT4o-mini, and xVerify [6]. The results are shown in Table 1. Our model achieves a score of 98.0% on the validation set.

We also attempted to use a vision language model (Qwen-2.5-VL-7B) for consistency modeling, but its performance was slightly inferior to that of a language model, as shown in Table 1. This may be because the judgment of consistency relies less on images, and the reasoning capability of the corresponding language model is slightly higher than that of the vision model. Therefore, we choose not to use image input in the modeling of the consistency model.

We note that while CRM itself operates most effectively on text pairs, the contribution is multimodal in three senses: (1) it addresses a problem unique to multimodal reasoning (format-variant answers from vision tasks), (2) it supports image input and we ablate this design choice, and (3) VLM policies trained with CRM gain consistently across vision benchmarks.

Table 1: The judgment performance of various methods on validation set.

Model	**GPT-4o-mini**	**xVerify**	**CRM**	**CRM (w/o CoT)**	**CRM (w/ Image)**
Val Acc.	96.9%	93.2%	**98.0%**	97.2%	96.7%

We also compared our method against other LLM-as-a-judge approaches on public benchmarks VerifyBench-Hard [69], as shown in Table 2. Our model demonstrates a significant advantage despite using comparable base models.

CRM's Judging Win Rate Against Existing Judge. To directly compare judging quality, we introduce win rate, which measures how often CRM gives

Table 2: Overall performance(%) of VerifyBench-Hard.

Model/Method	VerifyBench-Hard				
	Num	Exp	MC	Str	AVG
Qwen3-8B	68.65	78.41	73.02	66.52	70.90
Qwen3-8B (+HLE prompt)	69.05	76.14	75.35	72.17	73.20
xVerify-8B-I [6]	69.05	76.14	93.49	81.74	83.10
Tencent-Qwen2.5-7B-RLVR [57]	75.43	75.00	61.36	63.64	72.20
CompassVerifier-7B [33]	78.97	**85.23**	95.35	76.09	85.90
CRM-7B (Ours)	**79.37**	84.09	**96.74**	**84.35**	**88.40**

the correct judgment in the cases where it disagrees with the baseline. The following sections will detail: 1) the methodology for calculating win rate, 2) the benchmark datasets, 3) the policy model pool, and 4) the baseline judge.

1) **Win Rate Calculation.** We first identify the samples which the CRM and baseline judge give different judgments. We then manually verify these samples and classify into three group. Win. The CRM is correct; Lose. The baseline judge is correct; Tie. The situations where we cannot determine correctness, such as the ground truth has mistakes. We only take 'Win' and 'Lose' samples as account. The win rate is computed as:

$$\text{Win Rate} = \frac{\text{Win}}{\text{Win} + \text{Lose}}, \tag{7}$$

2) **Datasets Used for Judging.** We choose MathVision [62], MathVista [35], and MathVerse [76] (vision only). These benchmarks include a variety of question types such as multiple-choice, judgment-based, and free-form questions.

3) **Policy Model Pool Used for Judging.** We select a total of nine multiple series models, including both short and long CoT models, open source and closed source, ranging from 7B to over 70B parameters. All responses are sourced from the official open-source release of VLMEvalKit.

4) **Baseline Judge.** VLMEvalKit employs a two-step approach to assess the correctness of the responses: first, it extracts the answer, and then compares it to the ground truth. We adopt VLMEvalKit as the baseline judge, which employs GPT-4o-mini for answer extraction followed by strict string matching, with the extracted content also being sourced from the official open-source release.

5) **Win Rate Results.** Across all evaluation tasks, our 7B consistency reward model performs significantly better than VLMEvalKit's pipeline. The average win rate against VLMEvalKit is 77.8% as shown in Table 3.

3.4 Evaluating CRM in Online RL Training

CRM Enables Using 4× RL Data Directly. Different from previous rule-based approaches that rely on a meticulously designed data cleaning pipeline, where large amounts of data are filtered out or require relabeling due to lacking standardized answer formats, our consistency reward model enables direct utilization of vast amounts of raw data.

We directly use the mathematical portion of the open-source Llava-OneVision dataset [27]. We remove all multi-turn dialogues, images with dimensions larger

Table 3: The evaluation win rate conducted by VLMEvalKit and our 7B consistency reward model. This experiment shows that our consistency reward model has a significant advantage than the commonly use VLMEvalKit in judging tasks.

