SoTA with Less: MCTS-Guided Sample Selection for Data-Efficient Visual Reasoning Self-Improvement

Xiyao Wang[1,2†], **Zhengyuan Yang**[2], **Chao Feng**[3,4], **Hongjin Lu**[1]
Linjie Li[2], **Chung-Ching Lin**[2], **Kevin Lin**[2], **Furong Huang**[1,‡], **Lijuan Wang**[2,‡]
[1]University of Maryland, College Park [2]Microsoft
[3]University of Michigan [4]Cornell University
[†]xywang@umd.edu [‡]Equal advise

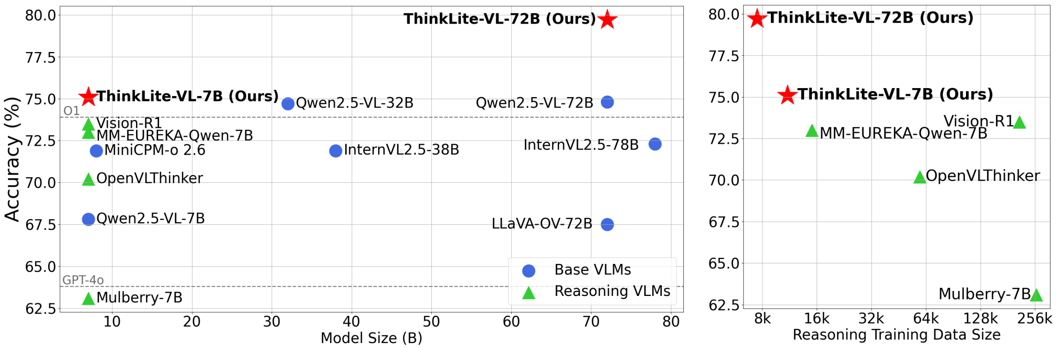


Figure 1: Recent "Reasoning VLMs" studies finetune "Base VLMs" with extra reasoning training data to improve visual reasoning. This paper presents a data-efficient self-improving method for better training reasoning VLMs. **(Left)** Comparison of VLMs with different parameter sizes on **MathVista**. Our model ThinkLite-VL-7B achieves the state-of-the-art (SoTA) accuracy of **75.1**, surpassing Qwen2.5-VL-72B-Instruct, GPT-4o, O1, and other 7B-level reasoning VLMs. ThinkLite-VL-72B further pushes this boundary to **79.7**. **(Right)** Comparison of the reasoning training data size used by 7B-level and 72B-level reasoning models. Our model achieves SoTA performance using only 11k data (7B) and 7.5k data (72B), and without any additional knowledge distillation.

Abstract

We introduce **ThinkLite-VL**, a family of visual reasoning models that achieve state-of-the-art (SoTA) performance using an order of magnitude fewer training samples, relying purely on reinforcement fine-tuning (RFT) self-improvement without any knowledge distillation. Our central insight is that *sample difficulty* critically influences RFT effectiveness: appropriately challenging examples can drive substantial reasoning improvements, even in low-data regimes. However, quantifying sample difficulty in a reliable and scalable manner remains non-trivial. To address this, we repurpose Monte Carlo Tree Search (MCTS) to measure sample difficulty via the number of reasoning iterations a vision-language model (VLM) requires to solve each instance. This MCTS-based selection procedure identifies samples that induce deeper reasoning while remaining solvable, allowing us to filter a high-quality subset from 70k open-source examples spanning math, natural image understanding, and chart comprehension. Using this approach, we select just 11k challenging samples for RFT on Qwen2.5-VL-7B-Instruct and 7.5k samples for Qwen2.5-VL-72B-Instruct. The resulting models, ThinkLite-VL-7B and ThinkLite-VL-72B, significantly outperform their respective base models across eight visual reasoning benchmarks. In particular, ThinkLite-VL-7B improves the average

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

performance of Qwen2.5-VL-7B-Instruct by 7% and surpasses all existing 7B-level models, as well as much larger models such as GPT-4o, O1 and Qwen2.5-VL-72B, achieving a new SoTA score of **75.1** on MathVista. ThinkLite-VL-72B further advances the SoTA frontier, achieving an accuracy of **79.7** on MathVista and an average benchmark improvement of **4.42** over the open-source SOTA. These results demonstrate that MCTS-guided difficulty filtering provides a scalable and effective path toward data-efficient self-improvement in multimodal reasoning. Our code, data, and model are available at https://github.com/si0wang/ThinkLite-VL.

