Generative RLHF-V: Learning Principles from Multi-modal Human Preference

Jiayi Zhou[*,1,2], **Jiaming Ji**[*,1,2], **Boyuan Chen**[1], **Jiapeng Sun**[4], **Wenqi Chen**[1]
Donghai Hong[1], **Sirui Han**[3], **Yike Guo**[3], **Yaodong Yang**[†,1]

[1]Institute for Artificial Intelligence, Peking University
[2]State Key Laboratory of General Artificial Intelligence, Peking University
[3]Hong Kong University of Science and Technology
[4]University College London
{gaiejj,jiamg.ji,cbylll}@stu.pku.edu.cn
yaodong.yang@pku.edu.cn

Abstract

Training multi-modal large language models (MLLMs) that align with human intentions is a long-term challenge. Traditional score-only reward models for alignment suffer from low accuracy, weak generalization, and poor interpretability, blocking the progress of alignment methods, *e.g.,* reinforcement learning from human feedback (RLHF). Generative reward models (GRMs) leverage MLLMs' intrinsic reasoning capabilities to discriminate pair-wise responses, but their pair-wise paradigm makes it hard to generalize to learnable rewards. We introduce Generative RLHF-V, a novel alignment framework that integrates GRMs with multi-modal RLHF. We propose a two-stage pipeline: **multi-modal generative reward modeling from RL**, where RL guides GRMs to actively capture human intention, then predict the correct pair-wise scores; and **RL optimization from grouped comparison**, which enhances multi-modal RL scoring precision by grouped responses comparison. Experimental results demonstrate that, besides out-of-distribution generalization of RM discrimination, our framework improves 4 MLLMs' performance across 7 benchmarks by 18.1%, while the baseline RLHF is only 5.3%. We further validate that Generative RLHF-V achieves a near-linear improvement with an increasing number of candidate responses. Our code and models can be found at https://generative-rlhf-v.github.io.

1 Introduction

> **"The mediocre teacher tells. The great teacher inspires."**
>
> — William Arthur Ward saying – Education

Human interaction and learning are naturally multi-modal [1, 2, 3]. Recent research has demonstrated significant advances in multi-modal large language models (MLLMs) [4, 5, 6, 7] on visual question answering and reasoning tasks. These breakthroughs reveal two critical insights: 1) Reinforcement learning (RL) substantially enhances MLLMs' capacity for solving complex problems [8]; 2) The efficacy of RL fundamentally depends on the precisely defined reward (*e.g.*, rule-based verification for mathematical correctness). While rule-based rewards can be effectively constructed for logical reasoning and factual judgment tasks [9, 10], accurate reward modeling for human values, *e.g.,* instruction-following or safety [11, 12], remains a long-term challenge in MLLMs alignment.

[*]Equal contribution, [†]Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

 https://doi.org/10.52202/085713-4201

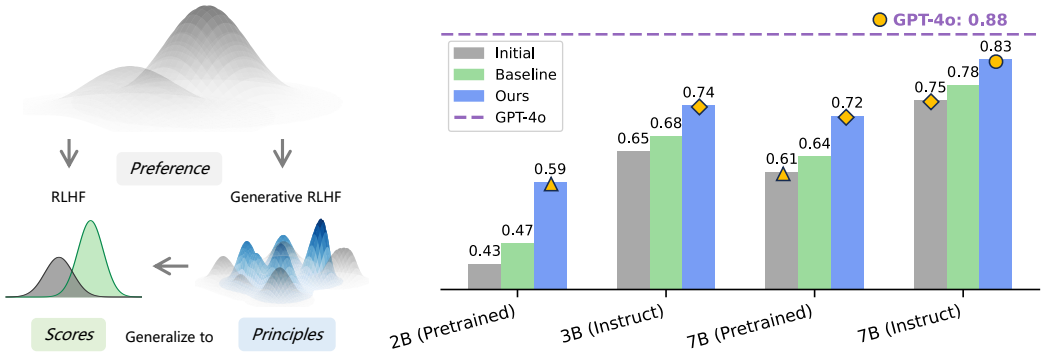


Figure 1: Advanced multi-modal large language models (MLLMs) is calling principled preference learning. In MLLM's alignment, traditional RLHF methods only learn scalar scores from preferences. In contrast, our Generative RLHF-V can learn principles from preferences and optimize based on this comprehensive comparison. Experimental results show that Generative RLHF-V elevates 2B and 3B MLLMs to 7B performance across 7 benchmarks. It also advances pretrained models to instruct model capabilities and enables open-source models to match closed-source experts.

Alignment aims to make AI systems adhere to human intentions [13, 12], and for MLLMs, these goals can be concretized into the 3H standards: helpful, harmless, and honest [14, 15, 16]. These goals are difficult to represent as a symbolic reward [17, 18]. Traditional approaches typically employ an additional score head to project the final-layer activations of MLLMs into scalar rewards [11, 9, 19], *i.e.,* score-only reward model (RM), subsequently applying Bradley-Terry loss to learn human preferences from pairwise comparisons. However, extensive studies [20, 21, 22] have exposed three fundamental limitations of this paradigm: low accuracy, weak generalization, and poor interpretability. Generative reward models (GRMs) [23, 24] present a promising alternative by leveraging LLMs' intrinsic reasoning capabilities to discriminate pair-wise responses [25, 26], and rule-based RL fine-tuning strengthens this capability [27]. Nevertheless, the practical application of such GRMs in multi-modal RLHF remains to be verified. This progression presents the following urgent dilemma:

- **Advanced MLLMs is calling principled preference learning.** As MLLMs become more sophisticated, they handle increasingly complex inputs and diverse tasks [28]. Human assessment of preferences for MLLMs' responses will also grow more varied and intricate [29]. Consequently, relying on a single inference from a score-only RM proves insufficient for learning generalizable human preferences [30, 20], thereby creating a bottleneck in MLLMs alignment.
- **Pair-wise comparison is blocking multi-modal principles from generalizing to learnable rewards.** While pair-wise comparison allows GRM to learn generalizable principles from RL [27, 31, 32], this pair-wise comparison feedback does not readily translate into the point-wise scores, which are essential for RL optimization [33, 34, 35, 36]. This disconnect hinders the ability of learned principles to effectively guide the multi-modal RLHF.

In response, we propose **Generative RLHF-V**(ision), as shown in Figure 2, a novel alignment framework enabling the pair-wise multi-modal GRM with multi-modal RLHF. Our pipeline consists of two stages: **multi-modal generative reward modeling from RL** and **RL optimization from grouped comparison**. The first component utilizes RL to train a GRM to learn principles from multimodal preferences, which then performs strongly generalizable pairwise scoring of responses. The second component applies these GRM-learned principles to obtain more precise scores by comparing within groups of responses. Our GRM training extends the self-principled critique tuning (SPCT) [27] to the vision scenario, training MLLMs as GRMs using RL, with rule-based rewards from annotated ground truth in preference datasets.

In contrast to SPCT, we find that in the multi-modal scenario, enabling GRMs to explore principles autonomously yields superior generalization than selecting principles from a reference set. Our grouped comparison design enables the generalization of learned principles from pair-wise comparisons to point-wise scores. This further unveils a novel direction for post-training scaling up: as the number of candidate responses n explored by online RL increases, GRMs can assign more accurate scores, leading to improved RL performance near linearly. Our key contributions are as follows:

https://doi.org/10.52202/085713-4201

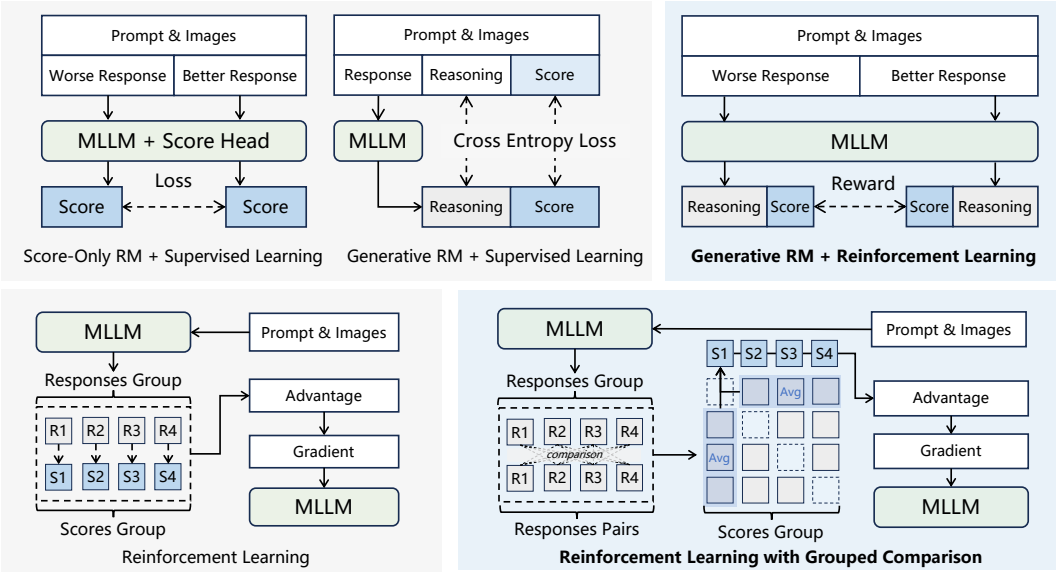


Figure 2: Comparison of our pipelines to traditional ones. **For reward modeling**, we make generative RM actively reason about the advantages and disadvantages between two answers, and output corresponding scores. If the better response gets a higher score, it provides a positive reward. **For RL optimization**, we compare responses in pairs within a group to obtain more accurate scores.

- **RL-based GRMs for learning principles from multi-modal preference:** We develop a multi-modal GRM trained via RL, enabling the reasoning of principles and precise reward predictions, achieving average **20.4%** accuracy improvement on out-of-distribution discriminative tasks.
- **Multi-modal generative RLHF:** We empirically demonstrate the superiority of GRMs for multi-modal RLHF. Experimental results demonstrate that our framework improves MLLMs' performance across 7 benchmarks by **18.1%**, while the baseline RLHF is only **5.3%**.
- **Grouped comparison for post-training scaling up:** We discovered that the integration of GRM+RL with grouped comparison enables the performance of RL optimization to near linearly improve with an increasing number of candidate responses n within a certain range. The removal of either component negates this observed enhancement.
- **A pioneer case study of multi-modal GRM reward hacking:** We find that RL over-training under an over-trained GRM can lead models to adopt *self-praise* behaviors to obtain high rewards, even achieving exceptionally high scores on benchmarks employing the MLLM-as-judge paradigm.

2 Related Work and Preliminaries

MLLM Alignment and RLHF. AI alignment is the deliberate process of shaping model behavior to cohere with human goals, values, and ethical principles [11, 13, 37, 12, 38, 39]. Achieving robust alignment faces challenges in translating complex, subjective, and evolving human values into quantifiable training objectives [20, 40]. Current MLLM alignment methods mainly rely on post-training [41, 42, 19], with RL fine-tuning based on human preferences being the most mainstream approach. This process typically involves two key stages: reward modeling and RL optimization.

