First SFT, Second RL, Third UPT: Continual Improving Multi-Modal LLM Reasoning via Unsupervised Post-Training

Lai Wei[1,2,*] Yuting Li[1] Chen Wang[2] Yue Wang[2] Linghe Kong[1]
Weiran Huang[1,3,†] Lichao Sun[4]

[1] School of Computer Science, Shanghai Jiao Tong University
[2] Zhongguancun Academy
[3] Shanghai Innovation Institute [4] Lehigh University

Abstract

Improving Multi-modal Large Language Models (MLLMs) in the post-training stage typically relies on supervised fine-tuning (SFT) or reinforcement learning (RL), which require expensive and manually annotated multi-modal data–an ultimately unsustainable resource. This limitation has motivated a growing interest in unsupervised paradigms as a third stage of post-training after SFT and RL. While recent efforts have explored this direction, their methods are complex and difficult to iterate. To address this, we propose MM-UPT, a simple yet effective framework for unsupervised post-training of MLLMs, enabling continual self-improvement without any external supervision. The training method of MM-UPT builds upon GRPO, replacing traditional reward signals with a self-rewarding mechanism based on majority voting over multiple sampled responses. Our experiments demonstrate that such training method effectively improves the reasoning ability of Qwen2.5-VL-7B (e.g., 66.3%→72.9% on MathVista, 62.9%→68.7% on We-Math), using standard dataset without ground truth labels. To further explore scalability, we extend our framework to a data self-generation setting, designing two strategies that prompt the MLLM to synthesize new training samples on its own. Additional experiments show that combining these synthetic data with the unsupervised training method can also boost performance, highlighting a promising approach for scalable self-improvement. Overall, MM-UPT offers a new paradigm for autonomous enhancement of MLLMs, serving as a critical third step after initial SFT and RL in the absence of external supervision. Our code is available at https://github.com/waltonfuture/MM-UPT.

1 Introduction

Multi-modal Large Language Models (MLLMs) have achieved remarkable performance on a variety of vision-language tasks, ranging from image captioning to visual reasoning [22, 51, 67, 73, 81, 83]. The dominant paradigm for improving MLLMs in the post-training stage typically involves supervised fine-tuning (SFT) and reinforcement learning (RL) [2, 42, 47, 59, 65, 77]. However, both SFT and RL rely on large volumes of high-quality and annotated multi-modal data, such as image captions, visual reasoning traces, verifiable ground truth answers, and human preference signals. As real-world tasks

[*] Email: waltonfuture@sjtu.edu.cn

[†] Correspondence to Weiran Huang (weiran.huang@outlook.com).

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

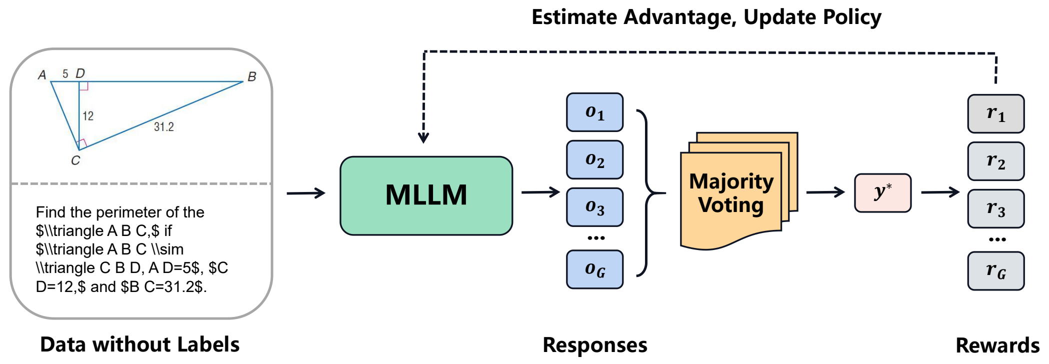

Estimate Advantage, Update Policy
A
5
D
B
12
31.2
C
Find the perimeter of the $\\triangle A B C,$ if $\\triangle A B C \\sim \\triangle C B D, A D=5$, $C D=12,$ and $B C=31.2$.
MLLM
o_1
o_2
o_3
...
o_G
Majority Voting
y^*
r_1
r_2
r_3
...
r_G
Data without Labels
Responses
Rewards

Figure 1: Overview of the MM-UPT framework. Given an unlabeled multi-modal input, the MLLM samples multiple responses, and uses majority voting to determine the pseudo-label. The MLLM is then updated via GRPO, enabling self-improvement without external supervision.

grow in complexity and quantity, a critical challenge emerges: curating and annotating high-quality data at scale becomes increasingly impractical. To overcome this data-dependency, a paradigm shift is required towards a third stage of post-training beyond SFT and RL, dedicated to the continual self-improvement of MLLMs through synthetic and unlabeled data. We formalize this third-stage paradigm as *Unsupervised Post-Training*.

Previous works have studied the use of MLLMs themselves to generate synthetic instruction data for self-improvement through offline training techniques like SFT and DPO [6, 23, 39–41, 64, 74]. These approaches typically involve complex pipelines with multiple stages, such as data generation, verification, and filtering, which are hard to iterate online. Fortunately, recent studies demonstrate notable success using online reinforcement learning (e.g. GRPO [37]) with verifiable rewards to enhance the reasoning capabilities of MLLMs [7, 31, 53]. A concurrent work, TTRL [84], further extends this line by applying GRPO on test-time scaling of LLMs. It is promising that online RL enables models to continuously improve, thus acquiring novel reasoning abilities that exceed corresponding base models' capacity.

Motivated by these insights, we propose MM-UPT (Multi-Modal Unsupervised Post-Training), an easy-to-implement framework for unsupervised post-training in MLLMs. As illustrated in Figure 1, our approach adapts the online reinforcement learning method GRPO [37], which is known for its stability and scalability. The core challenge in applying GRPO to an unsupervised setting is the absence of ground-truth labels for reward calculation. To overcome this, MM-UPT works by deriving implicit reward signals via majority voting over multiple sampled responses. In particular, majority voting aggregates multiple responses and selects the most frequent one, which has been widely used and shown effective to improve model performance [44, 49, 84]. Thus, we adopt the majority-voted answer to serves as a dynamic pseudo-label in GRPO: responses that align with this consensus receive a positive reward, while divergent ones are penalized. This process effectively encourages the model to bootstrap its own high-confidence knowledge, promoting the generation of stable and consistent answers without relying on any external supervision or reward models.

Beyond learning from existing unlabeled data, MM-UPT further extends to a data self-generation setting that enhances scalability under the unsupervised paradigm. Specifically, we design two strategies that allow the MLLM itself to synthesize new training samples: (1) *in-context synthesizing*, where the model generates new questions conditioned on original examples (image, question, and answer); and (2) *direct synthesizing*, where the model generates questions solely from the given image. These self-generated samples are then used within the same unsupervised reinforcement learning method, enabling continual self-improvement even in the absence of human-created questions.

In our experiments, we focus on the domain of multi-modal reasoning, which is widely focused and inherently challenging. We explore two key scenarios for constructing unlabeled data assuming that labels are not available: (1) using human-created questions without ground-truth labels, and (2) employing synthetic questions generated by the model itself, inherently lacking ground-truth labels. This setup allows us to examine both the effectiveness of unsupervised training on existing data and self-generated data in enhancing reasoning performance. We evaluate MM-UPT across a range of reasoning benchmarks and observe notable performance improvements over the base models (e.g., 66.3%→72.9% on MathVista, 62.9%→68.7% on We-Math using Qwen2.5-VL-7B [1]) in the first scenario. Our method also outperforms previous baseline methods, and is even competitive with supervised GRPO, underscoring the effectiveness of MM-UPT as a self-improving training strategy. As for the second scenario, we find that models trained on unlabeled synthetic data achieve performance competitive with those trained on the original unlabeled dataset, revealing a viable path for scalable self-improvement. Additionally, our deeper analysis reveals a clear trade-off in MM-UPT: it improves accuracy by reinforcing high-confidence knowledge but reduces response diversity and requires sufficient initial competence to prevent error amplification.

