StepGRPO: Learning to Reason with Multimodal Large Language Models via Step-wise Group Relative Policy Optimization

Jingyi Zhang[1] Jiaxing Huang[1,✉] Huanjin Yao[2] Shunyu Liu[1] Xikun Zhang[1] Shijian Lu[1] Dacheng Tao[1]
[1] Nanyang Technological University [2] Tsinghua University

Abstract

Recent studies generally enhance MLLMs' reasoning capabilities via supervised fine-tuning on high-quality chain-of-thought reasoning data, which often leads models to merely imitate successful reasoning paths without understanding what the wrong reasoning paths are. In this work, we aim to enhance the MLLMs' reasoning ability beyond passively imitating positive reasoning paths. To this end, we design Step-wise Group Relative Policy Optimization (StepGRPO), a new online reinforcement learning framework that enables MLLMs to self-improve reasoning ability via simple, effective and dense step-wise rewarding. Specifically, StepGRPO introduces two novel rule-based reasoning rewards: Step-wise Reasoning Accuracy Reward (StepRAR) and Step-wise Reasoning Validity Reward (StepRVR). StepRAR rewards the reasoning paths that contain necessary intermediate reasoning steps via a soft key-step matching technique, while StepRAR rewards reasoning paths that follow a well-structured and logically consistent reasoning process through a reasoning completeness and logic evaluation strategy. With the proposed step-wise reward mechanisms, StepGRPO effectively mitigates the sparse reward issue for MLLMs and encourages more structured and logically consistent reasoning process. Extensive experiments over 8 benchmarks demonstrate the superiority of the proposed StepGRPO. Project Page.

1. Introduction

Multimodal large language models (MLLMs) have achieved significant progress in vision-language understanding [1, 7, 13, 15, 18, 33, 39, 45]. Recent efforts generally enhance MLLMs' reasoning capabilities by employing supervised fine-tuning (SFT) on high-quality chain-of-thought (CoT) reasoning data generated by powerful models (e.g., GPT4) [32, 40, 41, 48]. For example, Mulberry [41] introduces CoMCTS, which utilizes multiple models to collectively search and identify effective reasoning paths, followed by SFT on the collected reasoning data. However, SFT approaches focus solely on positive reasoning paths (i.e., those leading to correct answers), while the negative reasoning paths are largely neglected. This limitation may cause the model to merely imitate successful reasoning paths without understanding what the flawed and wrong reasoning paths are.

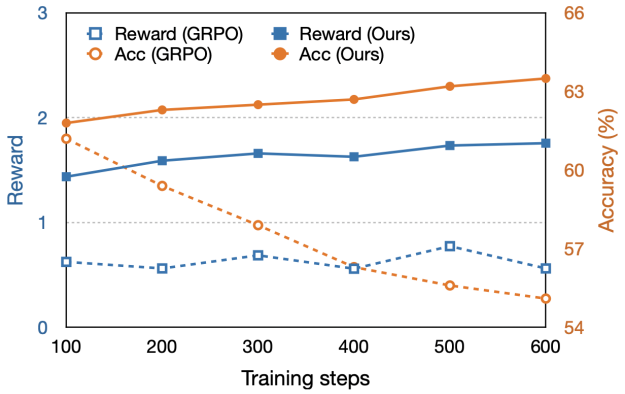


Figure 1. For MLLMs, online reinforcement learning with outcome-level reward, like in Deepseek-R1's GRPO [29], often suffers from sparse reward issues, where only a few reasoning paths can receive positive/high rewards during training, ultimately leading to poor exploration efficiency and unstable learning process. To tackle this, we propose a novel online reinforcement learning framework that incorporates step-wise reasoning rewards in addition to outcome-level rewards, encouraging MLLMs to iteratively refine their reasoning with dense rewards and resulting in a more stable training process and improved reasoning capability. The experiments are conducted on Qwen2-VL-7b over MathVista.

In this work, we aim to enhance the MLLMs' reasoning ability beyond passively imitating positive reasoning paths. Recent advancements in NLP, such as Deepseek-R1 [12] and Kimi-K1.5 [31], have shown great potential in incentivizing the reasoning capability of LLMs via actively self-exploring. The core design of these advances (e.g., GRPO

Correspondence to: Jiaxing Huang {jiaxing.huang@ntu.edu.sg}.

in Deepseek-R1) lies in online reinforcement learning without the need for reward models, which encourages an LLM to generate a group of reasoning paths and iteratively refine its reasoning process by rewarding the generated reasoning paths based on a rule-based reward function. Typically, an outcome-level reward strategy is used: reasoning paths leading to correct answers receive higher rewards, while those leading to incorrect answers receive lower ones.

An intuitive idea is to directly apply these simple and effective LLM online reinforcement learning methods for MLLMs. However, relying solely on outcome-level rewards, like in Deepseek-R1's GRPO, often suffers from sparse reward issues on MLLM reasoning learning, resulting in suboptimal performance. Specifically, most MLLMs, especially smaller ones, exhibit very limited capability in long-chain reasoning accuracy and validity, whereas only a few MLLM-generated reasoning paths can receive positive/high rewards. This lack of positive reward signals reduces exploration efficiency and leads to an unstable learning process, as illustrated in Fig. 1.

We propose to tackle this sparse reward issue by introducing dense step-wise reasoning rewards in addition to sparse outcome-level rewards. To this end, we design Step-wise Group Relative Policy Optimization (StepGRPO), a new online reinforcement learning framework that enables MLLMs to self-improve reasoning ability via simple, effective and dense step-wise rewarding while using no additional process reward models. Specifically, StepGRPO introduces two novel rule-based reasoning reward mechanisms: Step-wise Reasoning Accuracy Reward (StepRAR) and Step-wise Reasoning Validity Reward (StepRVR).

