Beyond Monolithic Rewards: Hybrid Multi-Aspect Reward Optimization

Radha Gulhane* **Sathish Reddy Indurthi***
Zoom Communications, Inc.
{radha.gulhane, sathishreddy.indurthi}@zoom.us

Abstract

Reinforcement learning optimization policies have traditionally relied on a single reward mechanism, most commonly a model-based reward. Such monolithic rewards often lack confidence calibration across domain-specific tasks and fail to capture diverse aspects of model responses. This approach requires extensive data annotation and reward model training, which is particularly challenging for multimodal models. In this work, we propose and provide a thorough study of **hybrid reward** and **multi-aspect reward modeling**. For accuracy and confidence calibration, we introduce a **hybrid reward modeling** framework that integrates complementary reward paradigms: *model-based rewards*, in which a learned reward model predicts scalar or vector scores, and *rule-based rewards*, in which domain-specific heuristics provide explicit correctness signals with confidence. Beyond accuracy, we further incorporate **multi-aspect rewards** to enforce instruction adherence and introduce a generalized length-penalty reward to stabilize training and improve performance. Our experiments demonstrate that this approach significantly enhances reasoning capabilities: our best-performing 3B model achieves an average improvement of 9.5% across multimodal benchmarks, with a notable 16% gain in mathematical reasoning tasks.

1 Introduction

The advent of multimodal models has enabled AI to reason over and generate content that integrates text, images, and other modalities (OpenAI et al., 2024; Liu et al., 2023). A common approach for aligning these models with human preferences is Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2017; Ouyang et al., 2022), typically implemented with Proximal Policy Optimization (PPO) (Schulman et al., 2017), which fine-tunes the model using a learned reward model (RM).

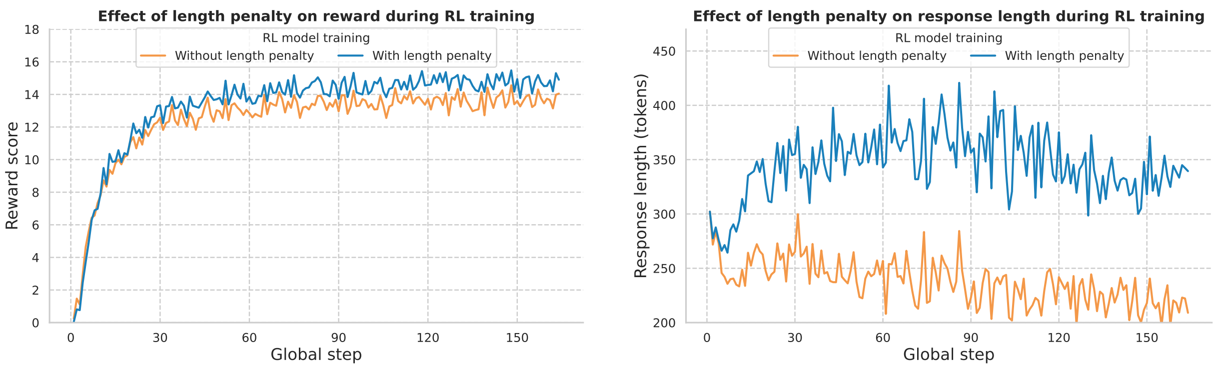
Standard RLHF relies on a single, monolithic RM, which can struggle in multimodal settings. Vision-language tasks introduce ambiguity, making evaluation of text-image consistency difficult, and monolithic RMs are prone to reward hacking (Amodei et al., 2016). Moreover, high-quality multimodal preference datasets and effective reward models (RMs) are scarce in the vision domain, further limiting scalability. Rule-based or verifiable rewards help for deterministic tasks (DeepSeek-AI et al., 2025), but fail to provide nuanced feedback for open-ended, subjective tasks.

To address these challenges, we propose a hybrid, multi-aspect reward framework. Our approach combines a rule-based, verifiable reward to ensure objective correctness with a model-based reward to guide subjective quality. We further incorporate behavioral rewards, including a length penalty, to enforce fine-grained constraints and stabilize training. Finally, we introduce an embedding-based surrogate model as a lightweight proxy for a full RM, reducing the dependency on costly annotation and training. Our primary contributions are:

- We demonstrate that a hybrid and multi-aspect reward portfolio, which combines rule-based, model-based, and behavioral signals, outperforms single-source approaches for multimodal reasoning (Sections 3.2.1 and 3.2.2).
- We provide a reward modeling optimization that integrates seamlessly with RL policy optimization, ensuring generalizability and scalability (Section 3.2.3).
- We demonstrate significant performance improvements over SOTA and RM-based baselines across mathematical, general VQA, and

*Equal contribution.

Findings of the Association for Computational Linguistics: ACL 2026, pages 26519–26533



(a) Accuracy reward. (b) Response length.

Figure 1: Training dynamics: (a) accuracy improves, while (b) the length penalty prevents the collapse into shorter, simpler responses.

OCR tasks (Section 3.2.4).

2 Hybrid and Multi-Aspect Reward Modeling Optimization (HARMO)

2.1 Background: Critic-Free Policy Optimization

While RLHF via PPO (Schulman et al., 2017) is the standard for alignment, it is resource-intensive. Recent methods like RLOO (Ahmadian et al., 2024) and GRPO (DeepSeek-AI et al., 2025) reduce complexity by eliminating the critic and reward model, instead computing advantages via group-relative comparisons. However, advantage normalization methods can introduce bias (Liu et al., 2025). Our framework, HARMO, builds on this by employing a simplified, critic-free objective with a robust, multi-faceted reward signal.