Policy Model	MathVista [35]	MathVision [62]	MathVerse [76]
Short CoT Models			
Claude3.7-Sonnet [1]	50.0%	97.0%	92.1%
GPT-4o-241120 [24]	53.8%	83.3%	90.0%
Qwen-2.5-VL-7B [3]	58.3%	79.5%	93.5%
Llama-3.2-11B-V-Inst [19]	45.4%	64.8%	90.0%
Gemma3-27B [58]	61.9%	82.4%	83.3%
Qwen-2.5-VL-72B [3]	33.3%	91.5%	96.9%
InternVL2.5-78B-MPO [10]	75.0%	86.9%	88.5%
Long CoT Models			
Gemini2-flash [17]	53.3%	95.3%	88.9%
QvQ-72B-Preview [59]	73.5%	92.7%	100.0%

than 768×768 pixels and a short edge smaller than 28. Finally, we eliminate all descriptive and medical questions. We finally select 220K data as our training data. This scale of data can exceed that of prior work by more than 4×.

CRM Outperforms in RL Training. Our evaluation compares the performance of state-of-the-art closed-source and open-source multimodal models across four mathematical reasoning benchmarks: MathVista, MathVerse, MathVision, and WeMath. Table 4 presents our main results. Both hard and soft reward experiments demonstrate notable enhancement. For 7B parameter models, we observe that our model trained solely with open-source data achieves significant improvements. On the MathVista and MathVerse benchmarks, our 7B model achieves results close to those of the 72B model. On the MathVision, and WeMath benchmarks, we also set new state-of-the-art results for 7B models. The policy model trained by soft reward gets better results (+1.0) than hard reward setting, indicating that using dense rewards can further enhance the model's reasoning ability.

We also compared our approach against baselines using rule-based methods, xVerify [6], and CompassVerifier [33]. Given the same base model and training data, our CRM achieves significantly higher performance. We also observe that the use of CRM significantly enhances the stability of the training process and improves scaling capabilities. For the rule-based RL baseline, we use MathRuler [21], which handles LaTeX/SymPy normalization and unit-bearing numbers but still fails on broader semantic equivalences common in multimodal reasoning. As shown in Figure 1, the rule-based system can only provide accurate rewards for a small subset of the full dataset. Although its performance curve is initially similar to that of the consistency model during early training, the lack of robustness in its reward signals often leads to collapse after a period of training. In contrast, RL training based on our consistency reward model demonstrates continuous improvement, ultimately outperforming the rule-based experiments by 2.8% and 2.0% on MathVista and MathVision benchmarks, respectively.

When we apply our consistent reward model to the RL training of 72B model, the results still achieve significant improvements, demonstrating the strongest

Table 4: Performance comparison across multimodal mathematical benchmarks.

Model	Size	MathVista [35]	MathVerse [76]	Mathvision [62]	WeMath [48]	Avg.
Closed-Source Models						
Gemini2.5Pro [18]	-	80.9%	76.9%	69.1%	78.0%	76.2%
GPT-5 [44]	-	81.9%	81.2%	72.0%	71.1%	76.6%
GPT-4o [24]	-	63.8%	50.2%	30.4%	68.8%	53.3%
Open-Source 7B Models						
Qwen-3-VL-8B [2]	8B	77.2%	62.1%	53.9%	64.2%	64.4%
Qwen-2.5-VL-7B [3]	7B	68.2%	47.9%	25.4%	62.1%	50.9%
R1-Onevision-7B [71]	7B	64.1%	47.1%	29.9%	61.8%	50.7%
OpenVLThinker-7B [14]	7B	70.2%	47.9%	25.3%	64.3%	51.9%
MM-Eureka-7B [39]	7B	73.0%	50.3%	26.9%	66.1%	54.1%
InternVL2.5-8B-MPO [64]	8B	68.9%	35.5%	21.5%	53.5%	44.9%
Open-Source 32B+ Models						
Qwen-2.5-VL-32B [3]	32B	74.7%	49.9%	40.1%	69.1%	58.5%
InternVL2.5-38B-MPO [64]	38B	73.8%	46.5%	32.3%	66.2%	54.7%
Qwen-2.5-VL-72B [3]	72B	74.8%	57.6%	38.1%	72.4%	60.7%
Qwen-2.5-VL Experiments						
Rule-RL-7B-Baseline	7B	72.4%	52.3%	28.6%	67.5%	55.2%
xVerify-RL-7B [6]	7B	74.2%	53.5%	29.3%	67.8%	56.2%
CompassVerifier-RL-7B [33]	7B	74.4%	54.8%	29.2%	68.6%	56.8%
CRM-RL-7B (Ours, hard-reward)	7B	74.3%	56.0%	29.8%	69.6%	57.4%
CRM-RL-7B (Ours, soft-reward)	7B	**75.2%**	**57.1%**	**30.6%**	**70.5%**	**58.4%**
CRM-RL-72B (Ours, soft-reward)	72B	**76.5%**	**61.0%**	**41.7%**	**74.3%**	**63.4%**
Qwen-3-VL Experiments						
Rule-RL-8B-Baseline	8B	78.5%	63.2%	55.1%	68.5%	66.3%
CRM-RL-8B (Ours, hard-reward)	8B	78.9%	64.9%	56.4%	70.3%	67.6%
CRM-RL-8B (Ours, soft-reward)	8B	**79.2%**	**65.8%**	**57.2%**	**71.7%**	**68.5%**