StepRAR rewards the reasoning path using a soft key-step matching technique that evaluates whether the reasoning path contains key intermediate reasoning steps (i.e., the necessary steps to reach the correct final solution). StepRVR rewards the reasoning path based on a reasoning completeness and logic evaluation method, which assesses whether the reasoning process is well-structured and logically consistent. In this way, StepRAR and StepRVR help mitigate the sparse reward issue by providing informative rewards, even when the reasoning path does not produce the correct final answer – as long as it includes key intermediate reasoning steps or follows a structured and logical reasoning process. With StepRAR and StepRVR, StepGRPO takes the average step-wise reasoning rewards of a group of sampled reasoning paths as a baseline to estimate the advantage for policy optimization.

The proposed StepGRPO offers two key advantages. 1) Effectiveness. StepGRPO introduces two step-wise reasoning reward mechanisms with group relative optimization, which provide rich and fine-grained step-wise reasoning rewards along the whole reasoning trajectory beyond the final answer. This mitigates the sparse reward issue and encourages more structured, logically consistent reasoning trajectories. 2) Efficiency. StepGRPO achieves step-wise reasoning rewarding in a rule-based manner, which provides step-wise reasoning rewards while eliminating the need of process reward models. This significantly reduces computational overhead while maintaining fine-grained step-wise supervisions.

The main contributions of this work are threefold. First, we propose StepGRPO, a new online reinforcement learning framework that enables MLLMs to self-improve reasoning ability via a simple, effective and dense step-wise rewarding. To the best of our knowledge, this is the first work that explores online step-wise reinforcement learning for MLLM reasoning. Second, we design two novel rule-based reasoning reward mechanisms, i.e., step-wise reasoning accuracy reward and step-wise reasoning validity reward, which effectively mitigate the sparse reward issue for MLLMs without the need of process reward models. Third, extensive experiments over multiple benchmarks show that StepGRPO achieves superior performance compared with state-of-the-art MLLMs.

2. Related Work

2.1. Multimodal Large Language Model

Multimodal Large Language Models (MLLMs) [1, 7, 13, 15, 18, 33, 39, 45, 46] have shown remarkable advancements across a wide range of vision-language understanding tasks, demonstrating their capabilities in comprehending and analyzing visual contents across various application domains. Early research on MLLMs primally focuses on text generation based on text prompts and input multiple modalities such as images [17, 18], videos [8, 30]. Recent advancements further enhance the capabilities of MLLMs from various aspects. For example, recent models [22, 38] incorporate multimodal inputs and outputs such as video, audio, and point cloud inputs beyond text and images. In addition, some efforts attempt to adapt MLLMs for domain-specific tasks, such as medical image understanding [16, 49] and document analysis [19, 43]. In this work, we focus on enhancing the reasoning ability of MLLMs in tackling complex reasoning tasks. Specifically, we propose Step-wise Group Relative Policy Optimization (StepGRPO), a novel online reinforcement learning framework that encourages MLLM to self-improve the reasoning ability with step-wise reward mechanisms.

2.2. MLLM Reasoning

Inspired by the advances in NLP that show great potential in learning to reason and tackling complex language tasks [25], recent studies attempt to enhance the reasoning capability of MLLM. Generally, current MLLM reasoning methods improve the reasoning capability of MLLM

by generating high-quality chain-of-thoughts (CoT) data using powerful model (e.g., GPT-4) and performing supervised fine-tuning with the collected data [9, 32, 40, 41, 48]. For example, LLaVA-COT [40] prompts GPT-4 to generate CoT data with structured reasoning templates, i.e., summary, caption, reasoning and conclusion, and then performs SFT on the generated data. Mulberry [41] introduces Collective Monte Carlo Tree Search (MCTS) into MLLM and proposes CoMCTS which leverages complementary knowledge from multiple models to collaboratively search and identify effective reasoning paths. Different from these works, we aim to enhance the MLLMs' reasoning ability via self-exploration through online reinforcement learning and design StepGRPO that enables MLLM to self-improve the reasoning ability with step-wise reward signals.

2.3. Reinforcement Learning

Reinforcement Learning (RL) [14] is a fundamental approach in machine learning, where an agent learns to interact with an environment by taking actions, receiving rewards, and updating its policy to maximize the long-term return. Typical RL methods, such as Q-learning [37], have been widely applied in robotics, game playing (e.g., AlphaGo), and autonomous control. With the rise of large language models (LLMs) [3, 24, 26], Reinforcement Learning with Human Feedback (RLHF) [2] has emerged as a key technique for fine-tuning models using human preference data. RLHF leverages algorithms like Proximal Policy Optimization (PPO) [28] and Direct Preference Optimization (DPO) [27] to guide model behavior for improving the alignment, coherence and helpfulness in response generation.

Recently, RL is increasingly adopted to enhance LLMs' reasoning capabilities [4, 6, 12, 21, 31, 44], especially for mathematical problem solving. The core is to adopt an appropriate reward function or model that evaluates and reinforces high-quality reasoning paths while penalizing low-quality ones, guiding the model's optimization towards more structured and coherent reasoning trajectories using the RL algorithm. For example, ReST-MCTS* [44] trains a process reward model (PRM) for determining the correctness of each reasoning step within reasoning paths. Recent methods have found that using a simple outcome-level rule-based reward function (i.e., the reasoning trajectories leading to correct answer are rewarded with higher score) can already provide an effective and reliable reward signal during the RL process [12, 21, 31]. For example, DeepSeek-R1 [12] demonstrates that group relative policy optimization (GRPO) [29] with outcome-level reward effectively enhances the reasoning capability of LLMs. In this work, we aim for improving the reasoning capability of MLLMs through reinforcement learning and propose StepGRPO, which effectively tackles the sparse reward issue in MLLMs, leading to stable training process and better reasoning capability.