Our main contributions are summarized as follows:

- We are the first to formalize a three-stage post-training paradigm for MLLMs, with Unsupervised Post-Training (UPT) as the critical third stage enabling continual improvement without external supervision. We instantiate this stage with MM-UPT, a simple and effective framework using majority voting as a self-rewarding signal in online reinforcement learning.
- Extensive experiments on multi-modal reasoning tasks demonstrate the effectiveness of majority voting as a pseudo-reward estimation for unsupervised training.
- We extend MM-UPT to utilize synthetic data generated by the MLLM itself, and find that training the MLLM on such data leads to notable performance gains. This reveals a promising path toward efficient and scalable self-improvement in unsupervised post-training.

2 Related Works

Self Improvement. High-quality data obtained from human annotations has been shown to significantly boost the performance of LLMs across a wide range of tasks [12, 18, 32]. However, such high-quality annotated data may be exhausted in the future. This presents a substantial obstacle to the continual learning of advanced models. As a result, recent research has shifted toward self-improvement, leveraging data generated by the LLM itself without any external supervision [8, 16, 29, 54, 84]. Several following works also explore self-improvement in the multi-modal domain [6, 11, 39, 64, 72, 74]. Genixer [74] firstly introduces a self-improvement pipeline including complex data generation and filtering for SFT. STIC [6] and SENA [40] construct preference data pairs for DPO [36] in a self-supervised manner, focusing on enhancing perceptual capabilities. In contrast to these approaches which are complex and hard to scale, the key distinction is that our work leverages online reinforcement learning using GRPO [12] with simple yet effective data synthesizing strategies at the post-training stage, which can be more scalable for self-improvement without reliance on any external supervision. In addition, none of these previous methods focus on multi-modal reasoning tasks, which are considered challenging for current models.

Multi-modal Reasoning. Recently, the reasoning abilities of MLLMs have become a central focus of research [27, 53, 56, 82]. In contrast to traditional LLM-based reasoning [12, 28, 62] that primarily relies on text, multi-modal approaches must both process and interpret visual inputs, significantly increasing the complexity of tasks such as geometric problem-solving and chart interpretation [3, 30, 75]. Several works in this field have sought to collect or synthesize a large scale of multi-modal reasoning data [5, 33, 38, 70]. Notably, the recent emergence of o1-like reasoning models [19] represents an initial step toward activating the slow-thinking capabilities of MLLMs, as demonstrated by several SFT-based methods, such as LLaVA-CoT [55], MAmmoTH-VL [13], and Mulberry [60]. Moreover, some concurrent works have further explored reinforcement learning approaches, particularly GRPO [37], in the post-training stage of MLLMs to enhance performance on multi-modal reasoning tasks [7, 31, 34, 48, 78]. While these supervised post-training methods have demonstrated promising results, our work explores a different direction by focusing on totally unsupervised post-training of MLLMs to self-improve the reasoning abilities.

3 The Framework of Multi-Modal Unsupervised Post-Training

Existing post-training techniques for MLLMs, such as supervised fine-tuning (SFT) and reinforcement learning (RLHF or RLVR), rely heavily on labeled data or external reward models. While these approaches have proven effective, their reliance on external supervision makes continual improvement unsustainable due to the high cost and limited scalability of manual annotation. To overcome this limitation, we formalize a new third-stage post-training paradigm that enables the model to self-improve without access to any external supervision, such as ground-truth labels or additional reward models. We instantiate this paradigm with **MM-UPT** (Multi-Modal Unsupervised Post-Training), a simple yet effective framework designed to operate purely on unlabeled multi-modal data. The overview of our complete framework is shown in Figure 1.

3.1 Problem Formulation

Firstly, we formulate the problem of unsupervised post-training for MLLMs as follows: Given a well-trained multi-modal LLM π_θ and a collection of unlabeled multi-modal data $Q = \{(I_i, q_i)\}_{i=1}^N$, where I_i represents an image and q_i denotes a corresponding question, our goal is to improve the model's performance without access to any ground-truth answers or external supervision signals. This setting differs significantly from conventional supervised fine-tuning (SFT), reinforcement learning with verifiable rewards (RLVR), or reinforcement learning with human feedback (RLHF), which typically rely on labeled data (I_i, q_i, y_i) or human preference data (I_i, q_i, y_i^+, y_i^-), where y_i denotes the answer of q_i and (y_i^+, y_i^-) denotes the preference pair of q_i. In contrast, we only allow to operate in a fully unsupervised manner for this setting, leveraging only the model's own responses to generate training signals. This presents significant challenges, as the model must learn to assess and improve its own outputs without any external guidance.

3.2 Training Method

To achieve the unsupervised training, MM-UPT introduces a self-rewarding mechanism using majority voting as pseudo-labels [49] based on the online reinforcement learning. In particular, MM-UPT is built upon the GRPO algorithm [37], which is widely used in the post-training stage of multi-modal LLMs. GRPO optimizes computational efficiency by eliminating the need for a separate value model; instead, it directly utilizes group-normalized rewards to estimate advantages. Specifically, for a question q and the correlated image I from the training dataset Q, GRPO samples a group of responses $O = \{o_i\}_{i=1}^G$ from the old policy π_{old} and then optimizes the policy model by maximizing the following objective:

$$\mathcal{J}(\theta) = \mathbb{E}_{(q,I)\sim Q, \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q,I)}$$
$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{ \min\left[\gamma_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(\gamma_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right] - \beta \mathrm{D}_{KL}\left[\pi_\theta \| \pi_{ref}\right]\right\},$$

where $\gamma_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}$, π_{ref} represents the reference model, and the term D_{KL} introduces a KL divergence constraint to limit how much the model can deviate from this reference. The advantage estimate $\hat{A}_i$ measures how much better the response o_i is compared to the average response, which is computed using a group of rewards $\{r_1, r_2, \dots, r_G\}$ for the responses in set O: $\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}$.

In the above standard GRPO formulation [12], the reward is computed in a supervised manner based on labels for each response in $O = \{o_i\}_{i=1}^G$. Shifting towards our unsupervised setting, where no ground-truth labels are available, one feasible way is to construct pseudo-labels to calculate the reward for GRPO. Motivated by [16, 49, 84], we use majority voting over the group of sampled responses O to serve as pseudo-labels. Majority voting selects the most frequent answer among the sampled responses O and has proven to be a simple yet effective technique [49, 84], making it suitable for deriving good pseudo-reward signals. Specifically, we first extract answers from the responses $O = \{o_i\}_{i=1}^G$ using an rule-based answer extractor [15] $E(\cdot)$, resulting in $\hat{Y} = E(O) = \{\hat{y}_i\}_{i=1}^G$. Then, the majority-voted answer y^* can be obtained by:

$$y^* = \arg\max_{y \in \hat{Y}} \sum_{i=1}^{G} \mathbb{I}[y = \hat{y}_i], \tag{1}$$

where $\mathbb{I}[\cdot]$ is the indicator function. The reward r_i is then determined based on the y^*:

$$r_i = \begin{cases} 1, & \text{if } \hat{y}_i = y^*, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Algorithm 1 The training method of MM-UPT

```
**Input:** Current policy $\pi_\theta$, old policy $\pi_{\theta_{old}}$, unlabeled training dataset $Q$, Group size $G$, reference model $\pi_{ref}$, clip parameter $\epsilon$, KL penalty coefficient $\beta$, answer extractor $E(\cdot)$.
**for** each sample $(I, q) \sim Q$ **do**
    Sample group of responses: $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(o \mid I, q)$;    // Sample multiple responses
    Extract answers: $\hat{Y} = E(O) = \{\hat{y}_i\}_{i=1}^{G}$;
    Determine majority vote: $y^* \leftarrow \arg\max_{y \in \hat{Y}} \sum_{i=1}^{G} \mathbb{I}[y = \hat{y}_i]$;    // Select the most frequent answer
    Compute pseudo-rewards: $r_i \leftarrow \mathbb{I}[\hat{y}_i = y^*]$;    // Reward based on majority agreement
    Compute advantage estimates: $\hat{A}_i \leftarrow \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}$;
    Compute GRPO objective:
     $\mathcal{J}(\theta) \leftarrow \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \left\{ \min \left[ \gamma_{i,t}(\theta) \hat{A}_i, \text{clip}\left(\gamma_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_i \right] - \beta \mathrm{D}_{KL}[\pi_\theta \| \pi_{ref}] \right\}$
     where $\gamma_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} | I, q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} | I, q, o_{i,<t})}$;
    Update policy parameters: $\theta \leftarrow \theta - \nabla_\theta \mathcal{J}_{GRPO}(\theta)$;
    Update old policy: $\theta_{\text{old}} \leftarrow \theta$;
**end for**
**return** $\pi_\theta$
```

In this way, we compute pseudo-rewards via majority voting and apply standard GRPO to update the MLLM. This majority-based reward encourages the model to converge toward consistent, high-consensus responses, thereby enabling the model to further exploit its existing self-knowledge leveraging unlabeled data. The pipeline of our training method in MM-UPT is shown in Algorithm 1.

3.3 Synthetic Data

To further explore the scalability of unsupervised post-training, we extend our framework to a data self-generation setting, where the model is asked to synthesize new training samples on its own. In particular, we design two simple yet effective data synthesizing strategies as follows.

In-Context Synthesizing. Inspired by Self-Instruct [50], we construct a data generation pipeline that leverages in-context examples to guide the synthesis process. Each original example consists of an image, a question, and its corresponding answer. To synthesize new samples, we provide the model with the full triplet and instruct it to generate a new question that is semantically distinct from the original but relevant to the same image. This strategy helps generate task-relevant and meaningful variations of the original question, as well as ensure the quality of synthetic questions. During unsupervised post-training, the model then attempts to answer each newly generated question, and pseudo-labels are derived from the majority vote among its sampled responses, consistent with the mechanism described in Section 3.

Direct Synthesizing. In addition to in-context generation, we also explore a *direct synthesizing* strategy that further increases diversity. Here, the model receives only the image and is prompted to freely create a new question without any reference to the original question. This open-ended formulation encourages the model to generate a wider range of diverse and novel questions based solely on the visual input, rather than being constrained by the original task. Similar to the in-context setup, we perform unsupervised post-training on these synthetic samples using majority voting to define pseudo-rewards.

This setup enables the MLLM to expand the training corpus without any human annotations, thus achieving a fully autonomous self-improvement loop.

4 Experiments

We conduct extensive experiments to evaluate the effectiveness of MM-UPT across various multi-modal LLMs, datasets, and benchmarks. Our experiments are designed to explore **two key scenarios**: (1) using human-created questions without ground-truth labels (Section 4.2), and (2) employing synthetic questions generated by the model itself, inherently lacking ground-truth labels (Section 4.3). Before presenting the experimental results, we first outline the baseline methods, evaluation benchmarks, and implementation details in the experimental setup as follows.

Table 1: Main results of **Scenario 1** on four multi-modal mathematical reasoning benchmarks. We report accuracy (%) for each method on MathVision, MathVerse, MathVista, and We-Math. All methods are conducted on the Qwen2.5-VL-7B backbone. MM-UPT outperforms other baseline methods, and is even competitive with supervised methods.

Model and Methods	Unsupervised?	Training Data	MathVision	MathVerse	MathVista	We-Math	Avg
Qwen2.5-VL-7B	-	-	24.87	43.83	66.30	62.87	49.47
+ **GRPO** [37]	✗	Geometry3K	28.32	46.40	69.30	68.85	53.22
+ **GRPO** [37]	✗	GeoQA	26.15	46.28	67.50	66.65	51.65
+ **GRPO** [37]	✗	MMR1	29.01	45.03	71.40	67.24	53.17
+ **SFT** [43]	✗	Geometry3K	25.92	43.73	67.90	64.94	50.63
+ **SFT** [43]	✗	GeoQA	25.72	44.70	67.40	65.10	50.73
+ **SFT** [43]	✗	MMR1	26.45	43.53	63.30	64.20	49.37
+ **SRLM** [54]	✓	Geometry3K	26.94	44.54	66.90	66.32	51.18
+ **SRLM** [54]	✓	GeoQA	25.16	44.62	66.30	65.00	50.27
+ **SRLM** [54]	✓	MMR1	25.33	45.08	67.00	64.66	50.52
+ **LMSI** [16]	✓	Geometry3K	25.10	43.96	65.50	64.43	49.75
+ **LMSI** [16]	✓	GeoQA	25.49	43.50	66.60	63.51	49.78
+ **LMSI** [16]	✓	MMR1	24.83	43.76	64.90	66.38	49.97
+ **Genixer** [74]	✓	Geometry3K	26.02	43.15	65.50	62.18	49.22
+ **Genixer** [74]	✓	GeoQA	25.30	44.11	66.80	64.25	50.12
+ **Genixer** [74]	✓	MMR1	23.68	43.30	65.50	64.66	49.29
+ **STIC** [6]	✓	Geometry3K	25.39	42.92	65.20	62.99	49.13
+ **STIC** [6]	✓	GeoQA	23.49	42.87	64.30	63.62	48.57
+ **STIC** [6]	✓	MMR1	23.78	42.72	66.10	63.74	49.09
+ **MM-UPT (Ours)**	✓	Geometry3K	27.33	42.46	68.50	66.61	**51.23**
+ **MM-UPT (Ours)**	✓	GeoQA	27.07	43.68	68.90	68.22	**51.97**
+ **MM-UPT (Ours)**	✓	MMR1	26.15	44.87	72.90	68.74	**53.17**

4.1 Experimental Setup

Baseline Methods. Several prior works have explored self-improvement in both LLMs and MLLMs. Note that we focus on unsupervised self-improvement, we do not compare with methods that rely on external models (e.g., GPT-4o [18]) for supervision [17, 64, 71, 79, 80]. Instead, we compare with several totally unsupervised methods: LMSI [16], SRLM [54], Genixer [74], and STIC [6]. In particular, LMSI corresponds to supervised fine-tuning with self-generated content selected by majority voting. SRLM uses the model itself as LLM-as-a-Judge [76] to provide its own rewards during DPO [36] training. Genixer prompts the MLLM to first self-generate an answer and then self-check it. STIC applies DPO where original images and good prompts are used to generate preferred answers, and corrupted images and bad prompts to produce rejected answers. Additionally, we also compare with GRPO [37] and rejection SFT [43], which are two strong supervised methods. The details of these baseline methods are shown in Appendix A.1.

Benchmarks. We evaluate our method on four popular multi-modal mathematical reasoning benchmarks: MathVision [45], MathVista [27], MathVerse [69], and We-Math [35]. These benchmarks offer comprehensive evaluations with diverse problem types, including geometry, charts, and tables, featuring multi-subject and meticulously categorized visual math challenges across various knowledge concepts and granularity levels. We provide more details in Appendix A.2.