performance of a 72B-class model across all benchmarks. This indicates that our method can be scaled to larger RL training scenarios, enabling robust training. Under the updated Qwen-3 setting, soft rewards exhibit superior downstream RL performance compared to hard rewards.

Table 5: Performance on MathVerse (%) v.s. Training Steps.

Reward Method	Step-200	Step-600	Step-900	Step-1000	Step-1200
Rule (MathRuler [21])	52.1	-	-	-	-
xVerify [6]	52.3	53.0	53.3	-	-
CompassVerifier-7B [33]	52.1	53.6	54.1	54.5	-
CRM (Ours)	52.3	53.7	54.8	55.4	55.7

To quantify training stability, we compare reward methods across training steps on MathVerse under the same 220K training data, as shown in Table 5. Rule-based rewards collapse earliest (after 200 steps), followed by xVerify and CompassVerifier, while CRM sustains continuous improvement up to 1200 steps.

3.5 Comparing CRM with Other Variants

Comparison with Outcome Reward Model. We additionally trained a generative outcome reward model for comparison. The inputs are text question, images and text response. We choose Qwen2.5-VL-7B-Instruct as the base model. The outcome reward model analyzes responses with CoT and provides a correct-

Table 6: Comparison of win rate between different reward modeling variants and VLMEvalKit on Claude-3.7 test data. Our CRM demonstrates significantly stronger judging capabilities.

Model	MathVista	MathVision	MathVerse	Avg.
Modeling as Outcome Reward Model				
Gen-ORM-7B	23.2%	13.7%	18.2%	17.8%
Modeling as Consistency Reward Model				
Doubao-1.5	100.0%	95.5%	90.9%	94.2%
Qwen2.5-7B-Inst	23.7%	15.5%	35.2%	26.7%
CRM-7B	50.0%	97.0%	92.1%	89.2%

ness judgment. The data sources and data pipeline are identical to those of the consistency reward model.

Comparison with Directly Prompting LLMs We directly prompt Qwen2.5-7B-Instruct model and our teacher model Doubao-1.5-vision-pro-32k-25011.

We employ these reward models to assess the performance of Claude-3.7 on three visual reasoning task benchmarks, and subsequently calculate the win rate metric by comparing the results with those obtained from VLMEvalKit.

As illustrated in the Table 6, the outcome reward model yields the worst results because it is challenging to accurately evaluate the correctness of the model's responses. When using the consistency modeling strategy, directly prompting the base model also gets suboptimal results but is slightly better than the outcome reward model. This indicates that consistency modeling has strong potential, but it still struggles to assess correctness without fine-tuning. In contrast, Doubao-1.5-vision-pro-32k-25011 demonstrates stronger performance.

3.6 Ablation Study on the Use of CoT

As shown in Table 7, we conducted experiments under both hard and soft reward settings, with and without the use of CoT. The results indicate that CoT provides significant improvements in both scenarios. It is particularly noteworthy that while the soft reward setting underperforms the hard reward setting without CoT, the introduction of CoT enables it to achieve a superior performance. This is attributed to the higher modeling requirements of soft rewards setting, where the differences between logits must be meaningful. With the aid of CoT, the consistency model's judgment is enhanced, allowing the soft reward to effectively incorporate confidence information.

Table 7: Ablation study results of the CoT used on both hard and soft reward settings.

Model	MathVista	Mathvision
Rule-RL-7B-Baseline	72.4%	28.6%
CRM-RL-7B (hard)	74.3%	29.8%
CRM-RL-7B (hard, w/o CoT)	73.4% (-1.0%)	28.9% (-0.9%)
CRM-RL-7B (soft)	75.2%	30.6%
CRM-RL-7B (soft, w/o CoT)	73.1% (-2.1%)	28.8% (-1.8%)

To investigate the underlying reasons for the effectiveness of soft rewards in a CoT setting, we manually analyzed the soft reward distribution across 50 random hard cases that xVerify [6] cannot correctly judge. We found that using CRM with CoT leads to higher accuracy and lower-variance soft rewards as shown in Table 8. For misjudgments (i.e., False Positives and False Negatives), the non-CoT model is prone to extreme rewards, leading to highly incorrect reward assignments for these samples. This might explain why CoT-based models achieve superior performance.