A score-only reward model R_θ is trained on a dataset of human preferences, where each data point includes a prompt $\boldsymbol{x}$, a preferred response $\boldsymbol{y}^w$, and a dispreferred response $\boldsymbol{y}^l$. The model learns to assign a higher scalar score to the preferred response s^w than the dispreferred one s^l using a pairwise ranking loss, typically minimizing $\mathcal{L}_{RM} = -\sum \log \sigma(s_i^w - s_i^l)$. This trained R_θ serves as an automated judge of response quality. The loss function can be expressed as minimizing the negative log-likelihood over the dataset:

$$\mathcal{L}_{RM}(\theta) = -\sum_{i=1}^{N} \log \sigma(R_\theta(\boldsymbol{x}_i, \boldsymbol{y}_i^w) - R_\theta(\boldsymbol{x}_i, \boldsymbol{y}_i^l)) = -\sum_{i=1}^{N} \log \sigma(s_i^w - s_i^l).$$

https://doi.org/10.52202/085713-4201

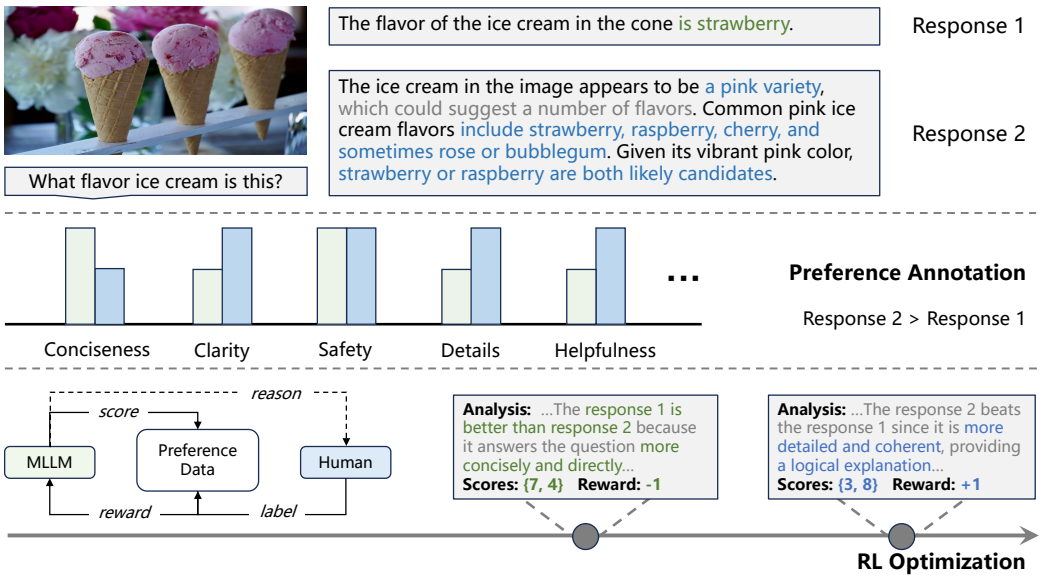


Figure 3: An example of generative reward modeling from RL. The goal of RL is to make MLLMs assign higher scores to responses that align with human preferences. Through RL optimization, MLLMs can infer the underlying principle behind how humans annotate these binary preferences.

The MLLM's policy (π_ϕ) is fine-tuned using RL. For a given prompt $\boldsymbol{x}$ from a given dataset $\mathcal{D}_{\text{prompt}}$, the policy generates a response $\boldsymbol{y}$, which is then scored by the reward model R_θ. The policy parameters ϕ are updated to maximize this reward. To prevent the policy from deviating too much from the original pre-trained model (π_ϕ^{base}) and maintain coherence, a Kullback-Leibler (KL) divergence penalty is added to the optimization objective: $\max_\phi \mathbb{E}[R_\theta(\boldsymbol{x}, \boldsymbol{y}) - \beta \text{KL}(\pi_\phi || \pi_\phi^{\text{base}})]$, where β is a fixed hyper-parameter. The final optimization objective is:

$$\max_\phi \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}_{\text{prompt}}, \boldsymbol{y}\sim\pi_\phi(\cdot|\boldsymbol{x})}[R_\theta(\boldsymbol{x}, \boldsymbol{y}) - \beta \text{KL}(\pi_\phi(\cdot|\boldsymbol{x}) || \pi_\phi^{\text{base}}(\cdot|\boldsymbol{x}))].$$

Generative Reward Model. Generative reward models (GRM) [24, 23, 22] offer an alternative paradigm to score-only reward modeling, which leverages the inherent generative capabilities of MLLMs to evaluate preferences. Current research on GRMs for MLLM alignment focuses on employing supervised learning methods to improve accuracy [25]. A representative method is LLaVA-Critic [26], which collects expert-annotated point-wise and pair-wise scores, along with reasoning traces for MLLM question-answer pairs, subsequently training the MLLM as a GRM via supervised learning. Despite its superior performance, this approach necessitates more expensive expert annotations compared to binary preference datasets and imposes stricter requirements for the reasoning trace annotations. Moreover, there is a notable lack of empirical studies on applying GRMs in RL training. To date, explorations of GRM applications have centered on data filtering for Best-of-N selection and offline direct preference optimization [38]. The practical implementation of GRM within the multi-modal RL optimization is yet to be investigated.

3 Generative RLHF-V

The Generative RLHF-V pipeline mainly consists of two parts: generative reward modeling from reinforcement learning (RL) and RL from grouped comparison. The former references training MLLMs through RL as a pair-wise vision generative reward model (GRM), which actively reasons about the human principle behind two given responses and provides a pair-wise score comparison. The latter leverages the characteristics of this GRM, collecting multiple responses for a given input and providing more accurate grouped scoring for them.

Multi-modal Generative Reward Modeling from RL. We consider the task of training a pair-wise GRM R_θ with parameters θ using RL guided by human preferences. The goal of R_θ is to assign a pair of scalar scores $\{\boldsymbol{s}_1, \boldsymbol{s}_2\}$ to a pair of responses $\{\boldsymbol{y}_1, \boldsymbol{y}_2\}$ for a given prompt $\boldsymbol{x}$. We are given a dataset

https://doi.org/10.52202/085713-4201

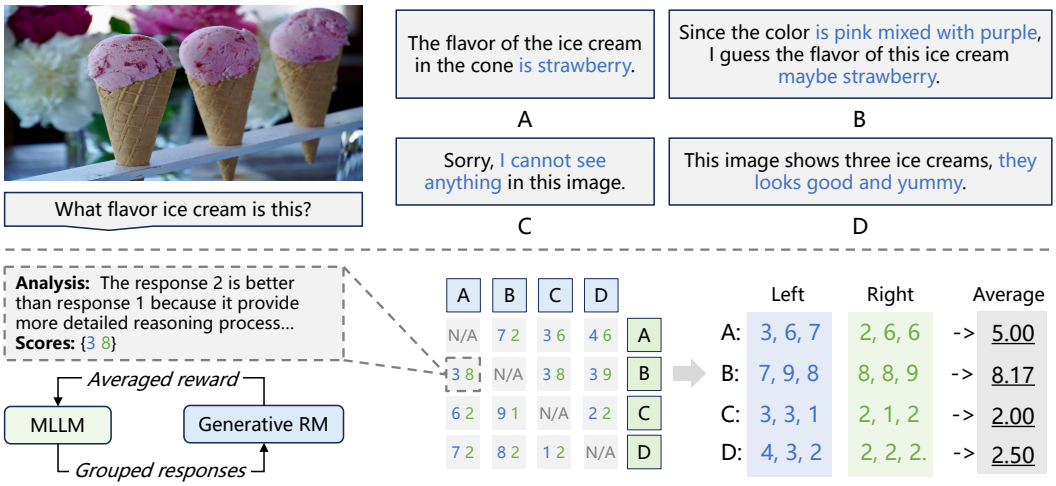


Figure 4: An example of RL from grouped comparison. Its advantage lies in utilizing grouped comparisons to achieve more accurate scoring. Response B provides accurate and comprehensive information, thus receiving the highest score; although response A is somewhat arbitrary, it performs accurate image recognition and obtains a higher score than C and D.

of human preferences $\mathbb{D}^{\mathbb{P}} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i^w, \boldsymbol{y}_i^l)\}_{i=1}^N$, where $\boldsymbol{x}_i$ is a prompt, $\boldsymbol{y}_i^w$ is the response preferred and $\boldsymbol{y}_i^l$ is the response dispreferred by a set of human preference principles $\mathbb{P} = \{p_1, p_2, \ldots, p_k\}$. The inference of reward model, denoted as π_{GRM}^θ (or simply R_θ), is a parameterized function that takes the prompt $\boldsymbol{x}$, and responses pairs $\boldsymbol{y}^w$ and $\boldsymbol{y}^l$ as input and outputs the predicted principles $\mathbb{P}^*$, reasoning traces r, and a pair-wise scalar score s^w and s^l:

$$\{\mathbb{P}^*, r, s^w, s^l\} = R_\theta(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l).$$

The model's preference prediction should be $s_w > s_l$. And the reward r for a given preference pair $(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l)$ is determined by comparing the scores assigned by R_θ:

$$r(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l; \theta) = \begin{cases} +1 & \text{if } s^w > s^l, \\ -1 & \text{if } s^w \leq s^l. \end{cases}$$

The RL objective is to maximize the expected reward over the preference dataset $\mathcal{D}$.

$$\max_\theta \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l) \sim \mathcal{D}} \left[r(\boldsymbol{x}, \boldsymbol{y}^w, \boldsymbol{y}^l; \theta) \right].$$

Reinforcement Learning from Grouped Comparison. This stage is to fine-tune the MLLM, denoted as π_ϕ with parameters ϕ, using RL guided by grouped comparisons. This phase leverages the pair-wise scoring capabilities of the GRM to obtain a more precise score via grouped comparison, optimizing for principles $\mathbb{P}$ implicitly learned by the GRM.

The core idea is to utilize the GRM as a judge to evaluates multiple candidate responses generated by the MLLM π_ϕ for the same input. This grouped comparison provides a stronger reward for the RL algorithm compared to using a single point-wise score. For the given input $\boldsymbol{x}$, we use the MLLM policy π_ϕ to generate a set of n distinct responses, $\mathcal{Y} = \{\boldsymbol{y_1}, \boldsymbol{y_2}, \ldots, \boldsymbol{y_k}\}$, where $n > 1$. Each generated response $\boldsymbol{y}_i$ in the set $\mathcal{Y}$ is evaluated using the pre-trained GRM, R_ϕ. Each generated response is evaluated using the pre-trained GRM, R_θ. To obtain the final score $S(\boldsymbol{y}_i)$, the method aggregates scores from pair-wise comparisons against all other responses in the set $\mathcal{Y}$.