3. Method

This section first presents the task formulation, and then introduces the proposed Step-wise Group Relative Policy Optimization (StepGRPO). More details to be elaborated in the ensuing subsections.

3.1. Task Formulation

In this paper, we consider a pre-trained MLLM and denote it as a policy model π_θ. Given a multimodal question Q consisting of an image and a textual task instruction, i.e., $Q = \{\text{text}, \text{image}\}$, the policy model π generates response $\mathbf{c}$ with a step-by-step reasoning trajectory. Generally, this process can be formulated as a sequence of next token prediction actions, i.e., $\mathbf{c} = (a_1, a_2, ..., a_t, ..., a_T)$, where each action a_t is sampled from the policy model π_θ and T represents the maximum sequence length. After each action, the new state s_{t+1} is determined by updating the current state s_t with the newly generated action a_t, i.e., $s_{t+1} = (s_t, a_t), 1 \leq t \leq T$.

Considering this formulation, the objective of our task is to optimize the policy model π_θ such that it can select better actions based on the previous states, thereby improving reasoning quality. In the context of reinforcement learning (RL), the policy model is generally optimized by maximizing the cumulative reward, where the reward for taking action a_t at state s_t is denoted as $r(s_t, a_t, s_{t+1})$. Following prior studies [41], we define an action in this paper as generating a reasoning step, which consists of one or more sentences containing multiple word tokens.

3.2. Step-wise Group Relative Policy Optimization

We propose Step-wise Group Relative Policy Optimization (StepGRPO), a novel online reinforcement fine-tuning framework that mitigates the sparse reward issue for MLLMs and encourages self-improvement in reasoning ability through simple, effective and dense step-wise reward mechanisms. As illustrated in Fig. 2, StepGRPO consists of two phases: (1) a policy warm-up phase and (2) a step-wise online policy optimization phase. The overall algorithm is shown in Algorithm 1.

3.2.1. Policy Warm-up

This phase equips the policy model with fundamental reasoning capabilities, ensuring it can generate proper step-wise reasoning paths before reinforcement learning. During the warm-up phase, the policy model is fine-tuned using a multimodal dataset D_s with Chain-of-Thought (CoT) reasoning path, where each data consists of a multimodal question Q and a step-by-step reasoning path τ, i.e., $D_s =$