2.2 The HARMO Framework

Hybrid Reward for Calibrated Accuracy. To balance verifiable correctness with open-ended quality, we define a hybrid reward $R^{\text{hybrid}}_{g,i}$. We use rule-based verifiers (R^{rule}) for deterministic tasks (e.g., math) and pretrained reward models (R^{RM}) for generative tasks:

$$R^{\text{hybrid}}_{g,i} = \begin{cases} R^{\text{rule}}_{g,i}, & \text{if response is verifiable,} \\ R^{\text{RM}}_{g,i}, & \text{if response is open-ended.} \end{cases} \tag{1}$$

Multi-Aspect Behavioral Rewards. Optimizing for accuracy alone often leads to *reward hacking* via brevity (see Figure 1). To counteract this and enforce structure, we introduce two auxiliary terms:

- **Length-Penalty Reward (R^{λ}).** We penalize incorrect responses that are shorter than the briefest correct response in the group. Let $\lambda^{\min}_g$ be the minimum length of any correct response in group g. The penalty is:

$$\mathrm{R}^{\lambda}_{g,i} = -\,\mathrm{clip}\Big(\lambda^{\min}_g - \lambda_{g,i},\ 0,\ P_{\max}\Big). \tag{2}$$

- **Format-Adherence Reward (R^{fmt}).** To ensure structural consistency (e.g., <think> tags), we apply positive signals for compliance and penalties for formatting violations.

2.3 Policy Optimization

The total reward is a normalized sum of the components: $R^{\text{HARMO}}_{g,i} = R^{\text{hybrid}}_{g,i} + R^{\lambda}_{g,i} + R^{\text{fmt}}_{g,i}$ (refer Appendix A.4 for normalization details). For optimization, we adopt a GRPO-based approach. However, standard GRPO normalizes using group standard deviation, which can bias training against difficult prompts with high variance (Liu et al., 2025). To mitigate this, we calculate advantage using only the group mean, ensuring a centered but uniformly scaled signal:

$$\hat{A}^{\text{HARMO}}_{g,i} = R^{\text{HARMO}}_{g,i} - \frac{1}{G}\sum_{j=1}^{G} R^{\text{HARMO}}_{g,j}. \tag{3}$$

3 Experiments

3.1 Experimental Setup

Training Data We curate our training corpus from the VLAA-Thinking dataset (Chen et al., 2025), comprising 21,192 samples with distilled reasoning traces. The dataset balances mathematical reasoning (15k samples, ~72%) with general visual question answering (6k samples, ~28%), covering diverse formats from closed-ended equations to open-ended descriptions.

Task Type	Dataset Source	Answer Type	# Samples
Math Reasoning	CLEVR-Math GeoQA170K MathPUMA	Numeric Multiple-Choice Equation	2,000 6,499 6,696
Visual Question Answering (VQA)	DocVQA VizWiz ArxivQA ALLaVA-LAION	Open-Ended Open-Ended Multiple-Choice Open-Ended	1,000 1,000 997 3,000
Total			**21,192**

Table 1: Detailed composition of the VLAA-Thinking training dataset, including task category, source dataset, answer format, and number of samples.

The Table 1 shows the detailed breakdown of the training dataset used in our experiments, along with descriptions of the fields for each sample . It also lists the number of samples per task type, dataset source, and answer format. We also include a description of the fields in each data entry to clarify how responses are categorized as verifiable or open-ended. Each sample in our dataset contains the following fields:

- **message**: a list of system and user messages, which may include both image and text information.
- **reasoning**: step-by-step ground truth reasoning leading to the answer.
- **answer**: the ground truth final answer.
- **answer_format**: the format of the answer (e.g., `open_ended`, `numeric`, `math_equation`).

Based on the `answer_format` field, we differentiate whether a response is verifiable or open-ended during scoring. Verifiable responses (numeric, multiple-choice, math equation) are compared against ground truth answers, while open-ended responses are evaluated qualitatively or via semantic similarity. Example data entries are also provided below to further clarify this procedure.

Sample examples from the dataset are provided in Appendix B.

Models We utilized Qwen2.5-VL-3B-Instruct (Bai et al., 2025) as our primary policy model for ablation studies and scaled to the 7B variant to evaluate the generalizability of the HARMO framework. We benchmarked performance against open-source baselines such as VLAA-Thinker (Chen et al., 2025) and leading proprietary models. For the reward model (R^{RM}), we employed a pre-trained 7B-parameter RM (Wang et al., 2025). To mitigate the high annotation cost of training a VLM-specific RM, we adopted a lightweight 22M-parameter embedding model[1] for comparative experiments.

Implementation Details Our reinforcement learning pipeline builds upon (Peng et al., 2025)[2], integrating the hybrid reward methodology defined in Section 2.2. See Appendix A.3 for more details.

3.2 Results and Analysis

We analyze the contribution of individual HARMO components via ablation studies before demonstrating scalability and comparing against state-of-the-art baselines. Qualitative examples of reasoning improvements are available in Appendix D.