Implementation Details. We adopt the EasyR1 [61] framework for multi-modal unsupervised post-training, which is based on GRPO. Specifically, we set the training episodes to 15, and use AdamW optimizer [24] with a learning rate of 1×10^{-6}, weight decay of 1×10^{-2}, and gradient clipping at a maximum norm of 1.0. The KL divergence constraint β in GRPO is set to 0.01 to stabilize the training. The vision tower of the multi-modal model is also tuned without freezing. In our training, we use a rollout temperature of 0.7, which strikes a good balance: lower temperatures produce low-diversity outputs, while higher temperatures often lead to lower-quality outputs. Within each episode, we perform one rollout group (G=10) per data point. Other hyperparameters follow the default settings provided in the EasyR1 framework.

Table 2: Performance comparison of MM-UPT using different synthetic data generation strategies in **Scenario 2**. Both "In-Context Synthesizing" and "Direct Synthesizing" approaches yield significant improvements over the base model and perform competitively with the "Original Questions" on average, demonstrating the effectiveness of synthetic data for unsupervised self-improvement.

Model and Methods	Dataset	MathVision	MathVerse	MathVista	We-Math	Avg
Qwen2.5-VL-7B	–	24.87	43.83	66.30	62.87	49.47
w/ Original Questions	Geo3K	27.33	42.46	68.50	66.61	51.23 (**3.6%**↑)
w/ In-Context Synthesizing	Geo3K	26.71	41.24	68.30	67.76	51.00 (**3.1%**↑)
w/ Direct Synthesizing	Geo3K	26.88	43.53	69.90	68.97	52.32 (**5.8%**↑)
w/ Original Questions	GeoQA	27.07	43.68	68.90	68.22	51.97 (**5.1%**↑)
w/ In-Context Synthesizing	GeoQA	26.09	42.87	70.60	69.25	52.20 (**5.5%**↑)
w/ Direct Synthesizing	GeoQA	26.25	44.64	71.50	68.28	52.67 (**6.5%**↑)
w/ Original Questions	MMR1	26.15	44.87	72.90	68.74	53.17 (**7.5%**↑)
w/ In-Context Synthesizing	MMR1	26.15	45.10	71.90	68.62	52.94 (**7.0%**↑)
w/ Direct Synthesizing	MMR1	26.15	44.11	70.40	67.99	52.16 (**5.4%**↑)

4.2 Scenario 1: Unsupervised Training on Standard Datasets

For our experiments, we firstly employ standard training datasets with masked labels to simulate the first scenario (i.e., using human-created questions without ground-truth answers). We conduct MM-UPT on Geometry3k [25], GeoQA [4], and MMR1 [20] using the Qwen2.5-VL-7B [1] model. These datasets cover a diverse set of visual math problems, including geometric diagrams, charts, and structured question formats (multiple-choice and fill-in-the-blank), serving as a strong foundation for models to self-improve the multi-modal mathematical reasoning abilities. More details of these datasets are introduced in Appendix A.3.

Table 1 presents the main results on four challenging multi-modal mathematical reasoning benchmarks. We observe that MM-UPT achieves consistent improvements in average over the base Qwen2.5-VL-7B model across all datasets, also outperforming other baseline methods such as SRLM, LMSI, Genixer, and STIC. Notably, MM-UPT is able to improve the average score from 49.47 (base model) to 53.17 (with MMR1 dataset), demonstrating its effectiveness in leveraging unlabeled data for self-improvement. In comparison, previous baselines provide only marginal gains or even degrade performance on certain benchmarks, highlighting the limitations of existing methods when applied to already strong models in multi-modal reasoning tasks. Furthermore, we find that MM-UPT is even competitive with supervised post-training methods, such as rejection sampling-based SFT [43] and GRPO [37]. These results underscore the potential of MM-UPT to further exploit the knowledge embedded in multi-modal models for self-improvement. Further experiments on the training dynamics, generalization and adaptability of our method are shown in Appendix B.

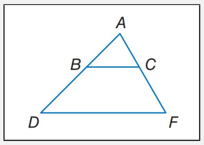
4.3 Scenario 2: Unsupervised Training on Synthetic Datasets

To further explore the potential of MM-UPT, we investigate the use of unlabeled *synthetic data* (mentioned in Section 3.3) to improve MLLMs. This aligns with the ultimate goal of MM-UPT: enabling continual self-improvement even after human-created data is exhausted.

In our experiment, we use the previous two methods in Section 3.3 to generate the synthetic data, leveraging Geometry3K [25], GeoQA [4], and MMR1 [20] as the seed datasets, and Qwen2.5-VL-7B as the base MLLM for data synthesis. MM-UPT is then applied to the same base model (i.e., Qwen2.5-VL-7B) using each of these different synthetic datasets separately. Table 2 presents experimental results using different synthetic data generation strategies. Both in-context and direct synthesizing lead to significant improvements over the base model, achieving performance comparable to training on original human-written questions. This shows that synthetic questions can effectively enhance the model's reasoning ability under MM-UPT. Notably, direct synthesizing even surpasses human-written questions (when applied to Geometry3K and GeoQA) on average, demonstrating the strong ability of the model to generate high-quality textual questions solely based on images. This highlights the potential for scalable and fully autonomous self-improvement in multi-modal domain via visual-centric data synthesis.

Moreover, we manually examine some synthetic data. We observe that in-context synthesizing often produces questions similar to the original ones by substituting conditions or expressions, resembling data rephrasing. In contrast, direct synthesizing generates more diverse and novel questions. While some of the directly synthesized questions still contain hallucinations, many are of high quality and beneficial for unsupervised post-training. This underscores the potential of the direct synthesizing approach as a simple yet effective method for data generation, without the need for textual in-context examples. Below, we present two illustrative examples that showcase the effectiveness and quality of synthetic questions generated through both approaches.

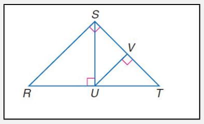
Demo: Examples of synthetic data using different strategies.



Original Question:
$BC \| DF$. $AB = x + 5$, $BD = 12$, $AC = 3x + 1$, and $CF = 15$. Find x.

In-Context Synthetic Question:
Given that $BC \| DF$, $AB = 2x - 3$, $BD = 18$, $AC = x + 7$, and $CF = 24$. Find the value of x.

Directly Synthetic Question:
In the given triangle $\triangle ADF$, point B lies on AD and point C lies on AF. If $BC \parallel DF$, what is the ratio of the area of $\triangle ABC$ to the area of $\triangle ADF$?



Original Question:
If $\angle RST$ is a right angle, $SU \perp RT$, $UV \perp ST$, and $\angle RTS = 47^\circ$, find $\angle RSU$

In-Context Synthetic Question:
If $\angle RST$ is a right angle, $SU \perp RT$, $UV \perp ST$, and $\angle RTS = 47^\circ$, find $\angle VST$.

Directly Synthetic Question:
In the given triangle $\triangle RST$, point U lies on RT such that SU is perpendicular to RT. Point V lies on ST such that UV is perpendicular to ST. If $RU = 12$ units, $UT = 16$ units, and $SV = 9$ units, find the length of VT.