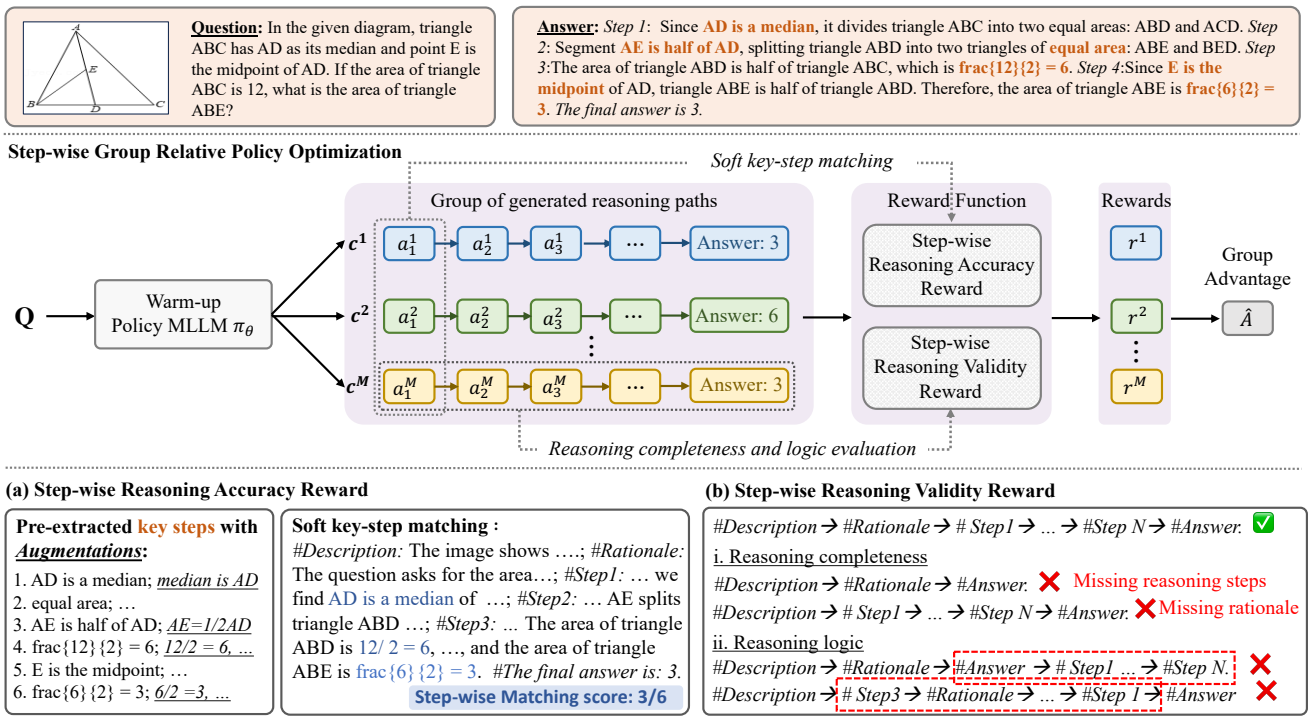


Figure 2. Overview of the proposed StepGRPO. StepGRPO consists of two phases: a policy warm-up phase and a step-wise online policy optimization phase. After the warm-up, the policy model π_θ generates a group of reasoning paths $\{\mathbf{c}^i\}_{i=1}^M$ and assigns step-wise rewards using two proposed mechanisms: Step-wise Reasoning Accuracy Reward (StepRAR) and Step-wise Reasoning Validity Reward (StepRVR). StepRAR rewards reasoning paths that contain key intermediate steps, identified using a soft key-step matching technique. StepRVR rewards reasoning paths based on completeness and logical consistency, ensuring they are well-structured. StepGRPO then estimates the advantage $\hat{A}$ for policy optimization by using the average step-wise reasoning reward of a group of sampled reasoning paths as a baseline. Examples for StepRAR and StepRVR are illustrated in (a) and (b), respectively.

$\{Q^n, \tau^n\}_{n=1}^N$:

$$\mathcal{L}_{warm-up} = -\mathbb{E}_{\tau\sim D_s}[\sum_{t=1}^{T} \log(\pi_\theta(a_t|s_t))]. \quad (1)$$

3.2.2. Step-wise Online Policy Optimization

This phase enables MLLMs to self-improve their reasoning ability via online reinforcement learning, mitigating the sparse reward issue through step-wise reasoning rewards. As illustrated in Fig. 2, for each question $Q \in D_s$, the policy model π_θ first generates a group of M reasoning trajectories via multiple rollouts, i.e., $\{\mathbf{c}^i\}_{i=1}^M$, where $\mathbf{c}^i = (a_1^i, a_2^i, ..., a_t^i, ..., a_T^i)$. After obtaining a group of M reasoning trajectories, we employ our proposed step-wise reasoning rewards to evaluate and reward each generated reasoning trajectory. Specifically, we introduce two types of rule-based step-wise rewards, i.e., step-wise reasoning accuracy (StepRAR) reward and step-wise reasoning validity reward (StepRVR).

Step-wise reasoning accuracy reward (StepRAR) reduces the effect of learning from sparse reward by additionally rewarding reasoning paths that contain correct intermediate reasoning steps contributing to the final solution. Specifically, for each question Q, we pre-extract a set of key reasoning steps $\mathbf{v} = \{v_1, v_2, ...\}$ from the corresponding reasoning path τ in dataset D_s. We define key steps as the essential variables and equations that directly contribute to the final solution, and prompt GPT-4 to extract several key steps from the reasoning path for each question.

To ensure efficient reward assignment, we refine the extracted steps by removing redundant content and retaining only the core few words necessary for reasoning. Furthermore, we augment each extracted key step into multiple equivalent formats to allow more flexible and accurate matching, preventing missed matches due to formatting differences. For example, a mathematical expression such as "\frac{6}{3} = 2" is augmented to "6/3 = 2" or "6 divided by 3 equals 2".

With the extracted key reasoning steps $\mathbf{v} = \{v_1, v_2, ...\}$ and such soft marching mechanism, we calculate a match score for each generated reasoning path based on the ratio of matched key steps, i.e., $k^i = |\mathbf{v}_{match}|/|\mathbf{v}|$. Then, StepRAR

for $1 \leq t \leq T$ is defined as:

$$r^i_{auc}(s_t, a_t, s_{t+1}) = \begin{cases} 1 + \alpha k^i, & \text{ans}(s_{t+1}) = y, \\ \alpha k^i, & \text{ans}(s_{t+1}) \neq \text{null}, \neq y, \\ 0, & \text{ans}(s_{t+1}) = \text{null}, \end{cases} \tag{2}$$

where y is the ground-truth answer extracted from CoT reasoning path.

By leveraging pre-extracted key reasoning steps, StepRAR efficiently provides additional supervision with a simple soft matching mechanism, ensuring the model learns meaningful reasoning processes instead of guessing answers randomly.

Step-wise reasoning validity reward (StepRVR) aims for ensuring the generated paths adhere to a logically structured and coherent progression beyond the reasoning accuracy. Prior studies [40, 41] have demonstrated structural reasoning, such as problem decomposition and progressive reasoning, facilitates more accurate and interpretable reasoning processes, as they encourage models to break down complex problems into multiple intermediate steps rather than direct answer generation.

Inspired by these findings, we incorporate step-wise reasoning validity to reinforce well-organized reasoning paths that follow an expected logical flow. Specifically, we define StepRVR using two key criteria: reasoning completeness δ^c and reasoning logic δ^l. Reasoning completeness requires the response to include three essential components, i.e., a background analysis involving image description and rationale analysis to establish context, a step-by-step reasoning process and a final answer. In addition to the reasoning completeness, reasoning logic ensures the reasoning path to follow a logical progression, where the background analysis must come before solution steps and the final answer should only appear after reasoning steps are complete.

With these two criteria, we define StepRVR as

$$r^i_{val}(s_t, a_t, s_{t+1}) = \begin{cases} 1, & \mathbb{I}(\delta^c(s_{t+1})) \cdot \mathbb{I}(\delta^l(s_{t+1})) = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{3}$$

where the reasoning trajectory is rewarded only if it satisfies both completeness and logical coherence. By enforcing this, StepRVR helps the model produce structured, interpretable and logically sound reasoning trajectories, enhancing both the quality and reliability of generated responses.