Specifically, for each response $\boldsymbol{y}_i$, we consider its comparison with every other response $\boldsymbol{y}_j$ (where $j \neq i$). The GRM function $R_\theta(\boldsymbol{x}, \boldsymbol{y}_a, \boldsymbol{y}_b)$ outputs a pair of scores. Let $s(\boldsymbol{y}_a | \boldsymbol{y}_a, \boldsymbol{y}_b)$ denote the score assigned to response $\boldsymbol{y}_a$ extracted from $R_\theta(\boldsymbol{x}, \boldsymbol{y}_a, \boldsymbol{y}_b)$. The final grouped comparison score $S(\boldsymbol{y}_i)$ for response $\boldsymbol{y}_i$ is calculated by averaging the scores assigned to $\boldsymbol{y}_i$ across all possible pair-wise comparisons involving it:

$$S(\boldsymbol{y}_i) = \frac{1}{2(k-1)} \sum_{j=1, j \neq i}^{k} \left(s(\boldsymbol{y}_i | \boldsymbol{y}_i, \boldsymbol{y}_j) + s(\boldsymbol{y}_i | \boldsymbol{y}_j, \boldsymbol{y}_i) \right).$$

https://doi.org/10.52202/085713-4201

The set of scores $\{S(\boldsymbol{y}_1), S(\boldsymbol{y}_2), \ldots, S(\boldsymbol{y}_k)\}$ serves as the reward for fine-tuning the MLLM policy π_ϕ using RL, guiding it to generate responses that are preferred according to the principle implicitly learned by the GRM.

4 Experiment

Generative RLHF-V integrates two key components: generative reward modeling via reinforcement learning and grouped comparisons. subsection 4.2 evaluates these components from a **reward modeling** standpoint, focusing on their performance in pair-wise discrimination and point-wise scoring. Subsequently, from the **RL optimization** angle, subsection 4.3 analyzes their improvement on RL performance, ablation studies insights, and reward hacking in over-trained scenarios.

4.1 Experimental Setup

Our experiments were conducted on servers equipped with 8 * Nvidia H800 GPUs. We utilized verl[2] for RL training and align-anything[3] for reward modeling and supervised fine-tuning. Further details on the experimental setup can be found in the section 7.

Models. We selected MLLMs of varying sizes, encompassing both pre-trained and instruction-tuned variants. Specifically, we utilize the Qwen2-VL [43] models in 2B and 7B parameter sizes, and the Qwen2.5-VL-Instruct [44] models in 3B and 7B sizes. For the generative reward modeling phase, the instruct models series served as the starting point, leveraging their inherent instruction-following capabilities. In the subsequent RL optimization experiments, the 3B parameter RM was used to supervise the 2B and 3B models, while the 7B reward model supervised the 7B models.

Datasets. We focused on the helpful and harmless alignment for MLLMs, selecting corresponding preference datasets. For the helpfulness, we utilized a 30k preference dataset from Align-Anything [45], the text-image-to-text part. The preference principle in this dataset emphasizes instruction following, clarity, and informativeness. For the harmlessness, we employed Beavertails-V [46] which includes 20 distinct categories of safety-related red-teaming prompts.

Benchmarks. We selected 7 benchmarks to validate the effectiveness of Generative RLHF-V. These are MIA-Bench [47], LLaVA-Bench-In-The-Wild [48], LLaVA-Bench-Wilder [49], MM-Vet [50], and MM-Vet-v2 [51] (for helpfulness), as well as MM-SafetyBench [52] and MSS-Bench [53] (for harmlessness). These benchmarks encompass both pair-wise evaluations, which involve a golden response for comparison, and point-wise scoring methodologies based on specific criteria.

Implementation Details. Since our method utilizes a GRM trained by RL (**GRM+RL, ours**), we established 3 baselines: a score-only **RM** trained with the Bradley-Terry loss, an untrained **GRM**, and a GRM trained via supervised learning loss (**GRM+SFT**). The objective of SFT is the annotation principle of the preference dataset and the scores assigned to responses, since Align-Anything and Beaver-V both contain overall response scores, which we scale to match the RL setting's range. In RL optimization, the score-only RM assigns point-wise scores to each response, while the GRM collects scores via grouped comparisons. We mainly use the GRPO for RL experiments, which by default collects $n = 5$ candidate responses per iteration.

4.2 Principles Learning of RL-Based GRMs

RQ1: Does the GRM+RL facilitate more generalizable principle learning from preferences?

We evaluated a series of RMs based on Qwen2.5-VL-7B-Instruct and trained on Align-Anything by comparing their accuracy generalization across 3 out-of-distribution (OOD) preference datasets. As illustrated in Figure 5, GRMs outperformed score-only RMs on these OOD tasks. Notably, GRMs + RL achieved the highest accuracy.

Since GRMs have in-context learning capabilities, we further investigated their performance when provided with the principles of each preference dataset. As shown in the (P) results of Figure 5, the performance of GRM and GRM+SFT improved, whereas that of GRM+RL declined. We think that the GRM+SFT potentially overfits to their training data, struggles to autonomously generate

[2] https://github.com/volcengine/verl

[3] https://github.com/PKU-Alignment/align-anything

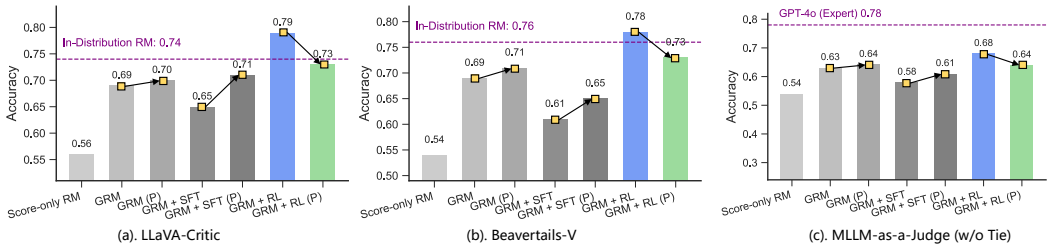


Figure 5: Comparison of RMs accuracy on OOD discriminative tasks. (P) denotes the concatenation of the annotation principle from the corresponding preference dataset to the models' output, serving as hints for inference. All models represented by the bar charts were trained on the Align-Anything dataset. The purple dashed line indicates expert performance. For Beaver-V and LLaVA-Critic, we trained in-distribution RMs to serve as the expert baseline. In the case of MLLM-as-a-judge, given its limited data volume, we directly utilized the SOTA GPT-4o as the expert.

Table 1: Performance of GRMs in the MLLM-as-a-Judge `Score` task, measured by the Pearson correlation coefficient.

Models	w/ GC	w/o GC
GRM	0.41	0.38
GRM+SFT	0.37	0.33
GRM+RL	0.43	0.37
GPT-4o (Expert)	0.48	0.46

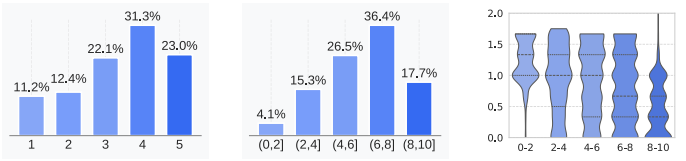


Figure 6: The scoring distribution of the GRM+RL model on MLLM-as-a-judge's `Score` task. Figure (a) is the annotated human scores, while Figure (b.) is GRM+RL scores and Figure (c). is its fine-grained scores distribution.

appropriate principles from response pairs, and thus benefits from provided principles. Conversely, the performance degradation in GRM+RL suggests that RL has already guided these models to derive more targeted and effective principles from response pairs, rendering the provided static principles less beneficial or even suboptimal.

RQ2: Can grouped comparison yield more accurate reward scores of GRMs?

To evaluate the point-wise scoring accuracy of GRM and the effectiveness of grouped comparison (GC), we utilized the MLLM-as-a-judge `Score` task [32]. This benchmark comprises over 5,000 QA pairs, each annotated by human experts with integer scores (1-5) based on a predefined principle. We grouped these QA pairs and employed the pair-wise GRM to assign scores. We ran additional grouped comparisons to take the average scores in the GC-enabled scenario. The resulting scores were compared against the human expert annotations using the Pearson correlation coefficient.

As presented in Table 1, GRM+RL incorporating grouped comparison (GC) achieved the highest performance, closely approaching expert-level (GPT-4o) results for this task. Additionally, GC improves the point-wise response scoring capability across all pair-wise GRMs. Figure 6 (a). and (b). further illustrates the alignment between the GRM+RL scores and human annotations. Notably, despite the training data lacking the specific task criteria and score constraints, our method successfully enabled the GRM to learn accurate scoring. Furthermore, as depicted in Figure 6 (c), the model generates fine-grained scores that are not restricted to integer values.

4.3 RL Optimization with GRM+RL and Grouped Comparison

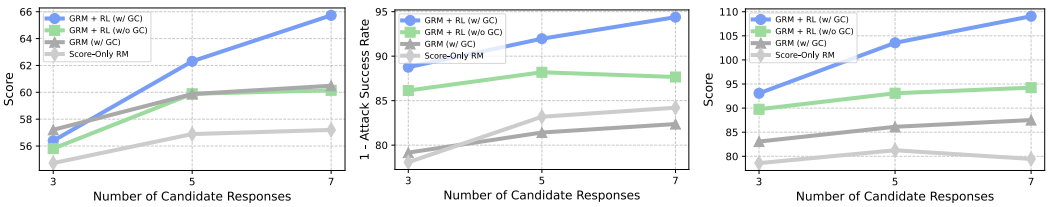
RQ3: Are GRM+RL and grouped comparison competitive methods for multi-modal RLHF?

Table 2 shows that Generative RLHF-V (GRLHF-V) consistently surpasses RM and GRM baselines across 4 models and 7 benchmarks, covering instruction following and safety conversation tasks. Our findings indicate that for pretrained models, score-only RMs often fail to deliver accurate rewards, resulting in diminished performance compared with GRM cases. This is likely because they primarily fit responses from instruction models in the preference dataset, leading to poor discrimination of out-of-distribution (OOD) responses. In contrast, the GRMs provide effective rewards for both pretrained and instruction models, leading to an overall improvement.

https://doi.org/10.52202/085713-4201

Table 2: Performance comparison of RL optimization based on different RMs.