4.4 Ablation Study

To evaluate the generality and effectiveness of MM-UPT, we conduct an ablation study across a range of backbone models beyond the primary Qwen2.5-VL-7B [1]. Specifically, we apply MM-UPT to several state-of-the-art models of varying scales, including Qwen2.5-VL-3B [1], MM-Eureka-7B [31], and ThinkLite-VL-7B [48]. All models are post-trained using MM-UPT on the Geometry3K dataset [25], without access to any labels (i.e., Scenario 1). As summarized in Table 3, MM-UPT consistently improves the performance of all tested models on average, despite the absence of supervision during post-training. Notably, ThinkLite-VL-7B with MM-UPT achieves the highest average score (54.07), and shows substantial gains on the MathVista [27] benchmark, reaching a score of 74.70. In addition, Qwen2.5-VL-3B, the smallest model in our study, also benefits well from MM-UPT (+7.4% on average), demonstrating the robustness and adaptability of MM-UPT for performance enhancement. These results collectively reveal that MM-UPT can be easily applied to various multi-modal models to enable consistent self-improvement.

Moreover, our results also show that MM-UPT is compatible with supervised GRPO. For instance, MM-Eureka-7B was already tuned with supervised GRPO on the K12 dataset, yet applying MM-UPT on a new unlabeled dataset (Geometry3K) further improved its average score from 53.10 to 53.78 (Table 3). Similarly, ThinkLite-VL-7B was trained with supervised GRPO on the ThinkLite dataset, and applying MM-UPT again boosted its average score to 54.07, including a substantial gain on MathVista (74.70). These results highlight the practical value of MM-UPT as a lightweight refinement step that remains effective even for models already optimized with supervised GRPO, enabling them to leverage new unlabeled data for further improvement.

5 Deeper Analysis

Going beyond standard benchmarking, we conduct a deeper analysis to investigate MM-UPT's performance boundaries (Section 5.1 and Section 5.2) and tradeoffs (Section 5.3). This helps better understand its behavior and potential applications.

Table 3: Ablation study using different models besides Qwen2.5-VL-7B. We conduct this experiment on Geometry3K [25] dataset without labels.

Models	**MathVision**	**MathVerse**	**MathVista**	**We-Math**	**Avg**
Qwen2.5-VL-7B	24.87	43.83	66.30	62.87	49.47
Qwen2.5-VL-7B + MM-UPT	27.33	42.46	68.50	66.61	51.23 (**3.6%**↑)
MM-Eureka-7B	28.06	50.46	69.40	64.48	53.10
MM-Eureka-7B + MM-UPT	28.95	50.63	69.10	66.44	53.78 (**1.3%**↑)
ThinkLite-VL-7B	26.94	46.58	69.00	67.99	52.63
ThinkLite-VL-7B + MM-UPT	26.91	47.26	74.70	67.41	54.07 (**2.8%**↑)
Qwen2.5-VL-3B	19.47	33.58	56.30	50.63	39.00
Qwen2.5-VL-3B + MM-UPT	22.17	32.39	57.10	55.22	41.72 (**7.4%**↑)

Table 4: Performance of MM-UPT on the difficult ThinkLite-11K dataset. Results show that MM-UPT leads to a decrease in performance when applied to a dataset where the model has limited prior knowledge, highlighting the limitations of majority voting in such scenarios.

Models	**Training Data**	**MathVision**	**MathVerse**	**MathVista**	**We-Math**	**Avg**
Qwen2.5-VL-7B	–	24.87	43.83	66.30	62.87	49.47
Qwen2.5-VL-7B + MM-UPT	ThinkLite-11K	21.12	37.10	59.20	59.02	44.11

5.1 Why Does MM-UPT Work?

Majority voting [49] is a fundamental ensemble technique that enhances prediction reliability by aggregating multiple independent responses. In our framework, it offers a simple yet powerful pseudo-reward signal to help model self-improve, particularly when the model are moderately reliable on the unlabeled datasets. We consider a simplified explanation for it using a classical toy example. Suppose that each response hits the correct answer with probability $p > 0.5$ in a binary question. Then, we sample the model's response n times independently. The final answer is determined by a majority vote, that is, the answer that appears more than $n/2$ times. Let X denote the number of correct predictions among the n samples. Since each prediction is correct with probability p, X follows a binomial distribution: $X \sim \text{Binomial}(n, p)$. The majority vote is correct if $X > n/2$, and the corresponding probability of this event (denoted as E) is:

$$P(E) = \sum_{i=\lceil n/2 \rceil}^{n} \binom{n}{i} p^i (1-p)^{n-i}.$$

When $p > 0.5$, it follows that $P(E) > p$, which means that the ensemble outperforms each individual response. For instance, if $p = 0.7$ and $n = 10$, then $P(E) \approx 0.849$, demonstrating a significant gain over the base accuracy. This analysis reveals the rationality of majority voting to serve as the pseudo-label for deriving reliable reward signal in the unsupervised setting. In our experimental setting, we mainly target datasets that are not especially hard, such as Geometry3k [25], GeoQA [4], and MMR1 [20], for unsupervised post-training. Hence, we hypothesize that the model has a relatively high chance of answering questions in these datasets correctly. This allows the model to yield stable improvements through MM-UPT using majority voting as the pseudo-label.

5.2 When Might MM-UPT Fail?

According to the analysis in Section 5.1, it reveals that the effectiveness of MM-UPT diminishes when the model lacks sufficient prior knowledge of the target dataset. To show that, we apply MM-UPT to ThinkLite-11K [48] dataset using Qwen2.5-VL-7B [1]. ThinkLite-11K is collected via difficulty-aware sampling that only retains samples that the model rarely answers correctly. Thus, this setting reflects a scenario where the model is more likely to be wrong than right. In such cases, majority voting amplifies incorrect answers rather than filtering them, leading to degraded performance. As shown in Table 4, applying MM-UPT to ThinkLite-11K results in a significant drop in accuracy across all benchmarks. This suggests that majority voting fails to provide reliable reward signals when

Table 5: Comparison of pass@10 across different benchmarks. Results show that MM-UPT improves single-response accuracy (pass@1, Table 1) but reduces response diversity, leading to a drop in pass@10. Supervised GRPO alleviates this issue to some extent.

Model	MathVision	MathVerse	MathVista	We-Math	Avg
Qwen2.5-VL-7B	0.6556	0.7307	0.8730	0.9420	0.8003
Qwen2.5-VL-7B + MM-UPT	0.5661	0.6477	0.8240	0.8621	0.7250
Qwen2.5-VL-7B + Supervised GRPO	0.6164	0.6726	0.8570	0.9075	0.7634

the model has limited prior understanding of the domain. To address this issue, alternative forms of algorithms using more fine-grained and complex rewarding methods, such as LLM-as-Judge [54, 76] and model collaboration [8, 21], may be necessary. Note that our work represents an initial attempt at self-improvement in MLLMs via GRPO, and we believe that these algorithms are complementary to our approach and could be integrated into our framework in the future.

5.3 Trade-Offs in MM-UPT

Although MM-UPT improves pass@1 accuracy across multiple benchmarks, we also observe a consistent decrease in pass@n (for large n, e.g., pass@10) performance in Table 5. This reflects a common trade-off between accuracy and diversity in reinforcement learning for multi-modal models. Similar findings have been reported in supervised GRPO [66] (also shown in Table 5), where models tend to collapse onto high-consensus reasoning patterns, thereby reducing output diversity. In MM-UPT, this effect is further amplified because the implicit reward signal from majority voting encourages the model to favor dominant responses, which may suppress minority answers that are occasionally correct. That is said, MM-UPT let the model reinforce problems it can already solve, thereby forgoing the opportunity to tackle more challenging ones and abandoning exploration. While pass@1 is typically the most relevant metric in real-world applications, mitigating the decline of pass@n remains an important open problem for future work.