1 Introduction

Large language models (LLMs) have demonstrated strong capabilities in solving complex reasoning tasks—such as mathematics and coding—by leveraging chain-of-thought prompting and reflection mechanisms [24, 36]. Recent work [16] highlights the critical role of reinforcement fine-tuning (RFT) in further enhancing reasoning performance. Remarkably, these improvements can be achieved purely via RFT, even without post-training supervised fine-tuning (SFT).

However, despite the success of RFT in LLMs, its impact on vision-language models (VLMs) has been less pronounced. A likely cause is the inherent modality gap: VLMs are pretrained on text-heavy objectives, while post-training tasks demand multimodal reasoning. Recent efforts [22, 12, 55, 84] have addressed this by incorporating knowledge distillation and supervised format alignment before RFT. While effective, these pipelines are cumbersome, and fundamentally limit the capacity for models to improve via self-training alone.

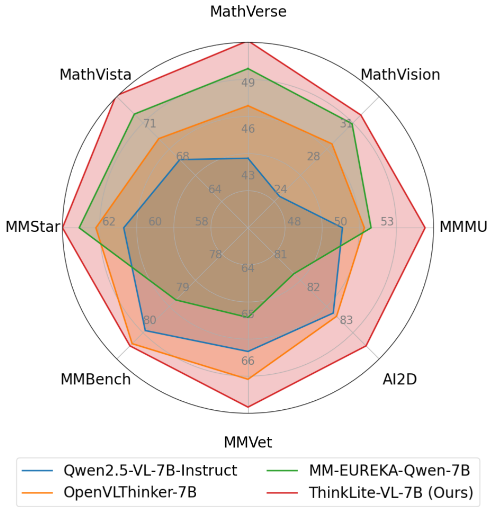


Figure 2: Performance comparison on 8 visual benchmarks. Our model significantly outperforms Qwen2.5-VL-7B and other reasoning models.

In this work, we demonstrate that high-quality and appropriately *challenging* training samples alone are sufficient to enable self-improvement in VLMs via RFT—without any knowledge distillation. When the training data matches the base model's capability level, RFT can explore informative rollouts by itself and substantially elevate multimodal reasoning ability. Based on this insight, we introduce **ThinkLite-VL**, a family of data-efficient reasoning VLMs trained via RFT on a small subset of difficulty-curated examples.

The key to ThinkLite-VL's performance lies in effective sample selection. We propose to repurpose Monte Carlo Tree Search (MCTS)—a classic inference-time search algorithm—to estimate the difficulty of each training instance. Specifically, we define difficulty as the number of MCTS reasoning iterations a VLM requires to solve a task. This search-based signal tightly correlates with sample difficulty and naturally identifies examples that promote deeper reasoning during training.

Our pipeline begins with 70k open-source samples spanning three core domains: mathematical reasoning, natural image understanding, and chart interpretation. For each example, we simulate an MCTS-based inference trace using the base VLM, and rank samples by the number of reasoning steps required to reach a correct solution. From this pool, we extract two difficulty-filtered subsets: 11k samples for Qwen2.5-VL-7B-Instruct and 7.5k samples for Qwen2.5-VL-72B-Instruct. We then apply RFT directly on these subsets—no supervised fine-tuning or distillation required.

We evaluate our resulting models, ThinkLite-VL-7B and ThinkLite-VL-72B, on eight established VLM benchmarks. After RFT, ThinkLite-VL-7B improves the average performance of Qwen2.5-VL-7B-Instruct from 59.69% to 64.18%, and outperforms a comparable baseline trained on randomly selected 11k samples (60.89%). Similarly, ThinkLite-VL-72B raises the average accuracy of Qwen2.5-VL-72B-Instruct from 68.25% to 72.67%, exceeding the baseline trained on randomly selected 7.5k samples 69.91%.