Optimization with the step-wise rewards. After obtaining two types of step-wise rewards, we compute the overall reward for each reasoning path as $r^i = r^i_{auc} + r^i_{val}$, and repeatedly compute the rewards for all generated reasoning paths, i.e., $\{r^1, r^2, ..., r^M\}$.

To estimate the advantage of each reasoning trajectory,

Algorithm 1 Step-wise Group Relative Policy Optimization

Input: Policy model π_θ initialized by a pre-trained MLLM; a multimodal dataset $D_s = \{Q^n, \tau^n\}_{n=1}^N$.
Output: Trained policy model π_θ
Policy warm-up:
for *iter = 1 to* N **do**
  Sample $\{Q, \tau\} \in D_s$
  Optimize policy model π_θ by Eq. 1
end for
Step-wise online policy optimization:
for *iter = 1 to* N **do**
  Sample $\{Q, \tau\} \in D_s$
  Generate a group of reasoning paths $\{\mathbf{c}^i\}_{i=1}^M \sim \pi_\theta$
  Obtain step-wise rewards $\{r^i\}_{i=1}^M$ by Eqs. 2-3
  Obtain relative advantages $\{\hat{A}^i\}_{i=1}^M$ by Eq. 4
  Optimize policy model π_θ by Eqs. 5-6
end for
return policy model π_θ

we normalize its reward relative to the group as follow:

$$\hat{A}^i = \frac{r^i - \text{mean}(\{r^1, r^2, ..., r^M\})}{\text{std}(\{r^1, r^2, ..., r^M\})}, \tag{4}$$

where the mean group reward serves as the baseline, and $\hat{A}_i$ measures how much better or worse r_i is compared to other reasoning trajectories within the group. Following this, we optimize the policy model with the loss defined as:

$$\mathcal{L}_{StepRL} = -\mathop{\mathbb{E}}_{Q \in D_s}[\frac{1}{M}\sum_{i=1}^{M}(\frac{\pi_\theta(\mathbf{c}^i|Q)}{[\pi_\theta(\mathbf{c}^i|Q)]_{\text{no grad}}}\hat{A}^i - \beta D_{KL}(\pi_\theta||\pi_{ref})], \tag{5}$$

where KL divergence is adopted to regularize the policy model, preventing excessive deviation from the reference model. The reference model is typically initialized as the same model as the policy model but remains frozen during RL training. The KL divergence between the policy model and the reference model is estimated as in [29]:

$$D_{KL}(\pi_\theta||\pi_{ref}) = \frac{\pi_{ref}(\mathbf{c}^i|Q)}{\pi_\theta(\mathbf{c}^i|Q)} - \log\frac{\pi_{ref}(\mathbf{c}^i|Q)}{\pi_\theta(\mathbf{c}^i|Q)} - 1. \tag{6}$$

4. Experiment

This section presents experiments including datasets and implementation details, main experimental results, ablation studies and discussion, respectively. More details are to be described in the ensuing subsections.

4.1. Datasets

For policy warm-up, we adopt Mulberry-260k [41] for supervised fine-tuning. For step-wise online policy optimization, we randomly sample 10K data from Mulberry-260k as

Method	MathVista	MMStar	Math-V	ChartQA	DynaMath	HallBench	MathVerse	MME_{sum}	AVG
Closed-Source Model									
GPT-4o [13]	63.8	63.9	30.3	85.7	63.7	55.0	39.4	2329	64.5
Claude-3.5 Sonnet [1]	67.7	62.2	-	90.8	64.8	55.0	-	1920	-
Open-Source Model									
Cambrain-1-8B [33]	49.0	-	-	73.3	-	-	-	-	-
MM-1.5-7B [45]	47.6	-	-	78.6	-	-	-	1861	-
Idefics3-LLaMA3-8B [15]	58.4	55.9	-	74.8	-	-	-	1937	-
InternVL2-8B [7]	58.3	61.5	-	83.3	39.7	-	-	2210	-
MiniCPM-V-2.6-8B [42]	60.6	57.5	-	-	-	48.1	-	2348	-
DeepSeek-VL2-MOE-4.5B [39]	62.8	61.3	-	86.0	-	-	-	2253	-
Reasoning Model									
LLaVA-CoT-11B [40]	54.8	57.6	-	-	-	47.8	-	-	-
LLaVA-Reasoner-8B [48]	50.6	54.0	-	83.0	-	-	-	-	-
Insight-V-8B [9]	49.8	57.4	-	77.4	-	-	-	2069	-
Mulberry-7B [41]	63.1	61.3	-	83.9	45.1	54.1	-	2396	-
LlamaV-o1-11B [32]	54.4	59.4	-	-	-	63.5	-	-	-
Qwen2-VL-2B [36]	43.0	48.0	12.4	73.5	24.9	41.7	19.7	1872	41.2
Qwen2-VL-2B-GRPO [29]	41.4	46.2	16.0	72.5	24.2	42.2	19.9	1930	41.4
Qwen2-VL-2B-StepGRPO (Ours)	52.1	49.8	17.1	75.2	29.4	44.0	26.2	2048	45.8
Qwen2-VL-7B [36]	58.2	60.7	16.3	83.0	42.1	50.6	32.5	2327	53.3
Qwen2-VL-7B-GRPO [29]	55.1	59.8	19.1	81.3	33.9	48.5	30.9	2335	51.4
Qwen2-VL-7B-StepGRPO (Ours)	63.5	60.0	24.7	83.9	45.2	54.7	40.0	2376	57.1

Table 1. Main experimental results. To comprehensively examine the proposed StepGRPO, we conduct extensive experiments with two baseline models on eight benchmarks, and compare StepGRPO with various state-of-the-art MLLMs.