Another key consideration is the trade-off between training and inference costs. A straightforward alternative to MM-UPT is to apply majority voting directly at inference time, which can also boost accuracy by aggregating multiple outputs. However, inference-time ensembling incurs substantial computational overhead, as n samples must be generated per query, making it expensive and often impractical at scale. By contrast, MM-UPT shifts this cost into a one-time training stage: after refinement, the model produces stronger single-pass outputs, avoiding the need for repeated sampling during deployment. This distinction highlights different usage scenarios: MM-UPT is particularly beneficial when inference efficiency and scalability are critical, whereas inference-time ensembling may be preferable when training resources are limited.

In summary, MM-UPT offers a simple unsupervised post-training framework but also entails clear trade-offs. Balancing accuracy and diversity, and choosing between training-time and inference-time costs, are crucial aspects to consider and represent promising directions for future exploration.

6 Conclusion

In this work, we formalize a third-stage post-training paradigm for multi-modal large language models after SFT and RL, termed *Unsupervised Post-Training (UPT)*. We instantiate this paradigm through **MM-UPT**, a simple yet effective framework that leverages majority voting as a self-rewarding mechanism within the GRPO algorithm, guiding models toward consistent and high-confidence responses on multi-modal reasoning tasks. This can be used to further exploit the model's internal knowledge by reinforcing consistent predictions, and thus enables self-improvement without any external supervision. Extensive experiments across multiple benchmarks demonstrate that this method effectively enhances the reasoning performance of strong MLLMs without relying on labeled data or external reward models. Furthermore, we extend the framework to a data self-generation setting, showing that synthetic questions produced by the model itself can further boost performance, revealing a scalable path toward autonomous self-improvement. Future work could explore other fine-grained methods to provide pseudo-reward signals based on our framework, and investigate the scaling laws of unsupervised post-training using synthetic data.

Acknowledgements and Disclosure of Funding

This project is supported by the National Natural Science Foundation of China (No. 62406192), Opening Project of the State Key Laboratory of General Artificial Intelligence (No. SKLAGI2024OP12), Tencent WeChat Rhino-Bird Focused Research Program, and Doubao LLM Fund.

References

[1] 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.

[2] Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yutong Dai, Philip Yu, and Lichao Sun. A survey of ai-generated content (aigc). *ACM Computing Surveys*, 57(5):1–38, 2025.

[3] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 513–523, 2021.

[4] Liang Chen, Lei Li, Haozhe Zhao, Yifan Song, and Vinci. R1-v: Reinforcing super generalization ability in vision-language models with less than $3. https://github.com/Deep-Agent/R1-V, 2025. Accessed: 2025-02-02.

[5] Kanzhi Cheng, Yantao Li, Fangzhi Xu, Jianbing Zhang, Hao Zhou, and Yang Liu. Vision-language models can self-improve reasoning via reflection. *arXiv preprint arXiv:2411.00855*, 2024.

[6] Yihe Deng, Pan Lu, Fan Yin, Ziniu Hu, Sheng Shen, Quanquan Gu, James Y Zou, Kai-Wei Chang, and Wei Wang. Enhancing large vision language models with self-training on image comprehension. *Advances in Neural Information Processing Systems*, 37:131369–131397, 2024.

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

[8] Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. Self-boosting large language models with synthetic preference data. *arXiv preprint arXiv:2410.06961*, 2024.

[9] Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 2024.

[10] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023.

[11] Shuhao Gu, Jialing Zhang, Siyuan Zhou, Kevin Yu, Zhaohu Xing, Liangdong Wang, Zhou Cao, Jintao Jia, Zhuoyi Zhang, Yixuan Wang, et al. Infinity-mm: Scaling multimodal performance with large-scale and high-quality instruction data. *arXiv preprint arXiv:2410.18558*, 2024.

[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.

[13] Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. *arXiv preprint arXiv:2412.05237*, 2024.

[14] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[15] hiyouga. Mathruler. https://github.com/hiyouga/MathRuler, 2025.

[16] Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. *arXiv preprint arXiv:2210.11610*, 2022.

[17] Xin Huang, Jing Bai, Yeqing Shen, Jia Wang, Zheng Ge, and Osamu Yoshie. Seeking the right question: Towards high-quality visual instruction generation. *openreview*, 2024.

[18] 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.

[19] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

[21] Zhenwen Liang, Ye Liu, Tong Niu, Xiangliang Zhang, Yingbo Zhou, and Semih Yavuz. Improving llm reasoning through scaling inference computation with collaborative verification. *arXiv preprint arXiv:2410.05318*, 2024.

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

[23] Yixin Liu, Yonghui Wu, Denghui Zhang, and Lichao Sun. Agentic autosurvey: Let llms survey llms. *arXiv preprint arXiv:2509.18661*, 2025.

[24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[25] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.

[26] Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. Iconqa: A new benchmark for abstract diagram understanding and visual language reasoning. *arXiv preprint arXiv:2110.13214*, 2021.

[27] 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.

[28] Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*, 2023.

[29] 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.

[30] 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.

[31] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

[32] OpenAI. Openai gpt-4.5 system card, 2025. URL https://cdn.openai.com/gpt-4-5-system-card-2272025.pdf.

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

[34] Yi Peng, Xiaokun Wang, Yichen Wei, Jiangbo Pei, Weijie Qiu, Ai Jian, Yunzhuo Hao, Jiachun Pan, Tianyidan Xie, Li Ge, et al. Skywork r1v: Pioneering multimodal reasoning with chain-of-thought. *arXiv preprint arXiv:2504.05599*, 2025.

[35] Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, et al. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*, 2024.

[36] 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, 2024.

[37] 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.

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

[39] Yucheng Shi, Quanzheng Li, Jin Sun, Xiang Li, and Ninghao Liu. Enhancing cognition and explainability of multimodal foundation models with self-synthesized data. *arXiv preprint arXiv:2502.14044*, 2025.

[40] Wentao Tan, Qiong Cao, Yibing Zhan, Chao Xue, and Changxing Ding. Beyond human data: Aligning multimodal large language models by iterative self-evolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025.

[41] Zhiquan Tan, Lai Wei, Jindong Wang, Xing Xie, and Weiran Huang. Can i understand what i create? self-knowledge evaluation of large language models. *arXiv preprint arXiv:2406.06140*, 2024.

[42] Guiyao Tie, Zeli Zhao, Dingjie Song, Fuyang Wei, Rong Zhou, Yurou Dai, Wen Yin, Zhejian Yang, Jiangyue Yan, Yao Su, et al. A survey on post-training of large language models. *arXiv e-prints*, pages arXiv–2503, 2025.

[43] Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. Dart-math: Difficulty-aware rejection tuning for mathematical problem-solving. *Advances in Neural Information Processing Systems*, 37:7821–7846, 2024.

[44] Fouad Trad and Ali Chehab. To ensemble or not: Assessing majority voting strategies for phishing detection with large language models. In *International Conference on Intelligent Systems and Pattern Recognition*, pages 158–173. Springer, 2024.

[45] 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.

[46] Xiaoxuan Wang, Yihe Deng, Mingyu Derek Ma, and Wei Wang. Entropy-based adaptive weighting for self-training. *arXiv preprint arXiv:2503.23913*, 2025.

[47] Xidong Wang, Dingjie Song, Shunian Chen, Chen Zhang, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture. *arXiv preprint arXiv:2409.02889*, 2024.

[48] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025.

[49] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[50] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[51] Lai Wei, Zihao Jiang, Weiran Huang, and Lichao Sun. Instructiongpt-4: A 200-instruction paradigm for fine-tuning minigpt-4. *arXiv preprint arXiv:2308.12067*, 2023.