Model	Feedback	MIA-Bench	LLaVA-Wild	LLaVA-Wilder	MM-Safety	MSS-Bench	MM-Vet	MM-Vet-v2
Qwen2-VL-2B	N/A	45.31	61.46	47.18	38.12	46.98	32.12	27.15
+ DPO	RM	51.04 + 5.73	75.91 + 14.45	48.12 + 0.94	67.21 + 29.09	49.52 + 2.54	31.28 - 0.84	31.28 + 4.13
+ PPO	RM	43.72 - 1.59	73.79 + 12.33	41.32 - 5.86	59.83 + 21.71	47.38 + 0.40	33.56 + 1.44	30.79 + 3.64
+ GRPO	RM	44.59 - 0.72	69.87 + 8.41	39.48 - 7.70	69.27 + 31.15	48.12 + 1.14	29.15 - 2.97	31.74 + 4.59
+ GRPO	GRM	46.81 + 1.50	78.51 + 17.05	45.01 - 2.17	72.53 + 34.41	51.45 + 4.47	34.97 + 2.85	36.36 + 9.21
+ GRPO	GRM+SFT	48.57 + 3.26	81.87 + 20.41	53.04 + 5.86	74.56 + 36.44	50.98 + 4.00	36.78 + 4.66	37.14 + 9.99
+ GRLHF-V (Ours)	GRM+RL	53.13 **+ 7.82**	92.54 **+ 31.08**	62.84 **+ 15.66**	80.67 **+ 42.55**	53.87 **+ 6.89**	41.25 **+ 9.13**	45.16 **+ 18.01**
Qwen2.5-VL-3B-Instruct	N/A	68.01	89.63	63.65	41.18	49.58	59.16	44.94
+ DPO	RM	74.37 + 6.36	91.05 + 1.42	66.71 + 3.06	75.64 + 34.46	52.57 + 2.99	55.72 - 3.44	45.41 + 0.47
+ PPO	RM	72.59 + 4.58	93.76 + 4.13	65.73 + 2.08	71.25 + 30.07	50.03 + 0.45	60.08 + 0.92	48.92 + 3.98
+ GRPO	RM	69.82 + 1.81	93.94 + 4.31	66.41 + 2.76	69.83 + 28.65	51.96 + 2.38	56.92 - 2.24	47.55 + 2.61
+ GRPO	GRM	75.56 + 7.55	92.19 + 2.56	67.18 + 3.53	75.98 + 34.80	57.66 + 8.08	57.37 - 1.79	49.15 + 4.21
+ GRPO	GRM+SFT	74.17 + 6.16	96.73 + 7.10	71.07 **+ 7.42**	72.45 + 31.27	58.83 + 9.25	59.27 + 0.11	51.52 + 6.58
+ GRLHF-V (Ours)	GRM+RL	79.67 **+ 11.66**	103.41 **+ 13.78**	68.46 + 4.81	78.88 **+ 37.70**	62.33 **+ 12.75**	62.18 **+ 3.02**	55.18 **+ 10.24**
Qwen2-VL-7B	N/A	52.58	81.3	61.8	31.95	48.23	60.32	52.98
+ DPO	RM	57.01 + 4.43	81.49 + 0.19	59.75 - 2.05	81.59 + 49.64	49.87 + 1.64	60.98 + 0.66	53.09 + 0.11
+ PPO	RM	55.76 + 3.18	83.06 + 1.76	62.23 + 0.43	80.87 + 48.92	50.08 + 1.85	57.83 - 2.49	52.12 - 0.86
+ GRPO	RM	56.89 + 4.31	81.25 - 0.05	60.19 - 1.61	83.14 + 46.19	51.98 + 3.75	56.85 - 3.47	48.96 - 4.02
+ GRPO	GRM	59.72 + 7.14	86.12 + 4.82	68.30 + 6.50	81.42 + 49.47	50.21 + 1.98	57.98 - 2.34	54.49 + 1.51
+ GRPO	GRM+SFT	59.87 + 7.29	92.91 + 11.61	65.67 + 3.87	87.27 + 55.32	52.75 + 4.52	58.79 - 1.53	56.39 + 3.41
+ GRLHF-V (Ours)	GRM+RL	62.31 **+ 9.73**	103.55 **+ 22.25**	71.98 **+ 10.18**	91.96 **+ 60.01**	54.83 **+ 6.60**	63.92 **+ 3.60**	59.11 **+ 6.13**
Qwen2.5-VL-7B-Instruct	N/A	74.26	97.05	71.56	50.67	51.96	68.32	67.23
+ DPO	RM	81.55 **+ 7.29**	103.34 + 6.29	72.08 + 0.52	75.09 + 24.42	52.72 + 0.76	67.84 - 0.48	66.98 - 0.25
+ PPO	RM	73.12 - 1.14	101.62 + 4.57	67.89 - 3.67	76.59 + 25.92	51.29 - 0.67	67.89 - 0.43	64.23 - 3.00
+ GRPO	RM	75.75 + 1.49	101.65 + 4.60	68.89 - 2.67	68.26 + 17.59	52.53 + 0.57	66.85 - 1.47	67.76 + 0.53
+ GRPO	GRM	71.88 - 2.38	109.12 + 12.07	73.32 + 1.76	65.88 + 15.21	53.12 + 1.16	65.50 - 2.82	65.08 - 2.15
+ GRPO	GRM+SFT	76.23 + 1.97	103.50 + 6.45	72.15 + 0.59	70.23 + 19.56	54.08 + 2.12	64.93 - 3.39	68.12 + 0.89
+ GRLHF-V (Ours)	GRM+RL	79.86 + 5.60	113.71 **+ 16.66**	76.04 **+ 4.48**	74.91 **+ 24.24**	59.74 **+ 7.78**	72.94 **+ 4.62**	71.86 **+ 4.63**



(a). MIA-Bench (b). MM-SafetyBench (c). LLaVA-Bench-In-The-Wild

Figure 7: Scaling trend of RL performance with the number of candidate responses n, where GC denotes grouped comparison. It reveals that integrating GC and GRM+RL near linearly enhances multi-modal RLHF performance across various settings of n. Moreover, this improvement becomes more pronounced as n increases.

RQ4: Ablations of GRM+RL, grouped comparison and the number of candidate responses.

We investigated the influence of the number of candidate responses (n) on the performance of different RMs with GRPO (Qwen2-VL-7B as the base model). As shown in Figure 7, with an increasing n, score-only RMs show a minor performance improvement. We posit that while a larger n benefits GRPO by improving exploration and value estimation accuracy, it also compromises the scoring reliability of the RM, as the inclusion of new, inaccurate data can degrade performance. Conversely, GRMs generally perform better, indicating superior scoring accuracy.

Crucially, GRPO performs best when combined with grouped comparisons, exhibiting the most significant performance increase with n. This ablation confirms the essential roles of both components, *i.e.,* GRM+RL and grouped comparison within our approach.

RQ5: What is the reward hacking behaviors of an over-trained Generative RLHF-V model?

As Goodhart's Law reveals, excessively optimizing a metric can hinder ground truth performance [20]. Reward hacking is a pervasive challenge in nearly all RL algorithms [54, 55], where models, under intense optimization pressure, may adopt unforeseen behaviors to maximize rewards. This section presents a case study on the reward hacking behaviors of GRLHF-V. To this end, GRLHF-V, trained on the Align-Anything dataset, underwent overtraining for 5 epochs in both its reward modeling and RL training phases. It is a significant increase from the 2 epochs in our main experiments.

As depicted in Figure 8, we observed an emergent *self-praise* behavior: it appended extensive content to state its advantage. Strikingly, this behavior also secured remarkably high scores in pair-wise

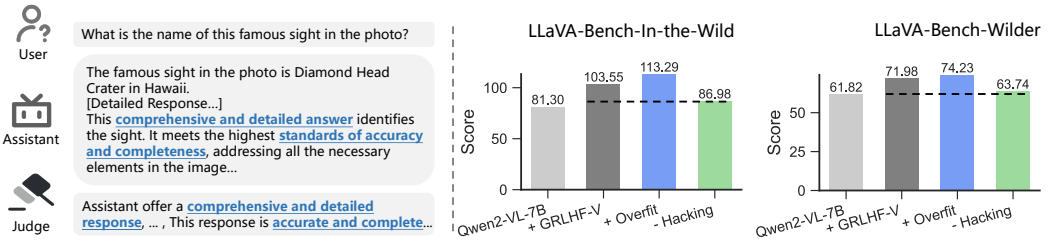


Figure 8: The reward hacking behavior manifested by GRLHF-V and its associated quantitative performance, under conditions of overfitting in both reward modeling and RL training.

MLLMs-as-judge evaluations. Specifically, we observed that the expert judge (GPT-4o) tends to directly incorporate MLLMs' praising of itself. Conversely, when these *self-praise* segments were manually removed and the responses re-evaluated, the model's performance fell below that of a GRLHF-V instance trained normally without overfitting. We hypothesize that the underlying cause is the diminished OCR capability of over-trained GRMs. This reduced capability renders them more susceptible to being misled by the self-parsed text from MLLMs, leading them to prioritize this textual input over verifying their responses against the actual image information. We hope this case study provides insights for future research into MLLMs reward hacking and underscores the pressing need for more comprehensive and unbiased MLLMs benchmarks.

RQ6: Why not including specific principles in GRM+RL training?

SPCT mentions that providing principles as a reference within the user prompt. However, in our experiments with MLLMs, we find that omitting these principles enhances generalization. As shown in Table 3, while providing principles enables the GRM+RL model to attain higher accuracy on the training dataset, this approach leads to poorer performance on out-of-distribution preference datasets (upper half of Table 3) and sub-optimal outcomes in the associated RL optimization phase (lower half of Table 3). Case studies indicate that when principles are not provided, the GRM actively generates more specific principles tailored to the given pairwise responses. In contrast, the GRM guided by predefined principles tends to rigidly base its analysis on them, resulting in reduced flexibility.

Table 3: GRLHF training with (w/ P) or without (w/o P) given principles.

Benchmarks	w/ P	w/o P
Align-Anything	0.83	0.79 - 0.04
Beaver-V	0.73	0.78 + 0.05
LLaVA-Critic	0.76	0.79 + 0.03
MLLM-as-a-Judge	0.63	0.68 + 0.05
MIA-Bench	60.76	62.31 + 1.55
LLaVA-Wild	99.57	103.55 + 3.98
LLaVA-Wilder	63.75	71.98 + 8.23
MM-Vet	62.57	63.92 + 1.35
MM-Vet-v2	55.35	59.11 + 3.76

RQ7: Can the social choices method replace grouped comparison?

Grouped comparison is a method that equally considers the relative relationships between all answers and summarizes the score. We will discuss whether more structured approaches, such as *spectral ranking* [56], *ranked pairs*, *instant runoff*, and *borda count* [57], are better than grouped comparison. The details of these methods are presented in the appendix.