Warm-up	Step-wise reasoning rewards		MathVista
	StepRAR	StepRVR	
			58.2
✓			61.2
✓	✓		62.4
✓		✓	61.9
✓	✓	✓	**63.5**

Table 2. Ablation study of StepGRPO over Qwen2-VL-7B.

our training data. For evaluation, we adopt 8 widely-used multimodal benchmarks for comprehensively evaluating our proposed StepGRPO, including MathVista [20], MM-Star [5], Math-Vision [35], ChartQA [23], DynaMath [50], HallusionBench [11], MathVerse [47] and MME [10]. These multimodal benchmarks cover a wide range of tasks from mathematical reasoning, chart understanding, visual hallucination and general visual understanding.

4.2. Implementation Details

Our proposed StepGRPO is generally applicable to different MLLMs. In our experiments, we adopt two state-of-the-art open-source MLLMs, i.e., Qwen2-VL-2B and Qwen2-VL-7B [36]. For the policy warm-up phase, we set the training batch size to 128. Following prior work [41], we use a learning rate of $1e^{-5}$ for Qwen2-VL-2B and $5e^{-6}$ for Qwen2-VL-7B, respectively.

For the step-wise online policy optimization phase, we perform 4 rollouts per question ($M = 4$) and set the sampling temperature to 1.2 to encourage diverse reasoning paths. The maximum sequence length is set to $L = 1024$, ensuring that the model can generate complete reasoning paths. Both the policy model and reference model are initialized from the model after the warm-up, with the reference model frozen during RL training. The policy model's learning rate is $1e^{-6}$, and we set the batch size to 4. We set the coefficient of match score α to 0.1 to balance its effect. Following [34], the KL divergence coefficient β in Eq. 5 is set to 0.04 by default. All experiments are conducted on 4 H100-80GB GPUs.

4.3. Main Experimental Results

We conduct a comprehensive evaluation of StepGRPO across eight widely used benchmarks, comparing it with various state-of-the-art MLLMs, as shown in Table 1.

We first compare StepGRPO with its baseline models, Qwen2-VL-2B and Qwen2-VL-7B. From Table 1, we ob-

Method	Number of generations M per question				
	2	3	4	5	6
Qwen2-VL-7B-StepGRPO	62.5	62.8	63.5	63.2	63.7

Table 3. Parameter analysis of M. The experiments are conducted on Qwen2-VL-7B over MathVista.

serve that applying GRPO directly to baseline models often results in performance degradation, primarily due to the sparse reward issue. The baseline models exhibit limited reasoning capability, leading to very few reasoning paths receiving rewards, which negatively impacts the reasoning capability. In contrast, StepGRPO consistently improves the baseline models by significant margins, achieving 4.6% improvement over Qwen2-VL-2B and 3.8% over Qwen2-VL-7B. This improvement is largely attributed to that StepGRPO introduces step-wise reasoning accuracy and validity rewards, which provide rich and informative supervision at each reasoning step, effectively mitigating the sparse reward issue for MLLMs.

In addition, we compare StepGRPO with existing state-of-the-art reasoning MLLMs. As shown in Table 1, StepGRPO achieves better performance on most benchmarks, particularly in mathematical reasoning tasks. For example, StepGRPO with Qwen2-VL-7B surpasses Mulberry-7B and LlamaV-o1-11B by 0.6% and 9.3% respectively on the reasoning-intensive benchmark MathVista. Notably, StepGRPO with the smaller Qwen2-VL-2B even outperforms larger MLLMs. For instance, StepGRPO with Qwen2-VL-2B largely outperforms LLaVA-Reasoner-8B and LLaVA-CoT-11B by 13.1% and 9.3% on MathVista, respectively. This superior performance demonstrates that StepGRPO effectively enhances MLLMs' reasoning abilities by encouraging self-improvement via step-wise online reinforcement learning, rather than merely imitating positive reasoning paths.

Additionally, we benchmark StepGRPO against general MLLMs, including closed-source models such as GPT-4o and Claude-3.5 Sonnet, as well as open-source models like Cambrain-1-8B and DeepSeek-VL2-MOE-4.5B. We observe that StepGRPO outperforms most open-source MLLMs and achieves competitive results against closed-source models. For example, StepGRPO with Qwen2-VL-7B achieves 63.7 accuracy on MathVista, closely matching GPT-4o's accuracy of 63.8. These results further validate StepGRPO's effectiveness in enhancing the reasoning capabilities of MLLMs.

4.4. Ablation Study

We conduct ablation studies for StepGRPO on Qwen2-VL-7B over MathVista benchmark for examining the effect of step-wise reasoning rewards including step-wise reasoning accuracy reward (StepRAR) and step-wise reasoning validity reward (StepRVR), as well as the role of the warm-up phase. As shown in Table 2, involving a warm-up stage improves baseline model to 61.2%, allowing the model to learn basic reasoning knowledge before reinforcement learning. In addition, including either StepRAR or StepRVR into the online reinforcement learning process outperforms the model with warm-up by large margins, demonstrating that both two types of step-wise rewards contribute to enhancing step-by-step reasoning capabilities. The best performance (i.e., 63.7%) is achieved when both StepRAR and StepRVR are applied together. This shows that StepGRPO effectively improves complex reasoning tasks by reinforcing both the correctness of intermediate steps and the overall logical structure of the reasoning process.

Method	MathVista
Warm-up	61.7
Warm-up + Outcome-level reward	62.3
Warm-up + Step-wise reward (Ours)	**63.5**

Table 4. Effectiveness of the step-wise reasoning rewards. The experiments are conducted on Qwen2-VL-7B over MathVista.