[52] Lai Wei, Zhiquan Tan, Chenghai Li, Jindong Wang, and Weiran Huang. Diff-erank: A novel rank-based metric for evaluating large language models. *arXiv preprint arXiv:2401.17139*, 2024.

[53] Lai Wei, Yuting Li, Kaipeng Zheng, Chen Wang, Yue Wang, Linghe Kong, Lichao Sun, and Weiran Huang. Advancing multimodal reasoning via reinforcement learning with cold start. *arXiv preprint arXiv:2505.22334*, 2025.

[54] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594*, 2024.

[55] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024.

[56] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024.

[57] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[58] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

[59] Zhejian Yang, Yongchao Chen, Xueyang Zhou, Jiangyue Yan, Dingjie Song, Yinuo Liu, Yuting Li, Yu Zhang, Pan Zhou, Hechang Chen, et al. Agentic robot: A brain-inspired framework for vision-language-action models in embodied agents. *arXiv preprint arXiv:2505.23450*, 2025.

[60] 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.

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

[62] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.

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

[64] 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.

[65] Zhengqing Yuan, Yang Wang, Zhaoxu Li, Yanfang Ye, and Lichao Sun. Tinygpt-moe: Scaling multi-modal large language model via advanced vision encoder with mixture-of-experts. *Proceedings of Machine Learning Research*, 2024.

[66] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.

[67] Kai Zhang, Rong Zhou, Eashan Adhikarla, Zhiling Yan, Yixin Liu, Jun Yu, Zhengliang Liu, Xun Chen, Brian D Davison, Hui Ren, et al. A generalist vision–language foundation model for diverse biomedical tasks. *Nature Medicine*, 30(11):3129–3141, 2024.

[68] Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025.

[69] 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.

[70] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024.

[71] Wenqi Zhang, Zhenglin Cheng, Yuanyu He, Mengna Wang, Yongliang Shen, Zeqi Tan, Guiyang Hou, Mingqian He, Yanna Ma, Weiming Lu, et al. Multimodal self-instruct: Synthetic abstract image and visual reasoning instruction using language model. *arXiv preprint arXiv:2407.07053*, 2024.

[72] Xiaoying Zhang, Da Peng, Yipeng Zhang, Zonghao Guo, Chengyue Wu, Chi Chen, Wei Ke, Helen Meng, and Maosong Sun. Will pre-training ever end? a first step toward next-generation foundation mllms via self-improving systematic cognition. *arXiv e-prints*, pages arXiv–2503, 2025.

[73] Yu Zhang, Jinlong Ma, Yongshuai Hou, Xuefeng Bai, Kehai Chen, Yang Xiang, Jun Yu, and Min Zhang. Evaluating and steering modality preferences in multimodal large language model. *arXiv preprint arXiv:2505.20977*, 2025.

[74] Henry Hengyuan Zhao, Pan Zhou, and Mike Zheng Shou. Genixer: Empowering multimodal large language model as a powerful data generator. In *European Conference on Computer Vision*, pages 129–147. Springer, 2024.

[75] Haojie Zheng, Tianyang Xu, Hanchi Sun, Shu Pu, Ruoxi Chen, and Lichao Sun. Thinking before looking: Improving multimodal llm reasoning via mitigating visual hallucination. *arXiv preprint arXiv:2411.12591*, 2024.

[76] 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. *arXiv preprint arXiv:2306.05685*, 2023.

[77] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, et al. A comprehensive survey on pretrained foundation models: A history from bert to chatgpt. *International Journal of Machine Learning and Cybernetics*, pages 1–65, 2024.

[78] Hengguang Zhou, Xirui Li, Ruochen Wang, Minhao Cheng, Tianyi Zhou, and Cho-Jui Hsieh. R1-zero's "aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025.

[79] Shijie Zhou, Ruiyi Zhang, Yufan Zhou, and Changyou Chen. A high-quality text-rich image instruction tuning dataset via hybrid instruction generation. *arXiv preprint arXiv:2412.16364*, 2024.

[80] Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. Calibrated self-rewarding vision language models. *arXiv preprint arXiv:2405.14622*, 2024.

[81] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[82] Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. *arXiv preprint arXiv:2408.08640*, 2024.

[83] Fei Zuo, Kehai Chen, Yu Zhang, Zhengshan Xue, and Min Zhang. InImageTrans: Multimodal LLM-based text image machine translation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20256–20277, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1039. URL https://aclanthology.org/2025.findings-acl.1039/.

[84] Yuxin Zuo, Kaiyan Zhang, Shang Qu, Li Sheng, Xuekai Zhu, Biqing Qi, Youbang Sun, Ganqu Cui, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

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: We have mentioned in the abstract.

   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: We have mentioned limitations in the "experiment" part of our paper.

   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?

   Answer: [Yes]

Justification: We provide them in our experiment part.

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 use open-sourced models, which are easy to reproduce.

   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 codes in our supplemental material. The data and models we used are all open-sourced.

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 provide in our experimental settings.

   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: [No]

   Justification: Experiments on MLLMs are too expensive to run for many times.

   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.
   - 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 provide 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 follow NeurIPS Code of Ethics.

   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: There is no societal impact of the work performed.

    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: The paper poses no such risks

    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 follow the license and terms of use in our experiments.

    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.
    - 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: [Yes]

    Justification: The 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: The paper does not involve crowdsourcing nor research with human subjects.

    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: [Yes]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    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.

Appendix

A Implementation Details

We provide the implementation details of our experiments as follows.

A.1 Baselines

Here, we explain how we implement different baseline methods in comparison.

LMSI [16] employs the majority-voted response as the target for supervised fine-tuning (SFT). For each question, we generate multiple responses and retain the ones that lead to the majority answer for training.

SRLM [54] studies Self-Rewarding Language Models, where the model itself is used via LLM-as-a-Judge prompting to provide its own rewards during training. In particular, for each question, we generate multiple candidate responses and use the prompt provided in the original paper to have the MLLM score its own outputs. Among the responses, the one with the highest score is selected as the positive example, and the one with the lowest score as the negative example. These pairs are then used to construct preference datasets for Direct Preference Optimization (DPO) [36].

Genixer [74] introduces a comprehensive data generation pipeline consisting of four key steps: (i) instruction data collection, (ii) instruction template design, (iii) empowering MLLMs, and (iv) data generation and filtering. To adapt Genixer in our setting, we remove the first two steps because we already have instruction data. After that, we use Qwen2.5-VL as the backbone model to self-generate responses 16 times per question for each dataset. In the filtering stage, we use the prompt to let the model self-judge the responses following Genixer:

> Here is a question-answer pair. Is $\{Q : X_q, A : X_a\}$ true for this image? Please answer this question with Yes or No.

In addition, Genixer calculates the probability of predicting the "Yes" rather than prompt the model to directly output "Yes" or "No" as the filtering label:

$$P(Y_r|X_I, X_q, X_a) = \prod_{i}^{L} p(y_i|X_I, X_q, X_{a,<i}), \tag{3}$$

where Y_r is the predicted judge, X_I is the image, X_q is the question, X_a is the self-generated response, and L is the length the total predicted judge. Then, it proposes a threshold λ to control the filtering in the following manner:

$$S^n = \begin{cases} \text{True}, & \text{if } Y_r = \text{Yes and } P(Y_r^n) > \lambda \\ \text{False}, & \text{if } Y_r = \text{Yes and } P(Y_r^n) \leq \lambda \\ \text{False}, & \text{if } Y_r = \text{No} \end{cases}, \tag{4}$$

where S^n is the filter label representing keeping or removing the current sample. $P(Y_r^n)$ denotes the probability of the result "Yes" of n-th candidates. λ is set to 0.7 following the paper.