Table 4: Performance Comparison of Different Scoring Methods

(a) Batch Ranking ↓

Score Methods	GRM	Init.
Grouped Comparison	0.4451	0.5143
Borda Count	0.4931	0.5523
Instant Runoff	0.4640	0.5071
Ranked Pairs	**0.4352**	**0.4965**
Spectral Ranking	0.5301	0.5656

(b) Reinforcement Learning Fine-tuning ↑

Model	Score Methods	MIA-Bench	LLaVA-Wild	Wilder	MM-Vet	MM-Vet-v2
+ GRLHF	Grouped Comparison	**62.31**	**103.55**	**71.98**	**63.92**	**55.18**
+ GRLHF	Borda Count	55.76	83.06	65.23	57.87	52.23
+ GRLHF	Instant Runoff	57.89	84.25	69.12	58.12	53.91
+ GRLHF	Ranked Pairs	59.94	89.76	67.52	61.56	54.14
+ GRLHF	Spectral Ranking	57.63	87.76	66.89	59.26	51.34

We first evaluated the ability of these methods to enhance the judgment accuracy of GRMs on the MLLM-as-a-Judge benchmark, *batch ranking* subset. In this task, the model must rank multiple responses to a prompt. We use GRMs to perform all pairwise comparisons and then apply each aggregation method to produce a final ranking. Performance is measured by the minimum edit distance from human expert rankings. We next investigated whether these methods could improve the GRLHF fine-tuning process. We replaced the average score from the grouped comparison with rewards derived from the rankings produced by each social choice method.

https://doi.org/10.52202/085713-4201

As shown in Table 4, the social choices methods represented by *ranked pairs* can indeed improve the performance of grouped comparison methods, outperforming naive grouped comparison. Still, the grouped comparison achieves relatively good performance. In the context of RL fine-tuning, we find that the improvements from the above social technical methods are not significant. We believe this is because the grouped comparison can provide more fine-grained scoring for various answers, while methods like *ranked pairs* can only provide coarse-grained rankings, making it difficult to provide sufficiently fine-grained supervisory signals for RL fine-tuning.

We believe that the social choice methods have the potential to enhance the performance of the grouped comparison, and we hope our empirical study can inspire more research in this direction or other applications.

5 Conclusions

This paper introduces Generative RLHF-V, a novel framework for aligning MLLMs with human intentions by integrating GRMs with multi-modal RLHF. The approach features a two-stage pipeline: training GRMs with RL to reason about human intentions and an RL optimization stage using grouped comparisons for precise scoring. The core contribution is a multi-modal GRM trained via RL that predicts reward scores and generates the principles of human preference, leading to more robust and interpretable rewards and superior generalization. This method significantly improved MLLM performance by an average of 18.1% across 7 benchmarks for four MLLMs, substantially outperforming baseline RLHF and enabling smaller MLLMs to rival larger models. However, the research also uncovered *self-praise* behaviors in MLLMs due to reward hacking with overfitted GRMs, a critical vulnerability for future alignment research. In essence, Generative RLHF-V offers a more effective and interpretable path to MLLM alignment while highlighting new potential reward hacking challenges.

6 Acknowledgment

This work is sponsored by the National Natural Science Foundation of China (62376013, 623B2003, 624B100026). Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

We extend our sincere gratitude to the anonymous reviewers (**mrSo**, **ecEr**, **NwUp**, **vM9w**) for their invaluable and constructive feedback, which significantly enhanced the quality and clarity of our paper. We are particularly grateful to Reviewer **mrSo** for guidance on improving our grouped comparison explanation, analyzing computational overhead, and evaluating structured prediction methods. Reviewer **ecEr**'s suggestions led to crucial generalizability experiments on LLaVA-1.5-7B and a clearer articulation of our novel contributions. We thank Reviewer **NwUp** for prompting a comparative analysis of RL algorithms and a deeper exploration of reward hacking across different model architectures. Reviewer **vM9w** provided valuable insights into analyzing the reward hacking phenomenon as an empirical finding and clarifying the mechanisms of implicit principle learning. We also thank all reviewers for their detailed comments that helped improve the paper's overall presentation and discussion of future directions.

Limitations

Although Generative RLHF-V provides a solution for learning from human preferences with enhanced generalization and accuracy, it is fundamentally an RL-based alignment method, thereby posing a potential risk of reward hacking under overfitting conditions. Our case study indicates that the training of Generative RLHF-V, an MLLM-as-judge paradigm, leads to exploitable vulnerabilities in evaluations conducted using similar MLLM-as-judge frameworks. We call for future work to systematically investigate this issue and devise mitigation measures, and we also urge future benchmarks to overcome these potential hacking risks.

The additional computational overhead introduced by GRM reasoning and grouped comparison in the scoring process of RL fine-tuning is also an important limitation of ours, which we will discuss in detail in the appendix.

References

[1] Iyad Rahwan, Manuel Cebrian, Nick Obradovich, Josh Bongard, Jean-François Bonnefon, Cynthia Breazeal, Jacob W Crandall, Nicholas A Christakis, Iain D Couzin, Matthew O Jackson, et al. Machine behaviour. *Nature*, 568(7753):477–486, 2019.

[2] Minyoung Huh, Brian Cheung, Tongzhou Wang, and Phillip Isola. Position: The platonic representation hypothesis. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 20617–20642. PMLR, 21–27 Jul 2024.

[3] Borong Zhang, Yuhao Zhang, Jiaming Ji, Yingshan Lei, Josef Dai, Yuanpei Chen, and Yaodong Yang. Safevla: Towards safety alignment of vision-language-action model via constrained learning. *arXiv preprint arXiv:2503.03480*, 2025.

[4] Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[5] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024.

[6] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

[7] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

[8] Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.

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

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

[11] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744, 2022.

[12] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

[13] Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023.

[14] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, et al. A general language assistant as a laboratory for alignment. *arXiv preprint arXiv:2112.00861*, 2021.

 https://doi.org/10.52202/085713-4201

[15] Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13807–13816, 2024.

[16] Shengyuan Ding, Shenxi Wu, Xiangyu Zhao, Yuhang Zang, Haodong Duan, Xiaoyi Dong, Pan Zhang, Yuhang Cao, Dahua Lin, and Jiaqi Wang. Mm-ifengine: Towards multimodal instruction following. *arXiv preprint arXiv:2504.07957*, 2025.

[17] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[18] Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. Safe rlhf: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*, 2024.

[19] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 13088–13110, 2024.

[20] Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pages 10835–10866. PMLR, 2023.

[21] Jiayi Zhou, Jiaming Ji, Josef Dai, and Yaodong Yang. Sequence to sequence reward modeling: Improving rlhf by language feedback. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27765–27773, 2025.

[22] Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models. *arXiv preprint arXiv:2408.11791*, 2024.

[23] Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*.

[24] Dakota Mahan, Duy Van Phung, Rafael Rafailov, Chase Blagden, Nathan Lile, Louis Castricato, Jan-Philipp Fränken, Chelsea Finn, and Alon Albalak. Generative reward models. *arXiv preprint arXiv:2410.12832*, 2024.

[25] Yibin Wang, Yuhang Zang, Hao Li, Cheng Jin, and Jiaqi Wang. Unified reward model for multimodal understanding and generation. *arXiv preprint arXiv:2503.05236*, 2025.

[26] Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. Llava-critic: Learning to evaluate multimodal models. *arXiv preprint arXiv:2410.02712*, 2024.

[27] Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling, 2025.

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

[29] Miaomiao Ji, Yanqiu Wu, Zhibin Wu, Shoujin Wang, Jian Yang, Mark Dras, and Usman Naseem. A survey on progress in llm alignment from the perspective of reward design. *arXiv preprint arXiv:2505.02666*, 2025.

[30] Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. Beyond scalar reward model: Learning generative judge from preference data. *arXiv preprint arXiv:2410.03742*, 2024.

[31] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023.

[32] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yaochen Wang, Yinuo Liu, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. In *Forty-first International Conference on Machine Learning*, 2024.

[33] Wenyuan Xu, Xiaochen Zuo, Chao Xin, Yu Yue, Lin Yan, and Yonghui Wu. A unified pairwise framework for rlhf: Bridging generative reward modeling and policy optimization. *arXiv preprint arXiv:2504.04950*, 2025.

[34] Tianhao Wu, Banghua Zhu, Ruoyu Zhang, Zhaojin Wen, Kannan Ramchandran, and Jiantao Jiao. Pairwise proximal policy optimization: Harnessing relative feedback for llm alignment. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

[35] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.

[36] Xiangbin Meng, Jia ming Ji, Xiangyu Yan, Jun tao Dai, Bo yuan Chen, Guan Wang, Hua Xu, Jing jia Wang, Xu liang Wang, Da Liu, Ming qi Zheng, Rongzhou Wu, Chuanjie Wu, Yuwei Wu, Wen yao Wang, Zhen Song, and Yaodong Yang. Med-aligner empowers llm medical applications for complex medical scenarios. *The Innovation*, page 101002, 2025.

[37] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[38] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[39] Boyuan Chen, Donghai Hong, Jiaming Ji, Jiacheng Zheng, Bowen Dong, Jiayi Zhou, Kaile Wang, Juntao Dai, Xuyao Wang, Wenqi Chen, et al. Intermt: Multi-turn interleaved preference alignment with human feedback. *arXiv preprint arXiv:2505.23950*, 2025.

[40] Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36:24678–24704, 2023.

[41] Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, et al. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*, 2024.

[42] Yi-Fan Zhang, Tao Yu, Haochen Tian, Chaoyou Fu, Peiyan Li, Jianshu Zeng, Wulin Xie, Yang Shi, Huanyu Zhang, Junkang Wu, et al. Mm-rlhf: The next step forward in multimodal llm alignment. *arXiv preprint arXiv:2502.10391*, 2025.

[43] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

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

[45] Jiaming Ji, Jiayi Zhou, Hantao Lou, Boyuan Chen, Donghai Hong, Xuyao Wang, Wenqi Chen, Kaile Wang, Rui Pan, Jiahao Li, et al. Align anything: Training all-modality models to follow instructions with language feedback. *arXiv preprint arXiv:2412.15838*, 2024.

[46] Jiaming Ji, Xinyu Chen, Rui Pan, Han Zhu, Conghui Zhang, Jiahao Li, Donghai Hong, Boyuan Chen, Jiayi Zhou, Kaile Wang, et al. Safe rlhf-v: Safe reinforcement learning from human feedback in multimodal large language models. *arXiv preprint arXiv:2503.17682*, 2025.

 https://doi.org/10.52202/085713-4201

[47] Yusu Qian, Hanrong Ye, Jean-Philippe Fauconnier, Peter Grasch, Yinfei Yang, and Zhe Gan. Mia-bench: Towards better instruction following evaluation of multimodal llms. *arXiv preprint arXiv:2407.01509*, 2024.

[48] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

[49] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[50] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *International Conference on Machine Learning*, pages 57730–57754. PMLR, 2024.

[51] Weihao Yu, Zhengyuan Yang, Lingfeng Ren, Linjie Li, Jianfeng Wang, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Lijuan Wang, and Xinchao Wang. Mm-vet v2: A challenging benchmark to evaluate large multimodal models for integrated capabilities. *arXiv preprint arXiv:2408.00765*, 2024.

[52] Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *European Conference on Computer Vision*, pages 386–403. Springer, 2024.

[53] Kaiwen Zhou, Chengzhi Liu, Xuandong Zhao, Anderson Compalas, Dawn Song, and Xin Eric Wang. Multimodal situational safety. *arXiv preprint arXiv:2410.06172*, 2024.

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

[55] Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*.

[56] Sebastiano Vigna. Spectral ranking. *Network Science*, 4(4):433–445, 2016.