3.2.1 Efficacy of Hybrid Accuracy Rewards

Table 2 evaluates accuracy-focused reward strategies. While a standalone learned *Reward Model (RM)* improves math scores by 7.89% over the baseline, it tends to over-prioritize verbose explanations, indicating poor calibration on verifiable tasks. Our proposed *RM + Rule-based Hybrid* is the most effective strategy, integrating the 7B RM for open-ended nuance with deterministic checks for correctness. This approach yields a 14.82% improvement in math reasoning and a 9.48% gain overall. We attribute this success to the hybrid signal: it leverages the RM's semantic understanding without sacrificing the precision of rule-based verification.

3.2.2 Impact of Multi-Aspect Behavioral Rewards

Table 3 isolates the additive gains from behavioral constraints. The hybrid accuracy reward ($\oplus$H) alone improves math performance by 13.0%. Adding format adherence ($\oplus$H+F) increases this to 14.8%. The full HARMO framework, incorporating a dynamic length penalty ($\oplus$H+F+λ), achieves a peak 16.0% improvement. The length penalty boosts performance on MathVerse (+9.75), and MathVista (+6.70), confirming that penalizing verbosity promotes precise reasoning.

3.2.3 Generalizability and Scalability

We validate HARMO as a scalable, *plug-and-play* framework in Table 4. When integrated with fine-grained, token-level reward models, HARMO con-

[1] `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`

[2] https://github.com/TideDra/lmm-r1

Reward Model	$MathVerse_{mini}$	$MATH\text{-}Vision_{test}$	$MathVista_{mini}$	$MMMU_{val}$	$MMMU\text{-}Pro_{standard}$
Qwen2.5-VL-3B-Instruct (Baseline)					
N/A	34.77	21.68	61.30	31.10	47.78
Reward Model Enhanced					
Skywork7B RM	41.04	22.30	63.70	31.91	47.78
Δ vs. Baseline	(+6.27)	(+0.62)	(+2.40)	(+0.81)	(0.00)
Embedding + Rule-based Hybrid Enhanced					
Hybrid (Rule + Embedding)	40.28	23.85	**67.40**	31.79	46.33
Δ vs. Baseline	(+5.51)	(+2.17)	**(+6.10)**	(+0.69)	(-1.45)
RM + Rule-based Hybrid Enhanced					
Hybrid (Rule + Skywork7B RM)	**41.88**	**25.92**	**67.40**	**32.08**	**48.00**
Δ vs. Baseline	**(+7.11)**	**(+4.24)**	**(+6.10)**	**(+0.98)**	**(+0.22)**

Table 2: Performance under accuracy-focused reward modeling. The hybrid model (pretrained RM + rule-based verification) consistently outperforms baselines.

Reward Model Components	$MathVerse_{mini}$	$MATH\text{-}Vision_{test}$	$MathVista_{mini}$	$MMMU_{val}$	$MMMU\text{-}Pro_{standard}$
Qwen2.5-VL-3B-Instruct Baseline (SFT Only)					
N/A	34.77	21.68	61.30	47.78	31.10
Incremental Reward Augmentation					
⊕ Hybrid (H)	40.38	25.49	67.20	**48.56**	30.98
Δ vs. Baseline	(+5.61)	(+3.81)	(+5.90)	(+0.78)	(-0.12)
⊕ Hybrid + Format (H+F)	41.88	**25.92**	67.40	48.00	**32.08**
Δ vs. Baseline	(+7.11)	(+4.24)	(+6.10)	(+0.22)	(+0.98)
⊕ Hybrid + Format + Length (H+F+λ) **[HARMO]**	**44.52**	24.08	**68.00**	47.11	31.56
Δ vs. Baseline	**(+9.75)**	(+2.40)	**(+6.70)**	(-0.67)	(+0.46)

Table 3: Ablation of reward components on Qwen2.5-VL-3B-Instruct.

Model Configuration	$MathVerse_{mini}$	$MATH\text{-}Vision_{test}$	$MathVista_{mini}$	$MMMU_{val}$	$MMMU\text{-}Pro_{standard}$
Plug-and-Play with Fine-Grained Rewards (3B Model)					
Token-Level Rewards (Baseline)	38.43	23.32	63.50	41.12	**31.79**
Token-Level Rewards + HARMO	**41.22**	**24.84**	**66.40**	**42.32**	31.45
Δ vs. Baseline	(+2.79)	(+1.52)	(+2.90)	(+1.20)	(-0.34)
Scalability to 7B Model Family					
Qwen2.5-VL-7B-Instruct (Baseline)	46.40	25.20	69.70	46.11	36.71
Qwen2.5-VL-7B-Instruct + HARMO	**50.89**	**27.66**	**72.00**	**47.79**	**36.82**
Δ vs. Baseline	(+4.49)	(+2.46)	(+2.30)	(+1.68)	(+0.11)

Table 4: Generalizability and scalability of HARMO across diverse reward schemes and model sizes.

tributes an additional 5.76% improvement. Moreover, scaling to Qwen2.5-VL-7B-Instruct yields a 6.55% gain, confirming the framework's robustness across architectures and reward schemes.

3.2.4 Main Results: Comparison with SOTA Models

As shown in Table 5, HARMO-aligned models significantly outperform baselines and top-tier proprietary systems. Notably, HARMO-VL-3B delivers significant reasoning gains, achieving a 9.48% average improvement across all benchmarks and a remarkable 28.1% boost on MathVerse. These models remain competitive with proprietary systems despite their smaller parameter counts; specifically, scores of HARMO-VL-3B (68.0) and HARMO-VL-7B (72.0) surpass Claude-3.5 Sonnet (67.7) on MathVista. Furthermore, this enhanced reasoning does not come at the cost of core capabilities, as visual robustness on OCR tasks (Table 6) remains comparable to strong baselines.