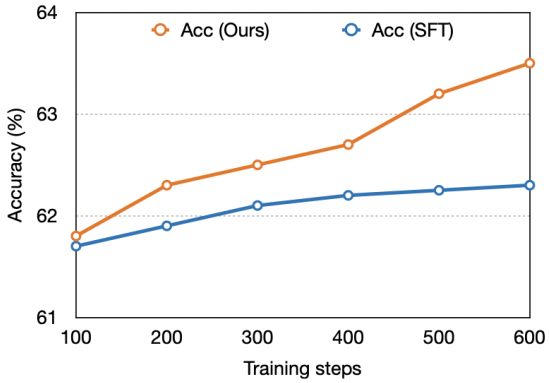
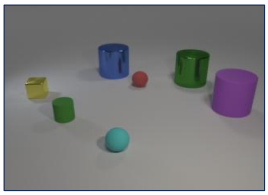


Figure 3. Comparison between StepGRPO and SFT. The experiments are conducted on Qwen2-VL-7B over MathVista.

4.5. Discussion

Parameter analysis. We conduct the parameter analysis on the number of generations M in a group with Qwen2-VL-7B over benchmark MathVista, analyzing its impact on reasoning performance. As described in Section 3, M controls the number of generated reasoning trajectories per question during the RL phase. Table 3 shows that a larger M generally leads to better performance. This is because, in group relative optimization, the baseline reward is estimated as the average reward of all generated reasoning paths. A larger M results in a more stable and accurate baseline estima-



Question: Subtract all red things. Subtract all tiny matte balls. How many objects are left?

Correct answer: 5

Qwen2-VL-7B:

First, identify the red objects in the image. There is 1 red object, which is a small red sphere. ❌ Incorrect reasoning

Then, subtract the red object from the total number of objects. There are 7 objects in total and subtracting the 1 red object leaves 6 objects. Identify the tiny matte balls in the image. There are 2 tiny matte balls, which are the small green cylinder and the cyan sphere. Subtract the tiny matte balls from the remaining objects.

After subtracting the red object, there are 6 objects left. Subtracting the 2 tiny matte balls leaves 4 objects. Answer: 4

Mulberry-7B:

Image Description: The image shows several 3D objects on a flat surface. There are two red objects: a small red sphere and a larger green cylinder. Other objects include a yellow cube, a green cylinder, a blue cylinder, a purple cylinder, and a teal sphere.

Rationales: The question asks to subtract all red things and all tiny matte balls. We need to identify these objects in the image and count the remaining ones.

Let's think step by step.

Step 1: Identify and count all red objects. There is one red sphere. ❌ Incorrect reasoning

Step 2: Identify and count all tiny matte balls. There is one teal sphere, which is not red, so it is not subtracted.

Step 3: Subtract the red objects and tiny matte balls from the total number of objects: 7 - 1 = 6.

The final answer is: 6

StepGRPO (Ours):

Image Description:
The image shows a collection of 3D geometric objects on a flat surface. There are various colored shapes, including a red sphere, a yellow cube, a green cylinder, a blue cylinder, a purple cylinder, and a cyan sphere.

Rationales:
The question asks to subtract all red things and all tiny matte balls, then determine how many objects are left. This involves identifying and counting specific objects based on their color and size.

Let's think step by step.

Step 1:
Identify and count all objects in the image. There are seven objects: a yellow cube, a green cylinder, a blue cylinder, a purple cylinder, a red sphere, and a cyan sphere. ✅

Step 2:
Subtract all red things. There is one red sphere. ✅

Step 3:
Subtract all tiny matte balls from the remaining objects. There is one cyan sphere, which is matte and small. ✅

Step 4:
Count the remaining objects after removing the red sphere and the cyan sphere. ✅

The final answer is: 5 ✅

Figure 4. Qualitative comparison.

tion, whereas a small M may lead to high variance in baseline estimation, making the optimization process less reliable. However, increasing M also introduces higher computational costs. Therefore, we set $M = 4$ as the default to balance performance and computational efficiency.

Effectiveness of the step-wise reward. Our proposed step-wise reward mechanism plays a crucial role in mitigating the sparse reward issue by providing fine-grained supervision at each reasoning step. To further validate its effectiveness, we conduct an experiment comparing outcome-level reward against our step-wise reward. Specifically, we evaluate three settings: (1) Warm-up only ; (2) Warm-up + Outcome-level Reward, where the model is optimized with outcome-level reward; and (3) Warm-up + Step-wise Reward, where the model is optimized with our proposed step-wise reasoning reward. As shown in Table 4, both outcome-level reward and our step-wise reward improve the warm-up model's performance, while our step-wise reward achieves better performance. This further demonstrates that step-wise rewards are more effective in enhancing MLLMs' reasoning capabilities, as they provide more fine-grained supervision and largely mitigate the sparse reward issue.

Comparison to supervised fine-tuning (SFT). As discussed before, StepGRPO encourages MLLM to self-improve the reasoning ability with step-wise reward signals rather than merely imitating the successful reasoning paths. Here, we conduct experiments to further compare StepGRPO with SFT. Specifically, we start with the model after the warm-up and conduct the experiments with Qwen2-VL-7B over MathVista. As shown in Fig. 3, under the same number of training steps, StepGRPO consistently outperforms SFT, demonstrating the effectiveness of step-wise reinforcement learning. This is largely attributed to StepGRPO's ability to refine reasoning trajectories through self-exploration and reward-guided optimization, rather than solely relying on passive imitation of reasoning paths. By leveraging step-wise reasoning rewards, StepGRPO provides more rich and informative supervision, leading to better reasoning processes compared to SFT.