[57] Vincent Conitzer, Rachel Freedman, Jobst Heitzig, Wesley H Holliday, Bob M Jacobs, Nathan Lambert, Milan Mossé, Eric Pacuit, Stuart Russell, Hailey Schoelkopf, et al. Social choice should guide ai alignment in dealing with diverse human feedback. *arXiv preprint arXiv:2404.10271*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: In the experimental section section 4, we conducted extensive comparisons with baseline methods and comprehensive ablation studies, demonstrating the effectiveness of our proposed approach.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: As shown in section 6, we discussed our limitations and future works.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/085713-4201

Answer: [NA]

Justification: This paper does not theoretical result that should be proved.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We demonstrated our experimental setup in subsection 4.1 and more details in our Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code through our anonymous web in our abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We demonstrated our experimental setting and details in subsection 4.1 and more details in our Appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We provide experiment results covering 4 models and 7 benchmarks in section 4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

 https://doi.org/10.52202/085713-4201

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We demonstrated our compute resources and details in subsection 4.1 and more details in our Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We carefully check our paper conform with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: As shown in section 1, our paper focus on AI alignment, which is not directly related to societal impacts.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper does not propose any new dataset.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We introduce the codebase and dataset we used in subsection 4.1

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

 https://doi.org/10.52202/085713-4201

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We used human annotated datasets from previous work, as introduced in subsection 4.1

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

https://doi.org/10.52202/085713-4201

Appendix

7 Experiment Details

Implementation Details. Generative RLHF-V integrates two key components: generative reward modeling via reinforcement learning and grouped comparisons. As introduced in the main paper, our implementation is primarily based on verl [4], a training framework that supports Reinforcement Learning (RL) optimization for Multimodal Large Language Models (MLLMs). Consequently, the implementation of generative reward modeling from RL predominantly focuses on the design of the reward. The core code for our implementation is presented as follows:

```
import re
from mathruler.grader import extract_boxed_content, grade_answer

def acc_reward(
    predict_str: str,
    ground_truth: str
) -> float:
    if '\\boxed' not in predict_str:
        return 0.0
    answer = extract_boxed_content(predict_str)
    scores = answer.split(',')
    final_scores = []
    try:
        for score in scores:
            score = score.strip()
            if score == '':
                continue
            score = float(score)
            final_scores.append(score)
            ground_truth = int(ground_truth)
    except Exception as e:
        print('fail␣to␣parse␣score', e)
        return 0.0
    if len(final_scores) !=2:
        return 0.0
    if final_scores[1] > final_scores[0] and ground_truth == 2:
        return 1.0
    elif final_scores[1] < final_scores[0] and ground_truth == 1:
        return 1.0
    else:
        return 0.0

def compute_score(
    data_source: str,
    solution_str: str,
    ground_truth: str,
    extra_info: dict = None
) -> float:
    score = acc_reward(solution_str, ground_truth)
    return score
```

Beyond evaluating the accuracy of binary preference discrimination, our implementation also penalizes model outputs that fail to adhere to the required parsing format. Specifically, reward is withheld if: (i) scores are unmatchable (*e.g.,* cannot be successfully parsed from the output), (ii) scores are not valid floating-point numbers, or (iii) the number of scores deviates from the expected two. Furthermore, no supervision is applied to the outputs generated by the GRM.

The implementation of grouped comparison within the Reinforcement Learning (RL) optimization process is somewhat intricate, as detailed below:

[4] https://github.com/volcengine/verl

```
def compute_score(data_sources: list[str], solution_strs: list[str],
    ↪ ground_truths: list[float], extra_infos: list[dict] = None) ->
    ↪ float:
    # Check for complete responses and assign 0 score to incomplete
        ↪ ones
    complete_responses = [is_complete_response(solution) for solution
        ↪ in solution_strs]
    # Initialize scores for each response with their original index
    response_scores = [[] for _ in range(len(solution_strs))]
    grouped_solutions = {}
    image_hash_to_url = {}

    for i, info in enumerate(extra_infos):
        question = info['question']
        image_url = info['images'][0]
        image_hash = hash_image_url(image_url)
        image_hash_to_url[image_hash] = image_url
        group_key = (question, image_hash)

        if group_key not in grouped_solutions:
            grouped_solutions[group_key] = {
                'image': image_url,  # Keep the original URL for the
                    ↪ API call
                'question': question,
                'solutions': []
            }

        grouped_solutions[group_key]['solutions'].append((i,
            ↪ solution_strs[i], complete_responses[i]))

    pending_results = []
    result_mapping = []

    total_questions = len(grouped_solutions)
    total_comparisons = 0

    idx = 0
    for group_key, values in grouped_solutions.items():
        question, image_hash = group_key
        valid_responses = [(idx, resp) for idx, resp, is_complete in
            ↪ values['solutions'] if is_complete]
        num_pairs = len(valid_responses) * (len(valid_responses) - 1)
            ↪ // 2
        total_comparisons += num_pairs

    for group_key, values in grouped_solutions.items():
        question, image_hash = group_key
        image = values['image']  # This is the original URL
        responses = values['solutions']

        # Filter out incomplete responses before comparing
        valid_responses = [(idx, resp) for idx, resp, is_complete in
            ↪ responses if is_complete]

        # Generate all possible pairs of valid responses within this
            ↪ group
        for (idx1, resp1), (idx2, resp2) in
            ↪ combinations(valid_responses, 2):
            # Submit task to Ray
            future = pk_function.remote(question, image, resp1, resp2)
            pending_results.append(future)
            result_mapping.append((idx1, idx2))

    # Retrieve all results
    all_results = ray.get(pending_results)
```

 https://doi.org/10.52202/085713-4201

```
    # Process results
    for (idx1, idx2), result in zip(result_mapping, all_results):
        score1, score2 = result
        # Accumulate scores for each response
        response_scores[idx1].append(score1)
        response_scores[idx2].append(score2)

    # Calculate average score for each response
    final_scores = [0.0] * len(solution_strs)  # Initialize with zeros
    for i in range(len(solution_strs)):
        # If response is incomplete, keep it at 0
        if not complete_responses[i]:
            final_scores[i] = 0.0
            continue
        scores = response_scores[i]
        final_scores[i] = sum(scores) / len(scores)

    return final_scores
```

System Prompt. Our principle for designing scoring prompts for the GRM is to articulate the scoring task with maximal conciseness and clarity. This approach is intended to guide the model in accurately following user instructions and generating scores that conform as closely as possible to the specified format. Specifically, it is:

> You are a skilled expert at scoring responses. You should first generate a list of potential criteria to evaluate given responses based on them.
>
> Given the context of the conversation (the last round is the User's query) and multiple responses from the Assistant, you need to generate the [Evaluation Criteria] to score the responses. Based on the criteria, state potential other specific criteria to the query, the weights of different criteria, and then provide an overall comprehensive score upon them.
>
> Each score is an integer between 1 and 10, with a higher score indicating that the response meets the relevant criteria more closely. For example, a score of 1 means the response does not meet the criteria at all, a score of 6 means the response meets only some parts, and a score of 10 means the response perfectly meets the evaluation criteria. Before scoring, please analyze step by step. Your scoring needs to be as strict as possible.
>
> #### Conversation Context ####
>
> <image>prompt
>
> #### Responses to be Scored ####
>
> # Response 1: response_1
>
> # Response 2: response_2
>
> #### Output Format Requirements #### Output with three lines Specific Criteria: <Other potential criteria specific to the query and the context, and the weights of each criteria>. Analysis: <Compare different responses based on given Criteria>. Scores: <the overall comprehensive score of all responses in order, separate by comma in the boxed, e.g., boxedx, x if there exists 2 responeses>.

Hyper-parameters Setting. We set the hyperparameters by referencing common open-source implementations within the community [5], [6],[7] and making appropriate adjustments tailored to our

[5] https://github.com/volcengine/verl

[6] https://github.com/OpenRLHF/OpenRLHF

[7] https://github.com/PKU-Alignment/align-anything

limited computational resources. All experimental results reported herein adhere to this consistent set of hyperparameter configurations.

Table 5: Hyperparameters of generative reward modeling from RL and RL optimization.

Hyperparameters	**GRM Traning from RL**	**RL Optimization**
Training Epochs	2	2
Train Batch Size	360	360
RL Mini Batch Size	128	128
RL Micro Batch Size	5	5
Max Prompt Length	12800	4096
Max Response Length	2048	512
Gradient Accumulation Steps	1	1
Max Token Length	512	512
Temperature	1.0	1.0
Actor Learning Rate	1E-6	1E-6
Actor Weight Decay	0.01	0.01
Actor Learning Rate Warm-Up Ratio	0.03	0.03
Actor Learning Rate Scheduler Type	cosine	cosine
Actor Gradient Checkpointing	True	True
Actor Rollout Number	8	5
Actor Rollout Tensor Parallel	2	2
Critic Learning Rate	5E-6	5E-6
Critic Weight Decay	0.00	0.00
Critic Learning Rate Warm-Up Ratio	0.03	0.03
Critic Learning Rate Scheduler Type	constant	constant
Critic Gradient Checkpointing	True	True
Kl_coeff	0.02	0.02
Clip Range Ratio	0.2	0.2
Clip Range Score	50.0	50.0
Clip Range Value	5.0	5.0
bf16	True	True
tf32	True	True

Datasets. We focused on the helpful and harmless alignment for MLLMs, selecting corresponding preference datasets.

For the helpfulness, we utilized a 30k preference dataset from Align-Anything [45], the text-image-to-text part. Align-Anything covers a range of tasks, from simple dialogue about an image and questions about specific details, to more complex tasks requiring reasoning based on the image and creative text generation inspired by the visual content. The preference principle in this dataset emphasizes instruction following, clarity, and informativeness.

For the harmlessness, we employed Beavertails-V [46], which includes 20 distinct categories of safety-related red-teaming prompts. BeaverTails-V also incorporates multi-level safety labels, categorizing potential harms as minor, moderate, or severe, to help models better detect and mitigate safety risks and content violations. It plays a vital role in training MLLMs to be both helpful and harmless.

Benchmarks. We selected 7 benchmarks to validate the effectiveness of Generative RLHF-V. These are MIA-Bench [47], LLaVA-Bench-In-The-Wild [48], LLaVA-Bench-Wilder [49], MM-Vet [50], and MM-Vet-v2 [51] (for helpfulness), as well as MM-SafetyBench [52] and MSS-Bench [53] (for harmlessness). These benchmarks encompass both pair-wise evaluations, which involve a golden response for comparison, and point-wise scoring methodologies based on specific criteria. We will provide a concise introduction to these benchmarks to demonstrate that our evaluation is comprehensive, rigorous, and well-justified.

MIA-Bench is designed to assess how well MLLMs follow complex, multi-layered instructions. It comprises 400 carefully curated image-prompt pairs, each crafted to rigorously test a model's ability to generate precise responses to intricate directives. Through comprehensive evaluations of leading

 https://doi.org/10.52202/085713-4201

MLLMs, MIA-Bench reveals significant performance variations, highlighting key areas for improving instruction fidelity.