4 Conclusion

We introduce HARMO, a hybrid and multi-aspect reward optimization framework that advances reinforcement learning beyond monolithic signals. This work highlights the critical role of multi-faceted reward modeling in stabilizing RL training and improving reward accuracy. Our evaluation demonstrates that HARMO significantly enhances reasoning, and mathematical performance over strong baselines. HARMO provides a strong foundation for future research, such as dynamic reward weighting or self-improving systems where agents learn to refine their own reward functions,

Models	MathVerse$_{mini}$	MATH-Vision$_{test}$	MathVista$_{mini}$	MMMU$_{val}$	MMMU-Pro$_{standard}$	Average
Proprietary Vision-Language Models						
GPT-4o	47.8	30.6	63.8	69.1	51.9	52.64
Claude-3.5 Sonnet	41.2	33.5	67.7	68.3	51.5	52.44
Gemini-1.5 Pro	54.8	19.2	63.9	65.8	46.9	50.12
Open-Source Vision-Language Models (3B Scale)						
Qwen2.5-VL-3B-Instruct	34.77	21.68	61.30	**47.78**	31.10	39.73
VLAA-Thinker-Qwen2.5VL-3B	38.78	24.13	64.20	47.56	28.90	40.71
HARMO-VL-3B (Ours)	**44.52**	**24.08**	**68.00**	47.11	**31.56**	**43.05**
Δ vs. Qwen2.5-VL-3B-Instruct	(+9.8)	(+2.4)	(+6.7)	(-0.7)	(+0.5)	(+3.74)
Open-Source Vision-Language Models (7B Scale)						
Qwen2.5-VL-7B-Instruct	46.40	25.20	69.70	**52.56**	36.71	46.11
VLAA-Thinker-Qwen2.5VL-7B	50.56	26.48	70.60	45.11	34.05	45.36
HARMO-VL-7B (Ours)	**50.89**	**27.66**	**72.00**	51.56	**36.82**	**47.79**
Δ vs. Qwen2.5-VL-7B-Instruct	(+4.5)	(+2.5)	(+2.3)	(-1.0)	(+0.1)	(+1.68)

Table 5: Results on general reasoning benchmarks. HARMO significantly improves upon strong open-source models and demonstrates competitive performance against leading proprietary models.

Models	ai2d$_{test}$	chartqa$_{test}$	docvqa$_{val}$
3B Model Family			
Qwen2.5-VL-3B-Instruct	78.43	83.28	**92.56**
HARMO-VL-3B (Ours)	**78.79**	**84.12**	91.88
Δ vs. Qwen2.5-VL-3B-Instruct	(+0.36)	(+0.84)	(-0.68)
7B Model Family			
Qwen2.5-VL-7B-Instruct	82.67	**82.96**	**94.72**
HARMO-VL-7B (Ours)	**82.87**	82.64	94.46
Δ vs. Qwen2.5-VL-3B-Instruct	(+0.20)	(-0.32)	(-0.26)

Table 6: Performance on OCR benchmarks. HARMO matches the baseline, preserving core VQA capabilities despite reasoning improvements.

paving the way for more robust and adaptable AI.

Limitations

We acknowledge several limitations in this work. First, our experiments were restricted to 3B and 7B parameter models, leaving the framework's scalability to larger foundation models unverified. Second, the hybrid reward approach depends on distinguishing verifiable from open-ended tasks, which may prove challenging for subjective domains lacking clear binary correctness rules. Third, to avoid high annotation costs, we relied on pre-trained reward models and embedding surrogates, inherently bounding performance on open-ended tasks to the quality of these proxies. Finally, our normalization hyperparameters were empirically derived from the MMIF-23k dataset, potentially necessitating recalibration when applying HARMO to data with significantly different reward distributions.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *Preprint*, arXiv:2402.14740.

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. 2016. Concrete problems in ai safety. *Preprint*, arXiv:1606.06565.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. 2025. Sft or rl? an early investigation into training r1-like reasoning large vision-language models. *Preprint*, arXiv:2504.11468.

Paul F Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, volume 30.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi.

2016. A diagram is worth a dozen images. *Preprint*, arXiv:1603.07396.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *Preprint*, arXiv:2503.20783.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *Preprint*, arXiv:2310.02255.

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

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. *Preprint*, arXiv:2007.00398.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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

Yingzhe Peng, Gongrui Zhang, Miaosen Zhang, Zhiyuan You, Jie Liu, Qipeng Zhu, Kai Yang, Xingzhong Xu, Xin Geng, and Xu Yang. 2025. Lmm-r1: Empowering 3b lmms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xiaokun Wang, Peiyu Wang, Jiangbo Pei, Wei Shen, Yi Peng, Yunzhuo Hao, Weijie Qiu, Ai Jian, Tianyidan Xie, Xuchen Song, Yang Liu, and Yahui Zhou. 2025. Skywork-vl reward: An effective reward model for multimodal understanding and reasoning. *Preprint*, arXiv:2505.07263.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of CVPR*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, Yu Su, Wenhu Chen, and Graham Neubig. 2025. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *Preprint*, arXiv:2409.02813.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, and Ziwei Liu. 2025. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, page 881–916. Association for Computational Linguistics.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? *Preprint*, arXiv:2403.14624.