Table 8: The soft reward distribution of CRM under different thinking cost.

CoT	Reward Acc	TP Reward	TN Reward	FP Reward	FN Reward
✗	58.0%	0.83	-0.74	0.75	-0.63
✓	72.0%	0.87	-0.68	0.24	-0.39

3.7 Parameter Scaling of CRM

We further train a 1.5B model for consistency reward modeling, keeping the data and hyperparameters unchanged. We evaluate its win rate against VLMEvalKit and RL training results. As shown in Table 9, compared with the 7B model, the 1.5B model still shows an average drop of 9.0% in win rate. Table 9 also shows 1.5B model underperforms 0.7% in RL performance. These results indicate that the capability of the consistency reward model scales with parameters.

Table 9: Comparison of win rate between 7B and 1.5B consistency reward model.

Model	Evaluation Win Rate			RL Test Acc		
	MathVista	MathVision	Avg.	MathVista	MathVision	Avg.
CRM-7B	50.0%	97.0%	73.5%	74.7%	30.4%	52.6%
CRM-1.5B	40.0%	89.0%	64.5% (-9.0%)	74.2%	29.6%	51.9% (-0.7%)

3.8 Task Zero-shot Generalization of CRM

Generalization on verifiable tasks. We investigate consistency reward model's generalization capabilities on OCR and MMMU tasks. These tasks are broader in scope and are not included in the CRM's training data, which allows us to evaluate the generalization capability of the CRM. We evaluate the performance of the consistency reward model on OCR tasks using the widely adopted OCR-Bench [34]. Following the same evaluation protocol, we tested the CRM's win rate in a static judge role and its downstream performance after RL training.

Table 10: The zero-shot judging win rate on OCRBench and MMMU between our CRM-7B and VLMEvalKit. Using CRM on OCR-related RL training brings gains.

Model	Win Rate OCRBench	Win Rate MMMU	RL Test Acc OCRBench
CRM-7B	66.7%	82.2%	86.9% (+0.7%)

The results are shown in Table 10. Although our consistency reward model is not trained with OCR or MMMU data, it still demonstrates better robustness than the rule-based VLMEvalKit method, achieving a more accurate evaluation. We also applied this consistency model to RL training for OCR tasks. By performing GRPO on Qwen-2.5-VL-7B using 50k data from PixMo-CapQA [12], we achieved a 0.7% improvement on OCRBench.

Generalization on open tasks. To enable the training of CRM on open-ended questions, we formulate captioning and creative writing tasks as objective verification problems. Specifically, we harness GPT-4o to extract checklists of visual details from 10k images in the ShareGPT-4v [9] dataset and writing constraints from 10k samples in the LitBench [16] dataset. During RL training, the reward is derived from the checklist recall rate computed by our CRM, combined with a length penalty (weighted at 0.3) to mitigate reward hacking. Empirical results demonstrate that 100 RL training steps yield significant performance gains, including +3.4% on $\text{CharXiv}_{\text{DQ}}$ [67], +2.6% on InfoVQA [38], and +5.7% on the Arena-Hard [29] creative writing benchmark.

Table 11: Performance gains after RL training on various open-task benchmarks.

Benchmark	Task	Base Model	Performance		Gain (Δ)
			Before RL	After RL	
$\text{CharXiv}_{\text{DQ}}$ [67]	Visual Description	Qwen2.5-VL-7B	73.9%	77.3%	+3.4%
InfoVQA [38]	Visual QA	Qwen2.5-VL-7B	82.6%	85.2%	+2.6%
Arena-Hard (Writing) [29]	Creative Writing	Qwen2.5-72B	10.2%	15.9%	+5.7%

4 Related Work

Rule-based Reward Systems Rule-based reward functions have been used in reinforcement learning across various symbolic and closed domains [5, 42, 43, 50, 51, 54]. Entering the era of large language models, rule-based rewards are also applied in mathematical competition [20,75], code [32] and logical reasoning [68]. In these standardized tasks, rule systems can deliver precise reward signals. More optimized rules have been developed in mathematical reasoning [21, 37]. In the real-world visual reasoning tasks studied in this work, rule systems struggle to handle issues such as unit conversion, estimation, and rounding. The LLM-based consistency reward models we proposed perform better in these scenarios.