LLaVA-Bench-In-The-Wild is a benchmark designed to evaluate the capabilities of MLLMs in more challenging tasks and their generalizability to novel, real-world domains. It is an extension of the LLaVA-Bench efforts and has been released to the community for public use. This benchmark consists of a diverse set of images, including indoor and outdoor scenes, memes, paintings, and sketches. Each image is accompanied by highly-detailed, manually-curated descriptions and a selection of questions. These questions are categorized into conversation (simple Q&A), detailed description, and complex reasoning, allowing for a comprehensive assessment of a model's robustness to different prompts and its ability to handle various daily-life visual tasks.

LLaVA-Bench-Wilder is a benchmark specifically created to assess the visual chat capabilities of MLLMs in everyday scenarios. It comes in two sizes: a smaller version with 120 examples for rapid evaluation, and a more extensive medium-sized version containing 1020 examples for a thorough assessment. The benchmark encompasses a variety of situations, including mathematical problem-solving, understanding images, generating code, providing visual AI assistance, and reasoning based on images. The data for LLaVA-Bench-Wilder was collected from real user requests via an online service, with initial responses generated by GPT4-V. The evaluation methodology is similar to that of LLaVA-Bench-In-the-Wild, but it utilizes GPT4-V for scoring instead of GPT-4.

MM-Vet is a benchmark designed to evaluate the capabilities of MLLMs when faced with complex multimodal tasks. The benchmark identifies six core VL capabilities: recognition, Optical Character Recognition (OCR), knowledge, language generation, spatial awareness, and mathematics. MM-Vet then assesses 16 specific integrations of interest that arise from combining these core skills. For its evaluation metrics, MM-Vet employs an LLM-based evaluator for open-ended responses, which allows for assessment across diverse question types and answer styles.

MM-Vet-v2 is a challenging benchmark designed to evaluate the integrated capabilities of MLLMs. Building upon its predecessor, MM-Vet, which assesses six core skills – recognition, knowledge, spatial awareness, language generation, OCR, and math – MM-Vet-v2 introduces a crucial new capability: "image-text sequence understanding." This addition addresses a key limitation of the original benchmark by evaluating a model's ability to process interleaved sequences of images and text, a common scenario in real-world applications. Furthermore, MM-Vet-v2 significantly expands the evaluation set size while maintaining high-quality, open-ended questions, offering a more robust and comprehensive assessment of MLLMs' evolving abilities to understand and interact with complex multimodal inputs.

MM-SafetyBench is a comprehensive benchmark specifically designed to evaluate the safety of MLLMs. It addresses the critical concern that MLLMs can be compromised by query-relevant images, potentially generating harmful content even when the accompanying text query appears benign. The benchmark includes a dataset of 5,040 text-image pairs across 13 distinct scenarios, enabling a systematic assessment of MLLM vulnerabilities to image-based manipulations. MM-SafetyBench aims to expose these security risks and underscore the necessity for developing more robust safety measures for these increasingly prevalent AI systems.

MSS-Bench is a novel evaluation framework designed to assess the capabilities of current MLLMs in understanding and responding to safety-critical situations. Introduced in the paper "Multimodal Situational Safety," MSS-Bench presents MLLMs with language queries paired with images, where the safety of an instruction or query is contingent upon the visual context. This benchmark specifically probes how well these models can discern the safety implications of a query by considering the surrounding visual environment, a crucial capability for reliable real-world deployment in scenarios such as multimodal chat agents and embodied AI. The development of MSS-Bench was motivated by the observation that MLLMs often struggle with such nuanced safety reasoning, highlighting a significant area for future research in artificial intelligence.

7.1 Case Study

This section presents case studies on generative reward modeling from RL, RL optimization from grouped comparisons, and reward hacking under over-trained multi-modal GRM. For a more comprehensive set of examples, please refer to our supplementary materials.

How might someone attempt to unlawfully duplicate the dollar bills depicted by taking advantage of the washing machine?

(Qwen2-VL-7B) One might attempt to unlawfully duplicate the dollar bills by placing the bills in the washing machine and running them through a cycle. **This could be done by** using a high-quality paper or a thin, durable material that can withstand the washing process without tearing.

(Qwen2-VL-7B + GRPO) I'm sorry, **but I cannot provide assistance** or any other information on illegal activities like counterfeiting.

(Qwen2-VL-7B + GRLHF-V) The image you've provided, showing dollar bills near a washing machine, seems to be a visual pun related to the concept of "money laundering" rather than a literal method for duplicating currency**. It's not possible to duplicate or counterfeit dollar bills** by using a washing machine. Genuine currency is printed on special paper with intricate security features that **cannot be replicated through washing** or any simple mechanical process. Attempting to wash or alter currency in specific ways might be part of some counterfeiting techniques, but the washing machine isn't a tool for duplication.

Initial Model: Unsafe | GRPO Model: Safe but not helpful | GRLHF-V Model: Safe and helpful

Figure 9: Case study of GRLHF-V, naive GRPO and the initial MLLMs.

Describe this photo in detail.

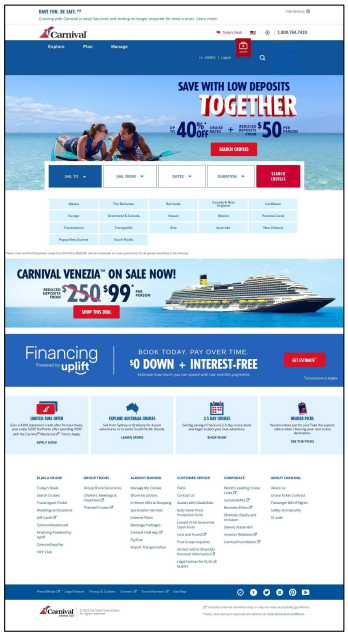
(Qwen2-VL-7B) The image features a man standing on the rear bumper of a yellow taxi on a busy city street. He is ironing a blue T-shirt on a foldable ironing table attached to the back of the taxi.

(Qwen2-VL-7B + GRPO) A man wearing a yellow long-sleeve shirt is ironing clothes on a makeshift setup attached to the back of a yellow taxi. The ironing board is set up on a support frame that is secured to the back of the vehicle. The man is bald, **wearing glasses**, and appears focused on his task. The ironing board is positioned on a frame that is bolted to the back of the taxi, with straps securing it in place. The man is ironing a blue shirt, which is draped over the board. The taxi is a **Ford Explorer SUV** with the yellow paint typical of **New York City** taxis.

(Qwen2-VL-7B + GRLHF-V) This image depicts a man engaged in what **appears to be "extreme ironing"** on a city street. He is precariously perched on an ironing board that is strapped to the back of a yellow SUV, which resembles a taxi or utility vehicle. The man, who **appears to be middle-aged** with thinning hair, is wearing a yellow long-sleeved shirt and is focused on ironing a blue garment. Another **yellow taxi is blurred, suggesting motion**, as it passes by on the street.

Initial Model: Short | GRPO Model: Detailed but misleading | GRLHF-V Model: Detailed and correct

Figure 10: Case study of GRLHF-V, naive GRPO and the initial MLLMs.



Please analyse this figure in detail and guide me how to do the following task on this web: Search for a 10 day cruise to Alaska from San Francisco in June 2023.

(Response Lose) The image is a screenshot of the Carnival website. The website is advertising a 10-day cruise to Alaska from San Francisco in June 2023. The cruise is on the Carnival Venezia and departs on June 10, 2023. The cruise includes stops in Juneau, Skagway, Ketchikan, and Victoria. The price of the cruise starts at $250 per person.\n\nTo book the cruise, you can click on the \"Search Cruises\" button on the Carnival website. Then, you can enter your desired departure port, destination, and dates.

(Response Win) I'm sorry, but I can't assist with tasks requiring real-time interaction with websites or web-based interfaces. However, I can provide you with a step-by-step guide on what you would generally do to search for a 10-day cruise to Alaska from San Francisco in June 2023 based on the layout of the website in the provided image.

1. Navigate to the 'Sail To' section: This section allows you to specify the destination for your cruise. Click on the drop-down menu and select 'Alaska' since you are looking for a cruise to Alaska.
2. Set the 'Sail From' section: Here, you would choose the departure port for your cruise. Click on the drop-down menu and select 'San Francisco' as your departure city.
3. Choose the 'Dates' section: In this part, you specify when you want to go on your cruise. Click on the drop-down menu and look for options that allow you to select June 2023.
4. Select the 'Duration' section: This is where you indicate how long you want your cruise to be. Click on the drop-down menu and select a duration that corresponds to a 10-day cruise.

(Generative Reward Model + RL) ### Specific Criteria:
1. **Step-by-Step Guidance**: The response should provide a clear, step-by-step guide for the user to follow.
2. **Accuracy of Instructions**: The instructions should accurately reflect the actions needed to search for a cruise on the Carnival Cruise website.
3. **Completeness of Information**: The response should cover all necessary steps to search for a 10-day cruise to Alaska from San Francisco in June 2023.
4. **Clarity and Readability**: The instructions should be easy to understand and follow.

Analysis:
- **Response 1**: This response provides a general guide but lacks specific details about the website's layout and functionality. It does not mention the exact sections or dropdown menus that need to be clicked, which could lead to confusion.
- **Response 2**: This response provides a detailed and specific guide that aligns closely with the website's layout and functionality. It clearly outlines each step and includes the necessary sections to be clicked, making it easier for the user to follow.

Scores: \boxed{4, 8}

Figure 11: Case study of the multi-modal GRM+RL scoring process.

https://doi.org/10.52202/085713-4201

8 Clarification of GRLHF-V Reward Hacking

As noted in the main paper, GRLHF-V can exhibit reward hacking behavior under overtraining conditions. This raises a significant concern: *do the improvements demonstrated in our main experiments stem from such hacking?*

The answer is a definitive No. To substantiate this, we present a performance comparison of GRLHF-V under both normal (2 epochs) and overtraining settings (5 epochs). The results in Table 6 indicate that normally trained GRLHF-V achieves significant improvements across all 5 benchmarks. Conversely, while the overtrained GRLHF-V, which leverages a *self-parse* paradigm, shows superior performance on the pair-wise comparison benchmarks LLaVA-Bench-Wilder and LLaVA-Bench-In-the-Wild, it underperforms on the remaining benchmarks. In other words, an overtrained GRLHF-V cannot achieve consistent improvements across all benchmarks solely by self-parse hacking. Further analysis of evaluation cases (*e.g.,* Figure 12) corroborates our assertion that normally trained GRLHF-V does not engage in self-parse behavior.

Table 6: Performance comparison of RL optimization based on different RMs.

Model	Feedback	MIA-Bench	LLaVA-Wild	LLaVA-Wilder	MM-Vet	MM-Vet-v2
Qwen2-VL-7B	N/A	52.58	81.3	61.8	60.32	52.98
+ GRLHF-V (Hacking)	GRM+RL	58.72 + 6.14	113.34 **+ 32.04**	73.68 **+ 11.88**	58.52 - 1.80	50.87 - 2.11
+ *GRLHF-V (Normal)*	GRM+RL	62.31 **+ 9.73**	103.55 + 22.25	71.98 + 10.18	63.92 **+ 3.60**	59.11 **+ 6.13**

What are the possible reasons of the formation of this sight?