A Implementation Details

A.1 Policy Optimization with the HARMO Reward Signal

We integrate the comprehensive reward signal, $R_{g,i}^{\text{HARMO}}$, defined as

$$R_{g,i}^{\text{HARMO}} = R_{g,i}^{\text{hybrid}} + R_{g,i}^{\lambda} + R_{g,i}^{\text{fmt}} \quad (4)$$

into a policy optimization framework based on GRPO, chosen for its stability and proven effectiveness in enhancing reasoning capabilities (Chen et al., 2025). We adopt the GRPO algorithm due to its stability and demonstrated success in enhancing reasoning capabilities in closely related work (Chen et al., 2025). While standard GRPO normalizes rewards using both the mean and standard deviation of a generation group, the standard deviation term can introduce a *difficulty-dependent bias* by disproportionately weighting prompts based on reward variance (Liu et al., 2025). To foster more stable and unbiased learning, we modify the advantage calculation to use only the group mean as a baseline, creating a centered but uniformly scaled signal:

$$\hat{A}_{g,i}^{\text{HARMO}} = R_{g,i}^{\text{HARMO}} - \frac{1}{G}\sum_{j=1}^{G} R_{g,j}^{\text{HARMO}}. \quad (5)$$

The policy π_θ is then updated to maximize the following objective function, which incorporates the PPO-style clipping mechanism and a KL penalty(KL) to ensure training stability:

$$\mathcal{L}^{\text{HARMO}}(\theta) = \mathbb{E}_{q,\{o_i\}\sim\pi_{\text{old}}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\min\Big(r_t(\theta,a_i)\,\hat{A}_{g,i}^{\text{HARMO}}, \text{clip}\big(r_t(\theta,a_i),1-\epsilon,1+\epsilon\big)\,\hat{A}_{g,i}^{\text{HARMO}}\Big) - \beta\,\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})\Bigg] \quad (6)$$

where $r_t(\theta, a_i)$ is the probability ratio $\frac{\pi_\theta(a_i|q)}{\pi_{\text{old}}(a_i|q)}$. The complete training procedure is outlined in Algorithm 1.

Algorithm 1 The HARMO Training Procedure

```
Input: Initial policy π_θinit, HARMO reward function R^HARMO, prompts D, hyperparameters ε, β, reward normalization parameters P_min, P_max.
Initialize: Actor policy π_θ ← π_θinit.
for each iteration i = 1, ..., I do
    Set reference policy: π_ref ← π_θ.
    for each step s = 1, ..., M do
        Sample a batch of questions D_b ⊂ D.
        Set old policy: π_θold ← π_θ.
        for each question q ∈ D_b do
            Sample G responses {o_j}_{j=1}^G ~ π_θold(· | q).
            Compute HARMO rewards {R_{q,j}^HARMO}_{j=1}^G for each response using Equation 4.
            Compute group-relative advantages {Â_{q,j}^HARMO}_{j=1}^G using Equation 5.
        end for
        Update the actor policy π_θ by optimizing the objective in Equation 6.
    end for
end for
Output: Optimized policy model π_θ.
```

A.2 RL training Framework

Our reinforcement learning implementation builds upon the LMM-R1 framework (Peng et al., 2025)[3]. On top of this foundation, we incorporate the methodology described in Section 2, extending the framework with additional functionalities. In particular, we implement hybrid and multi-aspect reward modeling, introduce support for MLLM training, and enable token-level reward assignment for MLLM reinforcement learning.

[3] https://github.com/TideDra/lmm-r1

A.3 Training Hyper-Parameters

The hyperparameters used for HARMO are summarized in Table 7. The same set of hyperparameters is applied to all variants of the model proposed in this paper to ensure a consistent training setup.

Table 7: HARMO Training Hyperparameters

Hyperparameter	**Value**
Training batch size	256
Rollout batch size	256
Samples per prompt	8
Temperature	1
Max output sequence length	4096
Max epochs	1
Number of episodes	2
Initial KL coefficient	$1e^{-3}$
Discount factor (γ)	1
GAE parameter (λ)	1
Actor learning rate	$1e^{-6}$
Min normalization value (P_{min})	-10
Max normalization value (P_{max})	10

A.4 HARMO Reward Normalization

The core HARMO reward mechanism consists of two main components: (i) the hybrid accuracy reward R^{hybrid}, which integrates the rule-based reward R^{ruled} and the learned reward model score R^{RM}, and (ii) behavioral regularization components, including the length penalty R^{λ} and the formatting reward R^{fmt}.

Since each reward component is computed independently and may exhibit different scales and distributions, normalization is essential to reduce variance and ensure stable training. In this section, we detail the computation of normalization factors and describe the normalization strategy applied to each reward component. Specifically, we decompose the HARMO reward function and explain how the normalization of individual raw rewards contributes to the final HARMO reward.

To harmonize heterogeneous reward sources, namely, binary rewards and scalar rewards produced by the learned reward model, we rescale all rewards to a common scale. Binary rewards are aligned with the statistical distribution of the learned reward model (R^{RM}) scores to ensure consistent contribution across components.