LLM-based Reward Models in Reasoning Tasks The preference reward model, initially proposed in Reinforcement Learning from Human Feedback (RLHF) for large language models (LLMs) [56], has become a cornerstone for aligning model outputs with human preferences. These methodologies have been applied across diverse domains, including mathematical reasoning [63], code generation [25,72], and multi-modal tasks [65]. In reasoning tasks, these reward models are categorized into outcome-based and process-based approaches [30, 60]. Outcome reward models (ORMs) rely on response-level correctness, typically evaluating the final answer, while process reward models (PRMs) leverage step-level correctness, assessing intermediate reasoning steps for finer-grained super-

vision [25,63,72]. In our work, we propose a consistency reward model that evaluates the semantic and mathematical equivalence between the model's response and the ground truth response, rather than relying on preference or absolute correctness to achieve high reward accuracy and robust RL training.

Multi-Modal Model Reasoning RL Methods Recent studies have explored reinforcement learning techniques to enhance vision-language models' reasoning capabilities. R1-V [7] and MM-Eureka [39] integrated the GRPO algorithm from into VLM training on object-counting tasks. VisualThinker-R1-Zero [78] demonstrated that applying RL training to a base model yields more performance gains. To improve multimodal reasoning, Vision-R1 [23] introduced a multimodal Chain-of-Thought dataset, converting visual information into a language format to obtain long CoT responses from language reasoning models. This dataset served as cold start data before applying GRPO to further refine the model's reasoning capabilities. Additionally, Curr-ReFT [13] proposed a three-stage RL framework with progressively increasing difficulty-level rewards to optimize training efficiency. Similarly, LMM-R1 [47] introduced a two-stage rule-based RL approach to use text-only data.

5 Conclusion

We introduced Consistency Reward Models (CRM) as a robust and scalable reward modeling framework for multimodal reasoning. By framing reward evaluation as a task of judging semantic and mathematical consistency—rather than solving the underlying problem—CRM provides clear hard decisions and fine-grained continuous signals through a simple generative process with lightweight CoT comparison. Combined with our large-scale consistency dataset, this enables training reward models that are flexible, accurate, and broadly applicable.

Across both static evaluation and online RL, CRM consistently improves the reliability of reward signals and strengthens policy learning. It outperforms existing judging pipelines, supports stable RL on significantly larger and more diverse data, and scales effectively to large model sizes. Moreover, CRM generalizes beyond mathematical reasoning to tasks such as OCR, MMMU and open tasks, demonstrating its versatility as a reward modeling approach.

Overall, our findings show that consistency-based rewards provide a strong foundation for future multimodal RL systems. We believe this direction opens opportunities for building more reliable, semantically aligned, and scalable reasoning models, and we expect CRM to facilitate further progress toward broadly capable multimodal intelligence.

6 Acknowledgments

We thank Yeyao Ma and Jinpeng Dong for their valuable discussions and assistance throughout this work. This work was supported by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China under Grant JYB2025XDXM504.

References

1. Anthropic: Claude 3.7 sonnet and claude code. (2025), https://www.anthropic.com/news/claude-3-7-sonnet/
2. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., Zhu, K.: Qwen3-vl technical report (2025), https://arxiv.org/abs/2511.21631
3. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al.: Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923 (2025)
4. Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al.: Training a helpful and harmless assistant with reinforcement learning from human feedback. arXiv preprint arXiv:2204.05862 (2022)
5. Berner, C., Brockman, G., Chan, B., Cheung, V., Dębiak, P., Dennison, C., Farhi, D., Fischer, Q., Hashme, S., Hesse, C., et al.: Dota 2 with large scale deep reinforcement learning. arXiv preprint arXiv:1912.06680 (2019)
6. Chen, D., Yu, Q., Wang, P., Zhang, W., Tang, B., Xiong, F., Li, X., Yang, M., Li, Z.: xverify: Efficient answer verifier for reasoning model evaluations. arXiv preprint arXiv:2504.10481 (2025)
7. Chen, L., Li, L., Zhao, H., Song, Y., Vinci: R1-v: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V (2025), accessed: 2025-02-02
8. Chen, L., Zhu, C., Soselia, D., Chen, J., Zhou, T., Goldstein, T., Huang, H., Shoeybi, M., Catanzaro, B.: Odin: Disentangled reward mitigates hacking in rlhf. arXiv preprint arXiv:2402.07319 (2024)
9. Chen, L., Li, J., Dong, X., Zhang, P., He, C., Wang, J., Zhao, F., Lin, D.: Sharegpt4v: Improving large multi-modal models with better captions. arXiv preprint arXiv:2311.12793 (2023)
10. Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Cui, E., Zhu, J., Ye, S., Tian, H., Liu, Z., et al.: Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. arXiv preprint arXiv:2412.05271 (2024)
11. Christiano, P.F., Leike, J., Brown, T., Martic, M., Legg, S., Amodei, D.: Deep reinforcement learning from human preferences. Advances in neural information processing systems **30** (2017)
12. Deitke, M., Clark, C., Lee, S., Tripathi, R., Yang, Y., Park, J.S., Salehi, M., Muennighoff, N., Lo, K., Soldaini, L., et al.: Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 91–104 (2025)
13. Deng, H., Zou, D., Ma, R., Luo, H., Cao, Y., Kang, Y.: Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. arXiv preprint arXiv:2503.07065 (2025)
14. Deng, Y., Bansal, H., Yin, F., Peng, N., Wang, W., Chang, K.W.: Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement (2025), https://arxiv.org/abs/2503.17352

15. Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., Dong, X., Zang, Y., Zhang, P., Wang, J., 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)
16. Fein, D., Russo, S., Xiang, V., Jolly, K., Rafailov, R., Haber, N.: Litbench: A benchmark and dataset for reliable evaluation of creative writing (2025), https://arxiv.org/abs/2507.00769
17. Google: gemini-2-flash (2025), https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#ceo-message
18. Google: gemini-2.5-pro-preview (2025), https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro-preview.pdf
19. Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., et al.: The llama 3 herd of models. arXiv preprint arXiv:2407.21783 (2024)
20. Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al.: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948 (2025)
21. hiyouga: Mathruler. https://github.com/hiyouga/MathRuler (2025)
22. Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., Shum, H.Y.: Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. arXiv preprint arXiv:2503.24290 (2025)
23. Huang, W., Jia, B., Zhai, Z., Cao, S., Ye, Z., Zhao, F., Xu, Z., Hu, Y., Lin, S.: Vision-r1: Incentivizing reasoning capability in multimodal large language models. arXiv preprint arXiv:2503.06749 (2025)
24. Hurst, A., Lerer, A., Goucher, A.P., Perelman, A., Ramesh, A., Clark, A., Ostrow, A., Welihinda, A., Hayes, A., Radford, A., et al.: Gpt-4o system card. arXiv preprint arXiv:2410.21276 (2024)
25. Jain, A.K., Gonzalez-Pumariega, G., Chen, W., Rush, A.M., Zhao, W., Choudhury, S.: Multi-turn code generation through single-step rewards. arXiv preprint arXiv:2502.20380 (2025)
26. Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C.H., Gonzalez, J., Zhang, H., Stoica, I.: Efficient memory management for large language model serving with pagedattention. In: Proceedings of the 29th Symposium on Operating Systems Principles. pp. 611–626 (2023)
27. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024)
28. LI, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S.C., Rasul, K., Yu, L., Jiang, A., Shen, Z., Qin, Z., Dong, B., Zhou, L., Fleureau, Y., Lample, G., Polu, S.: Numinamath. [https://huggingface.co/AI-MO/NuminaMath-CoT](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf) (2024)
29. Li, T., Chiang, W.L., Frick, E., Dunlap, L., Zhu, B., Gonzalez, J.E., Stoica, I.: From live data to high-quality benchmarks: The arena-hard pipeline (April 2024), https://lmsys.org/blog/2024-04-19-arena-hard/
30. Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., Cobbe, K.: Let's verify step by step. In: The Twelfth International Conference on Learning Representations (2023)
31. Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., et al.: Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437 (2024)