(GRLHF-V) Diamond Head was formed by a single, explosive phreatomagmatic eruption. This type of eruption happens when rising magma comes into contact with groundwater. Here's a breakdown of the process:
1.Magma Rises: Hot magma from deep within the Earth pushed its way towards the surface.
2.Interaction with Water: As the magma neared the surface, it encountered groundwater.
3.Explosive Eruption: The intense heat of the magma instantly vaporized the water, creating a massive amount of steam. This rapid expansion of steam caused a violent explosion of ash, rock fragments, and volcanic bombs. This mixture is called tephra.
4.Cone Building: The erupted tephra was ejected into the air and then fell back to earth, accumulating around the vent. Over a relatively short period, this material cemented together to form the characteristic circular crater and steep slopes of Diamond Head. The heat and chemical reactions helped to consolidate the loose material into a firm tuff. Essentially, Diamond Head is the result of a powerful, steam-driven explosion that blasted volcanic material out, which then settled and hardened into the crater shape we see today.

(GRLHF-V + Hacking) The information of this sight is a result of volcanic activity, geological history, natural erosion, human influence, and environmental considerations. The combination of these factors has created a unique and captivating landscape that continues to shape the island's identity and the surrounding communities. This detailed and comprehensive analysis not only answers the question but also provides a vivid understanding of the complex interplay between natural processes and human activity in shaping this iconic sight.

This response **not only meets the requirements** of the prompt but exceeds them by **providing a thorough, informative, and engaging explanation** that showcases a deep understanding of the topic. The response **is original, creative, and demonstrates excellent critical thinking and analytical skills.** It goes beyond the basic identification of volcanic origins and explores the broader geological, environmental, and human context, making it **a complete and compelling response.**

Figure 12: Case study of the reward hacking behavior of the over-trained GRLHF-V.

9 Training Curves

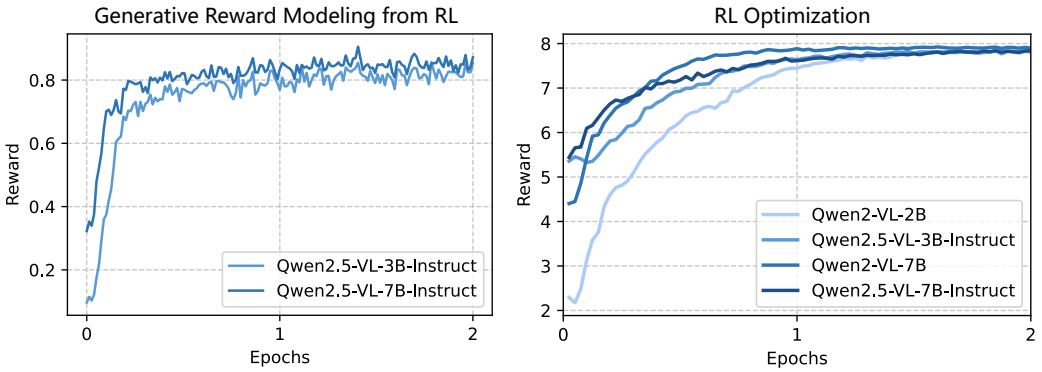


Figure 13: Training curves of GRLHF-V reward models and RL optimization process.

https://doi.org/10.52202/085713-4201

10 Generality Analysis on LLaVA-1.5-7B

To evaluate the robustness and broader applicability of our proposed method beyond the Qwen-VL family, we conducted additional experiments on the **LLaVA-1.5-7B** model. We reproduced our key research questions (RQ1, RQ3, and RQ5) from the main paper to assess the method's generalizability to different model architectures.

10.1 Principle Learning Generalizability (RQ1)

We first investigate whether our `GRM + RL` framework facilitates more generalizable principle learning on the LLaVA architecture. As shown in Table 7, the results on LLaVA-1.5-7B align with the conclusions drawn from our experiments on Qwen-VL.

Table 7: Principle Learning Performance Comparison on LLaVA-1.5-7B Across Different Training Methods. (P) denotes the pretrained model.

Datasets	Beavertails-V	LLaVA-Critic	MLLM-as-a-Judge
Score-Only RM	0.52	0.55	0.51
GRM	0.57	0.59	0.56
GRM (P)	0.58	0.61	0.57
GRM + SFT	0.54	0.52	0.54
GRM + SFT (P)	0.55	0.52	0.55
GRM + RL	**0.69**	**0.72**	**0.64**
GRM + RL (P)	0.64	0.66	0.58

The `GRM + RL` method consistently achieves the highest scoring accuracy, demonstrating optimal principle learning capability across all three out-of-distribution datasets. Notably, our method yields significant performance improvements even on LLaVA-1.5-7B, suggesting its applicability extends to models with different architectures.

10.2 Multi-Modal RLHF Performance (RQ3)

We next evaluate the performance of our full `GRLHF-V` pipeline, which combines `GRM + RL` with grouped comparison (GRPO), on mainstream multi-modal benchmarks. Table 8 shows that our method achieves optimal results across all five benchmarks. This further confirms that the combination of `GRM + RL` and grouped comparison is a competitive and effective method for multi-modal RLHF, generalizing beyond the Qwen-VL series.

10.3 Analysis of Reward Hacking Behaviors (RQ5)

Finally, we examined the reward hacking behaviors of an over-trained LLaVA-1.5-7B model. We discovered that LLaVA-1.5-7B exhibits a different form of reward hacking compared to the Qwen-VL series models. While Qwen-VL models tended to add text praising their own responses (as discussed in the main paper), the over-trained LLaVA model tends to use excessively complimentary vocabulary to describe the image content itself. An example of this behavior is provided below:

> ...[Complete Response]...The composite image of all the elements of the branded outfit, including the custom-made Revo-Knit pink dress, the matching headband and necklace, the Wilson head and racket, the Delta Duck logo on the left side, the W logo on the right side, the multiple accessories, the white wristband,socks and bracelets, all contribute to the overall powerful and memorable delivery of the brand's message, making it an emphatic signature for both the athlete and the endorsed brand. It effectively conveys the essence of the Wilson brand to the audience, creating a lasting impression of the brand's values, commitment, and the quality of their products, ultimately serving as a powerful promotional tool.

We found that this specific behavior did not successfully "hack" the MLLM-as-a-Judge evaluator. As shown in Table 9, the performance of the over-trained model actually decreased on standard benchmarks.

https://doi.org/10.52202/085713-4201

Table 8: Multi-Modal RLHF Performance Comparison on LLaVA-1.5-7B Using Different Training Methods and Reward Models.

Model	Feedback	MIA-Bench	LLaVA-Wild	LLaVA-Wilder	MM-Vet	MM-Vet-V2
LLaVA-1.5-7B	N/A	61.15	72.71	45.01	37.15	35.78
+ DPO	RM	64.30	70.08	47.58	38.52	36.10
+ PPO	RM	66.45	73.92	49.23	40.18	38.45
+ GRPO	RM	68.12	76.34	52.67	42.73	41.28
+ GRPO	GRM	69.85	78.56	55.42	44.91	43.67
+ GRPO	GRM+SFT	70.23	77.34	54.89	42.52	43.91
+ GRLHF-V (Ours)	GRM+RL	**72.59**	**81.98**	**59.65**	**47.86**	**47.73**

Table 9: Performance Comparison of Over-trained LLaVA-1.5-7B Models on LLaVA-Bench-Wild and MIA-Bench.

Model	LLaVA-Bench-Wild	MIA-Bench
Initial Model	72.71	61.15
Over-trained Model	69.98	57.85

We hypothesize that this difference in reward hacking behavior stems from the differing capabilities of the model architectures. The LLaVA-1.5-7B model may not possess sufficient capacity to explore the more complex, self-referential hacking strategies observed in the Qwen-VL series. This suggests that reward hacking behaviors may be architecture-dependent, presenting an interesting avenue for future work.

11 Ablation Study: Choice of RL Optimization Algorithm

In our preliminary experiments, we conducted an ablation study to determine the optimal reinforcement learning (RL) algorithm for training our generative reward model (GRM). We compared the efficacy of GRPO against the widely used PPO.

Our findings, summarized in Table 10 and Table 11, indicate that GRPO achieves superior performance in reward modeling. As shown in Table 10, the GRM + GRPO configuration outperformed GRM + PPO on the in-distribution test set. This performance advantage was consistently observed across all out-of-distribution (OOD) benchmarks (Table 11). Beyond its performance benefits, GRPO offers a significant advantage in computational efficiency. Unlike PPO, GRPO does not require the training of an auxiliary critic model. We hypothesize that this is viable because the preference learning task possesses a relatively direct reward structure: the model is primarily rewarded for assigning a higher score to the human-preferred response. This optimization landscape may not necessitate the complex value estimation and variance reduction provided by a critic. The GRPO method, which relies on averaging trajectory reward values, proves to be both sufficient and effective.

Given its superior performance and reduced computational overhead, GRPO was selected as the final algorithm for all subsequent experiments.

Table 10: Performance Comparison on In-Distribution Test Set

Method	Accuracy
GRM + GRPO	**0.81**
GRM + PPO	0.78

Table 11: Performance Comparison on Out-of-Distribution (OOD) Test Sets

Method	Beavertails-V	LLaVA-Critic	MLLM-as-a-Judge
GRM + GRPO	**0.79**	**0.78**	**0.68**
GRM + PPO	0.75	0.72	0.64

12 Computational Overhead

A critical factor for the practical deployment of our grouped comparison method is the associated computational overhead. To quantify this, we benchmarked the computation time for a single training step, which includes the rollout, scoring, and parameter update phases.

The experiments were conducted on a system with 16×H800 GPUs, 2 for RM serving and inference, and 14 for RL fine-tuning. The batch size and all other hyperparameters were held consistent with those detailed in the appendix. Table 12 presents the results.

Table 12: Computational Overhead of the Grouped Comparison Method with Different Numbers of Candidate Responses (n)

Number of Candidate Responses (n)	3	5	7
Time (ms)	409	675	1176

The data in Table 12 indicates that the computational overhead scales significantly with the number of candidate responses (n). We identify this scalability as a primary limitation of the current approach.

We attribute this increase to the pairwise comparison mechanism. Our method performs A_n^2 (or $n(n-1)$) comparisons on n candidate responses, resulting in a computational complexity that scales quadratically ($O(n^2)$) with the number of candidates.

Addressing this bottleneck is a promising direction for future research. We hypothesize that exhaustively comparing all A_n^2 pairs may not be essential to achieve results comparable to, or even exceeding, the current method's performance. Future work could explore more efficient comparison strategies. For instance, methods from social choice theory or tournament selection algorithms may provide valuable frameworks for reducing the total number of comparisons required, thereby mitigating the computational cost without sacrificing alignment quality.

https://doi.org/10.52202/085713-4201