Furthermore, compared with the most recent 7B-level reasoning VLMs, ThinkLite-VL-7B consistently demonstrates substantial performance advantages as shown in Figure 2. ThinkLite-VL-7B also outperforms much larger models—including GPT-4o, Qwen2.5-VL-72B, and o1—on the MathVista benchmark, achieving a new SoTA score of **75.1%** (Figure 1). ThinkLite-VL-72B further advances the frontier, attaining a SoTA accuracy of **79.7%** on MathVista.

Our key contributions are:

(1) **Difficulty as a learning signal.** We identify sample difficulty as a critical yet underutilized signal for enabling effective self-improvement in VLMs via RFT, and show the importance of scaling compute for identifying the appropriately challenging training sample.

(2) **MCTS-guided filtering.** We propose a novel use of Monte Carlo Tree Search to estimate sample difficulty by measuring model reasoning iteration count. Across diverse online and offline baselines, MCTS-guided filtering delivers superior performance, benefiting from the explicit tree search.

(3) **Data-efficient RFT pipeline.** We introduce ThinkLite-VL, a data-efficient visual reasoning framework that achieves SoTA performance using only 11k (7B) and 7.5k (72B) training samples, without any knowledge distillation.

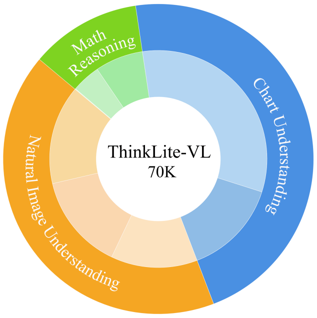
(4) **Strong empirical gains.** We demonstrate that ThinkLite-VL-7B and ThinkLite-VL-72B outperform strong baselines and existing SoTA models across eight VLM benchmarks. Notably, ThinkLite-VL-7B improves the average performance of its base model by 7%, and achieves a new SoTA score of 75.1% on MathVista—surpassing larger models such as GPT-4o, O1 and Qwen2.5-VL-72B. ThinkLite-VL-72B further advances this with a MathVista score of 79.7%.

(5) **Open-source release.** We will release the full ThinkLite-VL model family, including both ThinkLite-VL-7B and ThinkLite-VL-72B, and MCTS-filtered training sets for both Qwen2.5-VL-7B and Qwen2.5-VL-72B to support future research in multimodal reasoning.

2 Related work

Large language model reasoning. Simulating human-like thinking processes through intermediate reasoning steps has significantly improved the performance of large language models (LLMs) on tasks that require reasoning [24]. One family of methods focuses on explicitly controlling the structure or format of the model's outputs, such as by applying Chain-of-Thought (CoT) prompting [78] and Self-Consistency [77]. Related lines of work include more elaborate reasoning strategies like Tree of Thoughts [87] or Graph of Thoughts [4]. Additionally, some approaches involve supervised fine-tuning (SFT) on curated datasets with reasoning annotations [52, 88]. Researchers have also explored process reward models (PRMs) that encourage systematic thought processes [35, 66, 70, 28, 98, 47]. Others incorporate search techniques, including Monte Carlo Tree Search (MCTS) or beam search, to refine or verify reasoning paths [80, 81, 5, 15, 19, 73]. Recently, large-scale RL with outcome-based reward functions has been leveraged [16] to elicit powerful reasoning capabilities in LLMs. Unlike prior uses of MCTS at inference time [80, 81, 15], we employ MCTS during training to assess sample difficulty and curate a high-impact training subset for RFT. We focus on how to use large-scale RL to enhance the reasoning ability of VLMs.

Vision language model reasoning. Vision language models [1, 68, 38, 23, 37, 3, 10, 64, 31, 85] can perform vision tasks using language given visual input through vision encoders like [57, 94, 65]. These models demonstrate comprehensive multimodal capabilities across various scenarios [93, 39, 91, 49, 18, 92, 20, 32] and exhibit reasoning capabilities to some extent [43, 76, 41, 96, 69]. Inspired by the success of reasoning in LLMs, researchers have sought to improve the reasoning capabilities of VLMs. For instance, CoT prompting is applied to VLMs [97, 51, 46, 11, 100, 21] and some papers create multimodal datasets [86, 83, 61, 99, 12, 22, 17, 63], using SFT for knowledge distillation to improve reasoning abilities. Some prior works have also explored improving VLM performance through self-improvement strategies [102, 72, 75, 13]. More recently, RL training has emerged as a promising approach to further strengthen the reasoning capabilities of VLMs [12, 22, 50, 82]. While recent works explore SFT and RL [12, 22] for VLM reasoning, efficiently utilizing training data and avoiding costly knowledge distillation remains a challenge. In contrast, ThinkLite-VL eliminates the need for SFT or distillation entirely and achieves SoTA performance using just 11k (7B) and 7.5k (72B) samples—an order of magnitude less than prior work. Specifically, we propose a novel approach using MCTS to filter for high-quality training instances based on the difficulty level. We