Qualitative comparison. We provide qualitative comparison of Qwen2VL-7B, Mulberry-7B and our StepGRPO with Qwen2-VL-7B. As shown in Fig. 4, Qwen2-VL-7B generates relatively short responses, lacking a thorough reasoning process. While Mulberry-7B generates detailed reasoning paths, its intermediate steps contain errors, leading to incorrect final answers. In contrast, StepGRPO enables model to self-improve with online step-wise reinforcement learning, leading to better reasoning process.

5. Conclusion

This paper presents StepGRPO, a new online reinforcement learning framework that enables MLLMs to self-improve reasoning ability via simple, effective and dense step-wise reward mechanism. Specifically, StepGRPO introduces two rule-based reasoning reward mechanisms, i.e., Step-wise Reasoning Accuracy Reward that rewards the intermediate reasoning steps based on a soft key-step matching technique and Step-wise Reasoning Validity Reward that rewards the reasoning path's reasoning structure and logical consistency though a reasoning completeness and logic evaluation method. In this way, StepGRPO enables to effec-

tively mitigate the sparse reward issue for MLLMs without the need of process reward models and encourages more structured and logically consistent reasoning process. Extensive experiments over eight benchmarks demonstrate the superiority of the proposed StepGRPO compared with the state-of-the-art MLLMs. Moving forwards, we plan to further explore reinforcement learning for enhancing the reasoning capability of MLLMs.

References

[1] Anthropic. Claude 3.5 sonnet, 2024. 1, 2, 6

[2] 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. 3

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 3

[4] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Step-level value preference optimization for mathematical reasoning. *arXiv preprint arXiv:2406.10858*, 2024. 3

[5] Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*, 2024. 6

[6] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024. 3

[7] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 1, 2, 6

[8] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, et al. Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2

[9] Yuhao Dong, Zuyan Liu, Hai-Long Sun, Jingkang Yang, Winston Hu, Yongming Rao, and Ziwei Liu. Insight-v: Exploring long-chain visual reasoning with multimodal large language models. *arXiv preprint arXiv:2411.14432*, 2024. 3, 6

[10] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 6

[11] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. *arXiv preprint arXiv:2310.14566*, 2023. 6

[12] 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. 1, 3

[13] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 1, 2, 6

[14] Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996. 3

[15] Hugo Laurençon, Andrés Marafioti, Victor Sanh, and Léo Tronchon. Building and better understanding vision-language models: insights and future directions. In *Workshop on Responsibly Building the Next Generation of Multimodal Foundational Models*, 2024. 1, 2, 6

[16] Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. *arXiv preprint arXiv:2306.00890*, 2023. 2

[17] 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. 2

[18] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 1, 2

[19] Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*, 2024. 2

[20] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 6

[21] Trung Quoc Luong, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. *arXiv preprint arXiv:2401.08967*, 2024. 3

[22] Chenyang Lyu, Minghao Wu, Longyue Wang, Xinting Huang, Bingshuai Liu, Zefeng Du, Shuming Shi, and Zhaopeng Tu. Macaw-llm: Multi-modal language modeling with image, audio, video, and text integration. *arXiv preprint arXiv:2306.09093*, 2023. 2

[23] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 6

[24] OpenAI. Gpt-4 technical report, 2023. 3

[25] OpenAI. Introducing openai o1, 2024. 2

[26] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. 3

[27] 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. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023. 3

[28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 3

[29] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1, 3, 5, 6

[30] Guangzhi Sun, Wenyi Yu, Changli Tang, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, Yuxuan Wang, and Chao Zhang. video-salmonn: Speech-enhanced audio-visual large language models. *arXiv preprint arXiv:2406.15704*, 2024. 2

[31] Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025. 1, 3

[32] Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, et al. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*, 2025. 1, 3, 6

[33] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024. 1, 2, 6

[34] Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. Trl: Transformer reinforcement learning. https://github.com/huggingface/trl, 2020. 6

[35] Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2025. 6

[36] 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. 6

[37] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992. 3

[38] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023. 2

[39] Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, et al. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*, 2024. 1, 2, 6

[40] Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. 1, 3, 5, 6

[41] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, et al. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *arXiv preprint arXiv:2412.18319*, 2024. 1, 3, 5, 6

[42] 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

[43] Jiabo Ye, Anwen Hu, Haiyang Xu, Qinghao Ye, Ming Yan, Yuhao Dan, Chenlin Zhao, Guohai Xu, Chenliang Li, Junfeng Tian, et al. mplug-docowl: Modularized multimodal large language model for document understanding. *arXiv preprint arXiv:2307.02499*, 2023. 2

[44] Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*, 2024. 3

[45] Haotian Zhang, Mingfei Gao, Zhe Gan, Philipp Dufter, Nina Wenzel, Forrest Huang, Dhruti Shah, Xianzhi Du, Bowen Zhang, Yanghao Li, et al. Mm1. 5: Methods, analysis & insights from multimodal llm fine-tuning. *arXiv preprint arXiv:2409.20566*, 2024. 1, 2, 6

[46] Jingyi Zhang, Jiaxing Huang, Sheng Jin, and Shijian Lu. Vision-language models for vision tasks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[47] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer, 2024. 6

[48] Ruohong Zhang, Bowen Zhang, Yanghao Li, Haotian Zhang, Zhiqing Sun, Zhe Gan, Yinfei Yang, Ruoming Pang, and Yiming Yang. Improve vision language model chain-of-thought reasoning. *arXiv preprint arXiv:2410.16198*, 2024. 1, 3, 6

[49] Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. Pmc-vqa: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023. 2

[50] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*, 2024. 6