All reward values are normalized to the range $(P_{\text{min}}, P_{\text{max}})$. The normalization bounds are de-

```
{
  "message": [
    {
      "role": "system",
      "content": "You are a helpful assistant good at solving problems with step-by-step reasoning.
You should first think about the reasoning process and then provide the user with the answer."
    },
    {
      "role": "user",
      "content": [
        {
          "type": "image",
          "image": "path/to/image.png"
        },
        {
          "type": "text",
          "text": "Subtract 0 yellow cylinders. How many objects are left?
Output the step-by-step reasoning in <think> </think> and the final answer in <answer> </answer> tags.
The <answer> tag should include the final answer.
The output answer format should be as follows:
<think>Your reasoning process</think> <answer>Final answer</answer>."
        }
      ]
    }
  ],
  "reasoning": "<think>
Okay, let's see. The user is asking to subtract 0 yellow cylinders and then determine how many objects are left.
First, I need to visualize the scene described. ... (truncated for brevity) ... So the answer is five.
</think>
<answer>
The scene contains five distinct objects: a green sphere, a brown metallic sphere, a purple cylinder, a cyan sphere, and a gray cube.
Since there are no yellow cylinders present to subtract, all objects remain. The total count stays at **5**.
</answer>",
  "answer": "5",
  "answer_format": "number_format"
}
```


Figure 2: Example numeric (vision + text) entry from the dataset.

termined based on the empirical distribution of the learned reward model scores used by R^{RM}. In our implementation, we set $P_{\min} = -10$ and $P_{\max} = 10$. These values are obtained by evaluating the reward model on the uniformly sampled MMIF-23k dataset.[4]

Overall, the normalization strategy for each reward component is summarized as follows:

- R^{ruled}: returns a binary value indicating response correctness, which is rescaled to -10 or 10.
- R^{RM}: returns a scalar value produced by the learned reward model, naturally bounded within $[-10, 10]$.
- R^{fmt}: returns a binary value based on format correctness, rescaled to -10 or 10.
- R^{λ}: represents a length-based penalty; penalized responses receive a value of -10, while non-penalized responses receive 0.

B Dataset Details

The example in Figure 2 illustrates a verifiable numeric response with an image input.

[4]https://huggingface.co/datasets/ChrisDing1105/MMIF-23k

C Evaluation Benchmarks

We conduct a comprehensive evaluation across a diverse set of benchmarks to rigorously assess model capabilities. Mathematical reasoning is evaluated using MathVerse (Zhang et al., 2024), MATH-Vision (Wang et al., 2024), and MathVista (Lu et al., 2024). Multi-disciplinary reasoning is measured with MMMU (Yue et al., 2024) and MMMU-Pro (Yue et al., 2025). Finally, general visual question answering performance is tested on AI2D (Kembhavi et al., 2016), ChartQA (Masry et al., 2022), and DocVQA (Mathew et al., 2021). All evaluations were executed using the open-source LLMs-Eval framework (Zhang et al., 2025) under identical conditions (e.g., system prompts, response token limits) to ensure methodological consistency.

D Case Study

This section includes a case study illustrating HARMO's ability to reason through a variety of problem-solving questions. In these visualizations, the reasoning steps are color-coded to indicate their verification status:

- Text highlighted with indicates correctly verified reasoning steps.

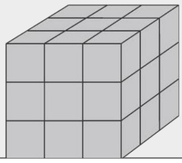
- Text highlighted with (green) indicates advanced or highly sophisticated reasoning steps.
- Text highlighted with (red) indicates incorrectly verified or flawed reasoning steps.

The following figures demonstrate HARMO's step-by-step problem-solving capabilities and provide a visual comparison with baseline model.

System Prompt :
You are a helpful assistant good at solving problems with step-by-step reasoning. You should first think about the reasoning process and then provide the user with the answer.

User Prompt :
A cube with the edge 3 cm long is painted grey and cut into smaller cubes each with an edge of 1 cm long. How many smaller cubes will have exactly 2 faces painted?
Output the step-by-step reasoning in <think> </think> and the final answer in <answer> </answer> tags.
The <answer> tag should include the final answer.
The output format should be:
<think>Your reasoning process</think> <answer>Final answer</answer>.

Input Image:



HARMO-3B answer:

<think>
To determine how many smaller cubes will have exactly 2 faces painted, we need to analyze the structure of the larger cube and the smaller cubes it is divided into.

1. The larger cube has an edge length of 3 cm. When it is divided into smaller cubes each with an edge length of 1 cm, it will be divided into 3 x 3 x 3 = 27 smaller cubes.

2. The smaller cubes that have exactly 2 faces painted are the cubes that are located on the edges of the larger cube, but not at the corners. Each edge of the larger cube is 3 cm long, and when divided into smaller cubes, it will have 3 smaller cubes along each edge. The cubes at the corners are painted on three faces, the cubes at the edges but not at the corners are painted on two faces, and the cubes at the vertices are painted on four faces.

3. Each edge of the larger cube has 3 smaller cubes, but the two cubes at the ends of each edge are corner cubes and are painted on four faces. Therefore, the number of smaller cubes with exactly two faces painted on one edge is 3 - 2 = 1. Since there are 12 edges in a cube, the total number of smaller cubes with exactly two faces painted is 12 x 1 = 1.

Thus, the number of smaller cubes that have exactly 2 faces painted is 12.
</think>
<answer>12</answer>

Qwen2.5-VL-3B-Instruct answer:
<think>
To solve this problem, we need to consider how the original cube is divided into smaller cubes and which smaller cubes will have exactly two faces painted.