32. Liu, J., Zhang, L.: Code-r1: Reproducing r1 for code with reliable rewards (2025)
33. Liu, S., Liu, H., Liu, J., Xiao, L., Gao, S., Lyu, C., Gu, Y., Zhang, W., Wong, D.F., Zhang, S., et al.: Compassverifier: A unified and robust verifier for llms evaluation and outcome reward. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 33454–33482 (2025)
34. Liu, Y., Li, Z., Huang, M., Yang, B., Yu, W., Li, C., Yin, X.C., Liu, C.L., Jin, L., Bai, X.: Ocrbench: on the hidden mystery of ocr in large multimodal models. Science China Information Sciences **67**(12) (Dec 2024). https://doi.org/10.1007/s11432-024-4235-6, http://dx.doi.org/10.1007/s11432-024-4235-6
35. Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.W., Galley, M., Gao, J.: Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. arXiv preprint arXiv:2310.02255 (2023)
36. Luo, M., Tan, S., Wong, J., Shi, X., Tang, W.Y., Roongta, M., Cai, C., Luo, J., Li, L.E., Popa, R.A., Stoica, I.: Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl (2025), notion Blog
37. Math-Verify: Math-verify (2025), https://github.com/huggingface/Math-Verify/
38. Mathew, M., Bagal, V., Tito, R.P., Karatzas, D., Valveny, E., Jawahar, C.V.: Infographicvqa (2021), https://arxiv.org/abs/2104.12756
39. Meng, F., Du, L., Liu, Z., Zhou, Z., Lu, Q., Fu, D., Shi, B., Wang, W., He, J., Zhang, K., et al.: Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. arXiv preprint arXiv:2503.07365 (2025)
40. Meurer, A., Smith, C.P., Paprocki, M., Čertík, O., Kirpichev, S.B., Rocklin, M., Kumar, A., Ivanov, S., Moore, J.K., Singh, S., Rathnayake, T., Vig, S., Granger, B.E., Muller, R.P., Bonazzi, F., Gupta, H., Vats, S., Johansson, F., Pedregosa, F., Curry, M.J., Terrel, A.R., Roučka, v., Saboo, A., Fernando, I., Kulal, S., Cimrman, R., Scopatz, A.: Sympy: symbolic computing in python. PeerJ Computer Science **3**, e103 (Jan 2017). https://doi.org/10.7717/peerj-cs.103, https://doi.org/10.7717/peerj-cs.103
41. Mistral: mistral-large-2407 (2024), https://mistral.ai/news/mistral-large-2407
42. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602 (2013)
43. Nazari, M., Oroojlooy, A., Snyder, L., Takác, M.: Reinforcement learning for solving the vehicle routing problem. Advances in neural information processing systems **31** (2018)
44. OpenAI: gpt-5 (2025)
45. OpenAI: Introducing-o3-and-o4-mini (2025), https://openai.com/index/introducing-o3-and-o4-mini/
46. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in neural information processing systems **35**, 27730–27744 (2022)
47. Peng, Y., Zhang, G., Zhang, M., You, Z., Liu, J., Zhu, Q., Yang, K., Xu, X., Geng, X., Yang, X.: Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. arXiv preprint arXiv:2503.07536 (2025)
48. Qiao, R., Tan, Q., Dong, G., MinhuiWu, M., Sun, C., Song, X., Wang, J., Gongque, Z., Lei, S., Zhang, Y., et al.: We-math: Does your large multimodal model achieve human-like mathematical reasoning? In: Proceedings of the 63rd Annual Meeting

of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 20023–20070 (2025)
49. Ren, Z., Shao, Z., Song, J., Xin, H., Wang, H., Zhao, W., Zhang, L., Fu, Z., Zhu, Q., Yang, D., et al.: Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. arXiv preprint arXiv:2504.21801 (2025)
50. Samvelyan, M., Rashid, T., De Witt, C.S., Farquhar, G., Nardelli, N., Rudner, T.G., Hung, C.M., Torr, P.H., Foerster, J., Whiteson, S.: The starcraft multi-agent challenge. arXiv preprint arXiv:1902.04043 (2019)
51. Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T.S., Koenig, S., Choset, H.: Primal: Pathfinding via reinforcement and imitation multi-agent learning. IEEE Robotics and Automation Letters **4**(3), 2378–2385 (2019)
52. Seed: Doubao-1.5-vision-pro-32k-25011 (2025), `https://www.volcengine.com/product/doubao`
53. Shi, W., Hu, Z., Bin, Y., Liu, J., Yang, Y., Ng, S.K., Bing, L., Lee, R.K.W.: Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. arXiv preprint arXiv:2406.17294 (2024)
54. Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al.: Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:1712.01815 (2017)
55. Singhal, P., Goyal, T., Xu, J., Durrett, G.: A long way to go: Investigating length correlations in rlhf. arXiv preprint arXiv:2310.03716 (2023)
56. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., Christiano, P.F.: Learning to summarize with human feedback. Advances in neural information processing systems **33**, 3008–3021 (2020)
57. Su, Y., Yu, D., Song, L., Li, J., Mi, H., Tu, Z., Zhang, M., Yu, D.: Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. arXiv preprint arXiv:2503.23829 (2025)
58. Team, G.: Gemma 3 (2025), `https://goo.gle/Gemma3Report`
59. Team, Q.: Qvq: To see the world with wisdom (December 2024), `https://qwenlm.github.io/blog/qvq-72b-preview/`
60. Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., Higgins, I.: Solving math word problems with process-and outcome-based feedback. arXiv preprint arXiv:2211.14275 (2022)
61. Wang, H., Unsal, M., Lin, X., Baksys, M., Liu, J., Santos, M.D., Sung, F., Vinyes, M., Ying, Z., Zhu, Z., et al.: Kimina-prover preview: Towards large formal reasoning models with reinforcement learning. arXiv preprint arXiv:2504.11354 (2025)
62. Wang, K., Pan, J., Shi, W., Lu, Z., Ren, H., Zhou, A., Zhan, M., Li, H.: Measuring multimodal mathematical reasoning with math-vision dataset. Advances in Neural Information Processing Systems **37**, 95095–95169 (2024)
63. Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., Sui, Z.: Math-shepherd: Verify and reinforce llms step-by-step without human annotations. arXiv preprint arXiv:2312.08935 (2023)
64. Wang, W., Chen, Z., Wang, W., Cao, Y., Liu, Y., Gao, Z., Zhu, J., Zhu, X., Lu, L., Qiao, Y., Dai, J.: Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. arXiv preprint arXiv:2411.10442 (2024)
65. Wang, Y., Li, Z., Zang, Y., Wang, C., Lu, Q., Jin, C., Wang, J.: Unified multimodal chain-of-thought reward model through reinforcement fine-tuning. arXiv preprint arXiv:2505.03318 (2025)