STIC [6] proposes a two-stage self-training algorithm focusing on the image comprehension capability of the MLLMs. In Stage 1, the base MLLM self-constructs its preference dataset for image description using well-designed prompts, poorly-designed prompts, and distorted images with diffusion noise. In Stage 2, a small portion of the previously used SFT data is recycled and infused with model-generated image descriptions to further fine-tune the base MLLM. In particular, since Qwen2.5-VL does not open-source the SFT data, we opt to use the model's self-generated responses sampled from different datasets to represent the previously used SFT data.

A.2 Benchmarks

We provide some details about the benchmarks we use to evaluate the models' reasoning ability. MathVision [45] is a challenging benchmark containing 3040 mathematical problems with visual contexts from real-world math competitions across 12 grades. It covers 16 subjects over 5 difficulty levels, including specialized topics like Analytic Geometry, Combinatorial Geometry, and Topology. MathVista [27] is a comprehensive benchmark for evaluating mathematical reasoning in visual contexts. It contains 1000 questions featuring diverse problem types including geometry, charts, and tables. MathVerse [69] is an all-around visual math benchmark designed for an equitable and in-depth evaluation of MLLMs. The test set contains 3940 multi-subject math problems with diagrams from publicly available sources, focusing on Plane Geometry and Solid Geometry. We-Math [35] meticulously collect and categorize 1740 visual math problems in the test set, spanning 67 hierarchical knowledge concepts and 5 layers of knowledge granularity.

For all benchmarks, we prompt the models to place their final answers within a designated box format. We then employ Qwen2.5-32B-Instruct [57] to evaluate answer correctness by comparing the extracted responses with ground truth answers, which often contain complex mathematical expressions. Note that our reported benchmark scores may differ from those in the original papers due to variations in evaluation protocols.

A.3 Standard Training Datasets

In our experiments, we use three standard training datasets for multi-modal reasoning: Geometry3K [25], GeoQA [4], and MMR1 [20]. Geometry3K consists of 2.1K multiple-choice questions in the training set, covering a wide range of geometric shapes. GeoQA includes 8K fill-in-the-blank questions sourced from the larger Geo170K dataset [10]. MMR1 consists of 7,000 samples and includes both multiple-choice questions and fill-in-the-blank questions. These samples cover a range of tasks, including understanding charts and geometric reasoning.

B Additional Experiments

B.1 Training Dynamics

To better understand the behavior of MM-UPT during training, we monitor several diagnostic metrics, including the majority voting reward and entropy, both of which are label-free and provide insights in the absence of ground-truth supervision. In particular, majority voting reward is calculated following Equation 2. Entropy can be used as an unsupervised objective that measures the uncertainty of the model's generation [46, 52, 68]. For a group of responses $O = \{o_i\}_{i=1}^{G}$ sampled from the question q and image I, we cluster the responses according to their meaning. That is, if two responses share the same meaning (i.e., extracted answers), they should be merged into one same cluster in the semantic space. This results to $K (K \leq G)$ clusters $C = \{c_j\}_{j=1}^{K}$. The empirical distribution over clusters is defined as:

$$p(c_j|q, I) = \frac{|c_j|}{G},$$

where $|c_j|$ denotes the number of responses that belongs to c_j. The semantic entropy (denoted as H) over the model's response meanings distribution can be estimated as follows:

$$H = -\sum_{c_j \in \{C\}} p(c_j|q, I) \log p(c_j|q, I).$$

Figure 2 presents the MM-UPT training curves of the key metrics on Qwen2.5-VL-7B using the MMR1 dataset. We observe that the majority voting reward consistently increases over time, accompanied by a steady decrease in the entropy. This indicates that the model is converging toward more consistent predictions, reflecting improved confidence and stability in its responses.

Additionally, we track the change in average benchmark accuracy and effective rank [52] throughout training. The accuracy exhibits an upward trend, demonstrating that our MM-UPT framework–based on an online reinforcement learning algorithm–effectively enables the model to self-improve continuously and iteratively. The effective rank [52] further measures the amount of knowledge the model comprehends in the datasets. During training, the internal knowledge of the model is exploited, leading to a consistent increase in the effective rank on the benchmark.

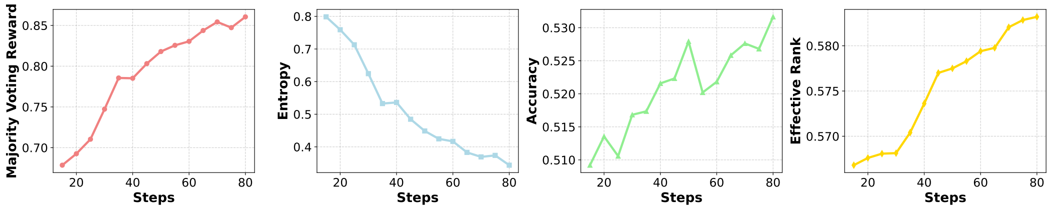


Figure 2: Training dynamics of MM-UPT using Qwen2.5-VL-7B on the MMR1 dataset. We plot the majority voting reward, semantic entropy, and average benchmark accuracy over the course of unsupervised post-training.

Table 6: Performance on non-mathematical VQA benchmarks. We evaluate Qwen2.5-VL-7B before and after applying MM-UPT on the MMR1 dataset. Scores are reported as accuracy.

Models	**ChartQA**	**IconQA**
Qwen2.5-VL-7B	71.96	54.20
Qwen2.5-VL-7B + MM-UPT	77.48 (**7.7%**↑)	56.55 (**4.3%**↑)

B.2 Generalization Beyond Multimodal Mathematical Reasoning

A potential concern is that by encouraging convergence toward high-consensus answers, MM-UPT might reduce response diversity and overspecialize on mathematical reasoning, potentially harming performance on other tasks. To investigate whether this method suffers from a negative impact on broader generalization, we extend our evaluation to two non-mathematical visual question answering benchmarks: ChartQA [30], which tests visual perception over charts, and IconQA [26], which focuses on abstract diagram understanding. In particular, we evaluate the Qwen2.5-VL-7B model trained with MM-UPT on the MMR1 dataset. As shown in Table 6, our method not only avoids performance degradation but leads to notable improvements on both benchmarks. This suggests that the underlying mechanism of MM-UPT—reinforcing the generation of more reliable and consistent answers—is a beneficial trait that positively transfers to other domains. The results indicate that the induced consistency does not degrade, and can even enhance, the model's general utility on related visual understanding tasks.

B.3 Adaptability to Language Tasks

Furthermore, to assess the adaptability of our MM-UPT framework to purely language-based domains, we apply MM-UPT to a language model, Qwen2.5-MATH-7B [58], trained on the DAPO-17K dataset [63] using the same self-rewarding mechanism. We evaluate the resulting model on two popular pure-math benchmarks: MATH [14] and Omni-MATH [9]. As shown in Table 7, MM-UPT leads to substantial improvements on both benchmarks. These results strongly suggest that our framework is not limited to multi-modal settings and can be effectively extended to purely language-based reasoning tasks, provided the base model demonstrates sufficient initial competency.

Table 7: Performance on pure-language mathematical reasoning benchmarks. We evaluate Qwen2.5-MATH-7B before and after applying our unsupervised post-training method.

Models	**MATH**	**Omni-MATH**
Qwen2.5-MATH-7B	51.20	18.09
Qwen2.5-MATH-7B + UPT	75.80 (**48.0%**↑)	32.72 (**80.9%**↑)

C Compute Resources

We conduct our experiments using NVIDIA H100-80G and A800-40G GPUs. The experimental time using 8 A800 for training Qwen2.5-VL-7B [1] on the Geometry3K [25] dataset using GRPO is around 10 hours.