1. The original cube has an edge length of 3 cm. When it is cut into smaller cubes with an edge length of 1 cm, it will result in a 3x3x3 grid of smaller cubes.

2. In a 3x3x3 grid, the smaller cubes that have exactly two faces painted are those located on the edges of the larger cube but not at the corners. This means we need to count the number of such cubes along each edge.

3. Each edge of the original cube is 3 cm long, so there are 3 smaller cubes along each edge (including the corner cubes).

4. Since there are 3 edges, and each edge contributes 3 smaller cubes with exactly two faces painted, we calculate the total number as follows:

Total = 3 edges * 3 cubes per edge = 9 cubes
Therefore, there are 9 smaller cubes that will have exactly two faces painted.

</think>
<answer>
9
</answer>

Figure 3: **Case Study 1 - Math Cube Problem**

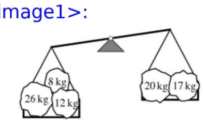
HARMO correctly identifies all 12 edge-centered cubes with exactly two painted faces. The baseline model, by contrast, fails to systematically exclude corner cubes or miscounts edge cases, often yielding an incorrect answer.

System Prompt :
You are a helpful assistant good at solving problems with step-by-step reasoning. You should first think about the reasoning process and then provide the user with the answer.

User Prompt :
Which stone should Mr Flintstone place on the right side of the scales, so that both sides weigh the same?
Options: ['A' 'B' 'C' 'D' 'E']
Output the step-by-step reasoning in <think> </think> and the final answer in <answer> </answer> tags.
The <answer> tag should include the correct option choice.
The output format should be:
<think>Your reasoning process</think> <answer>Correct option choice</answer>.

Input Image:

<image1>:



<image2>:

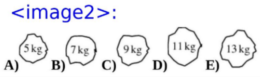


Figure 4: **Case Study 2 - Solving a General Math Problem (Part 1)**

HARMO-3B answer:

<think>
To determine which stone Mr. Flintstone should place on the right side of the scales to make both sides weigh the same, we need to calculate the total weight on the left side of the scales first.
The left side of the scales has three stones:

- A stone weighing 26 kg
- A stone weighing 12 kg
- A stone weighing 8 kg

Let's add these weights together:
26 kg + 12 kg + 8 kg = 46 kg

The right side of the scales has two stones:

- A stone weighing 20 kg
- A stone weighing 17 kg

Let's add these weights together:
20 kg + 17 kg = 37 kg
To make both sides weigh the same, we need to find a stone that, when added to the right side, will make the total weight equal to 46 kg. Let's denote the weight of the stone to be placed on the right side as x. We need to solve the equation:
37 kg + x = 46 kg

Subtract 37 kg from both sides:
x = 46 kg - 37 kg = 9 kg

Therefore, Mr. Flintstone should place a stone weighing 9 kg on the right side of the scales.
</think>
<answer>C</answer>

Qwen2.5-VL-3B-Instruct answer:

<think>
To solve this problem, we need to determine which weight will balance the scale when placed on the right side. First, let's calculate the total weight on the left side of the scale:

Left side weights: 26 kg + 12 kg + 8 kg = 46 kg

Now, we need to find out how much weight is needed on the right side to balance it. Since the total weight on the left side is 46 kg, the right side must also have 46 kg to balance.

Let's check each option to see which one adds up to 46 kg:

A) 5 kg: 46 kg (not enough)
B) 7 kg: 46 kg (not enough)
C) 9 kg: 46 kg (not enough)
D) 11 kg: 46 kg (not enough)
E) 13 kg: 46 kg (enough)

Therefore, the correct weight to place on the right side to balance the scale is 13 kg.

</think>
<answer>E)</answer>

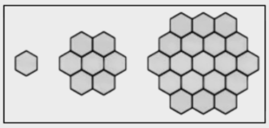
Figure 4: **Case Study 2 - Solving a General Math Problem (Part 2)**

HARMO correctly reasoned step-by-step to arrive at the correct answer, whereas the baseline model followed unreasonable steps and produced an incorrect solution.

System Prompt :
You are a helpful assistant good at solving problems with step-by-step reasoning. You should first think about the reasoning process and then provide the user with the answer.

User Prompt :
Consider the given image where there are three patterns shown. Analyze and determine the number of hexagons in the fifth pattern.
Output the step-by-step reasoning in <think> </think> and the final answer in <answer> </answer> tags.
The <answer> tag should include the correct option choice.
The output format should be:
<think>Your reasoning process</think> <answer>Correct option choice</answer>.

Input Image:



HARMO-3B answer:

<think>
Let's analyze the pattern of hexagons in Sylvia's drawings:
1. The first pattern has 1 hexagon.
2. The second pattern has 7 hexagons.
3. The third pattern has 19 hexagons.

We need to find the number of hexagons in the fifth pattern. Let's denote the number of hexagons in the n-th pattern as H(n).

From the given patterns, we can see that:
H(1) = 1
H(2) = 7
H(3) = 19

To find a pattern, let's calculate the difference between consecutive terms:
H(2) - H(1) = 7 - 1 = 6
H(3) - H(2) = 19 - 7 = 12

The differences are 6 and 12. We can see that the difference is increasing by 6 each time. So, the difference between the number of hexagons in the n-th pattern and the n-1-th pattern is 6(n-1).