66. Wang, Y., Ivison, H., Dasigi, P., Hessel, J., Khot, T., Chandu, K., Wadden, D., MacMillan, K., Smith, N.A., Beltagy, I., et al.: How far can camels go? exploring the state of instruction tuning on open resources. Advances in Neural Information Processing Systems **36**, 74764–74786 (2023)
67. Wang, Z., Xia, M., He, L., Chen, H., Liu, Y., Zhu, R., Liang, K., Wu, X., Liu, H., Malladi, S., Chevalier, A., Arora, S., Chen, D.: Charxiv: Charting gaps in realistic chart understanding in multimodal llms. arXiv preprint arXiv:2406.18521 (2024)
68. Xie, T., Gao, Z., Ren, Q., Luo, H., Hong, Y., Dai, B., Zhou, J., Qiu, K., Wu, Z., Luo, C.: Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. arXiv preprint arXiv:2502.14768 (2025)
69. Yan, Y., Jiang, J., Ren, Z., Li, Y., Cai, X., Liu, Y., Xu, X., Zhang, M., Shao, J., Shen, Y., Xiao, J., Zhuang, Y.: Verifybench: Benchmarking reference-based reward systems for large language models. In: The Fourteenth International Conference on Learning Representations (2026), https://openreview.net/forum?id=JfsjGmuFxz
70. Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al.: Qwen2. 5 technical report. arXiv preprint arXiv:2412.15115 (2024)
71. Yang, Y., He, X., Pan, H., Jiang, X., Deng, Y., Yang, X., Lu, H., Yin, D., Rao, F., Zhu, M., Zhang, B., Chen, W.: R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. arXiv preprint arXiv:2503.10615 (2025)
72. Ye, Y., Zhang, T., Jiang, W., Huang, H.: Process-supervised reinforcement learning for code generation. arXiv preprint arXiv:2502.01715 (2025)
73. Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Fan, T., Liu, G., Liu, L., Liu, X., et al.: Dapo: An open-source llm reinforcement learning system at scale. arXiv preprint arXiv:2503.14476 (2025)
74. Yue, X., Ni, Y., Zhang, K., Zheng, T., Liu, R., Zhang, G., Stevens, S., Jiang, D., Ren, W., Sun, Y., 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)
75. Zeng, W., Huang, Y., Liu, Q., Liu, W., He, K., Ma, Z., He, J.: Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild (2025), https://arxiv.org/abs/2503.18892
76. Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.W., Qiao, Y., 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 (2024)
77. Zhao, Y., Gu, A., Varma, R., Luo, L., Huang, C.C., Xu, M., Wright, L., Shojanazeri, H., Ott, M., Shleifer, S., et al.: Pytorch fsdp: experiences on scaling fully sharded data parallel. arXiv preprint arXiv:2304.11277 (2023)
78. Zhou, H., Li, X., Wang, R., Cheng, M., Zhou, T., Hsieh, C.J.: R1-zero's "aha moment" in visual reasoning on a 2b non-sft model (2025), https://arxiv.org/abs/2503.05132