Therefore, we can express H(n) as:
H(n) = H(n-1) + 6(n-1)

Using this recurrence relation, we can find H(4) and H(5):
H(4) = H(3) + 6(3) = 19 + 18 = 37
H(5) = H(4) + 6(4) = 37 + 24 = 61

So, the number of hexagons in the fifth pattern is 61.
</think>
<answer>61</answer>

Qwen2.5-VL-3B-Instruct answer:

<think>
To determine the number of hexagons in the fifth pattern, we need to observe the pattern and identify the rule governing the growth.

1. **First Pattern**: There is 1 hexagon.
2. **Second Pattern**: There are 7 hexagons (1 + 6).
3. **Third Pattern**: There are 19 hexagons (7 + 12).

We can see that each subsequent pattern adds more hexagons than the previous one. To find the number of hexagons in the fourth and fifth patterns, we continue this pattern:

4. **Fourth Pattern**: The number of hexagons is (19 + 18 = 37).
5. **Fifth Pattern**: The number of hexagons is (37 + 20 = 57).

Therefore, the fifth pattern will have 57 hexagons.
</think>

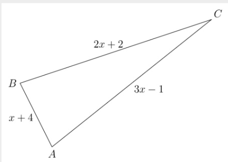
<answer>57</answer>

Figure 5: **Case Study 3 - Math Puzzle Problem.** The baseline failed to identify the correct puzzle equation, whereas HARMO succeeded in finding the correct puzzle equation, reasoning step-by-step, and arriving at the correct solution.

System Prompt :
You are a helpful assistant good at solving problems with step-by-step reasoning. You should first think about the reasoning process and then provide the user with the answer.

User Prompt :
In the triangle, ∠A=∠B. What is x?
Output the step-by-step reasoning in <think> </think> and the final answer in <answer> </answer> tags.
The <answer> tag should include the final answer.
The output format should be:\n<think>Your reasoning process</think> <answer>Final answer</answer>.

Input Image:



HARMO-3B answer:

<think>
Since ∠ A = ∠ B in the triangle ABC, then AC = BC. Given the lengths of the sides AC = 3x - 1 and BC = 2x + 2, we can set up the equation:
3x - 1 = 2x + 2
Subtract 2x from both sides:
x - 1 = 2
Add 1 to both sides:
x = 3
Thus, the value of x is 3.
</think>
<answer>3</answer>

Qwen2.5-VL-3B-Instruct answer:

Since ∠ A = ∠ B, triangle ABC is isosceles with AB = AC. Therefore, we can set up the equation x + 4 = 2x + 2.
Solving for x:
x + 4 = 2x + 2
Subtract x from both sides:
4 = x + 2
Subtract 2 from both sides:
2 = x
So, x = 2.
</think>
<answer>2</answer>

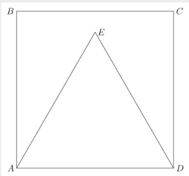
Figure 6: **Case Study 4 - Geometry Math Problem**

HARMO correctly solves for x with clear recognition of equal sides $AB = AC$, while the baseline model fails to identify the isosceles nature, leading to incorrect conclusions.

System Prompt :
You are a helpful assistant good at solving problems with step-by-step reasoning. You should first think about the reasoning process and then provide the user with the answer.

User Prompt :
Square ABCD and equilateral triangle AED are coplanar and share the line segment AD, as shown. What is the measure, in degrees, of angle BAE?
Output the step-by-step reasoning in <think> </think> and the final answer in <answer> </answer> tags.
The <answer> tag should include the final answer.
The output format should be:\n<think>Your reasoning process</think> <answer>Final answer</answer>.

Input Image:



HARMO-3B answer:

<think> Given the square ABCD and the equilateral triangle AED, we need to find the measure of angle BAE.

1. Since ABCD is a square, all angles in the square are 90 degrees and all sides are equal. Therefore, ∠DAB = 90°.
2. Since AED is an equilateral triangle, all its angles are 60 degrees and all its sides are equal. Therefore, ∠EAD = 60°.
3. We can find ∠BAE by subtracting ∠DAB from ∠EAD:
   ∠BAE = ∠EAD – ∠DAB = 60° – 90° + 90° = 30°
   Here, we subtract 90 degrees from 60 degrees because ∠DAB is the sum of ∠DAE and ∠EAB, and ∠EAB is 90 degrees.

Therefore, the measure of ∠BAE is 30 degrees.
</think>
<answer>30</answer>

Qwen2.5-VL-3B-Instruct answer:

<think> Given that square ABCD and equilateral triangle AED share side AD, we can deduce the following:

1.Since ABCD is a square, all its internal angles are 90°.

2 Since AED is an equilateral triangle, all its internal angles are 60°.

To find the measure of angle BAE, we need to consider the geometry formed by these shapes.

- The angle BAD is part of the square, so it is 90°.
- The angle EAD is part of the equilateral triangle, so it is 60°.

Since B and D are adjacent vertices of the square, the line segment BD forms a straight line passing through both points. Therefore, the angle BAE is the sum of the angles BAD and EAD.

Thus, angle BAE = 90° + 60° = 150°.
</think>
<answer>150°</answer>

Figure 7: **Case Study 5 - Geometry Math Problem**

HARMO correctly finds $\angle BAE = 30^\circ$, while the baseline incorrectly sums angles to 150° with incorrect geometric reasoning, misinterpreting the figure's layout.