Category	QA Category	Data source	Data size
Math Reasoning	Open-ended	🟩 Geometry3K	3001
	Multi-choice	🟩 GeoQA	5010
	Multi-choice	🟩 Geos	66
Natural Image Understanding	Open-ended	🟧 FigureQA	10000
	Multi-choice	🟧 ScienceQA	10332
	Open-ended	🟧 OK-VQA	9009
Chart Understanding	Open-ended	🟦 IconQA	10000
	Open-ended	🟦 TabMWP	22579

Figure 3: Data statistic of ThinkLite-VL-70k training dataset. We find that converting answers to open-ended format is critical in reliably assessing question difficulty and effective model training.

then directly apply RL training to enhance reasoning on this curated data, demonstrating strong performance without requiring any SFT stage.

Data filtration. Data filtration aims to identify and retain high-quality, diverse, and task-relevant data while discarding noisy or redundant information to optimize training efficiency and generalization performance. It is important for the pretraining phase [14, 29, 79, 58, 54, 2, 95, 67, 56] and instruction tuning phase [34, 33, 7, 9, 38, 103, 89] of both LLMs and VLMs. In this paper, we specifically focus on filtering training instances to curate data optimally for efficient downstream RL training to improve the reasoning capabilities of VLMs. A concurrent work, MM-Eureka [50], also investigates the impact of data filtration on RFT. While MM-Eureka [50] filters samples based on zero-shot accuracy, our MCTS-based method provides a more expressive and fine-grained estimate of sample difficulty, capturing both solved and unsolved-but-informative cases. Importantly, our findings reveal that samples requiring extended reasoning—even when not solved by the model—can be highly beneficial during RFT.

To our knowledge, ThinkLite-VL is the first framework to combine search-based sample difficulty estimation with reinforcement fine-tuning—achieving data-efficient self-improvement for visual reasoning at both 7B and 72B scale, without any SFT or distillation.

3 Training Recipe

In this section, we will introduce the complete training pipeline of ThinkLite-VL. First, in Section 3.1, we describe how we collect our training data that we later sample hard problems from. Then, in Section 3.2, we detail how we employ a base model combined with Monte Carlo Tree Search (MCTS) for data filtering to select prompts that are challenging for the base model. Finally, in Section 3.3, we explain how we use these filtered data to train ThinkLite-VL. We note that the proposed data filtering method, introduced in Section 3.2, is the core technical contribution of ThinkLite-VL. Specifically, ThinkLite-VL highlights the importance of difficulty-aware training sample selection in self-improving training, and effectively repurposes MCTS for sample difficulty prediction.

3.1 Data Collection

We collect a total of 70k datas from widely used open-source training datasets as our initial training set, covering three category: multimodel mathematical reasoning (Geometry3K [42], GeoQA [6], Geos [60]), natural image understanding (FigureQA [25], ScienceQA [43], OK-VQA [48]), and chart understanding (IconQA [45], TabMWP [44]). For FigureQA and IconQA, due to the large size of their original training sets, we only randomly sample 10k data points from each as our training set. The overall data distribution is shown in Figure 3. Each training sample is organized into the following format: `(Image, id, Prompt, Answer)`.

Furthermore, to prevent the VLM from obtaining correct answers by merely guessing from multiple-choice options, we reformulated IconQA, FigureQA, Geometry3K, TabMWP, and OK-VQA from a multiple-choice format to an open-ended format. This modification compels the VLM to derive the correct answer through reasoning rather than selection, thereby increasing the difficulty of the tasks and enhancing the reliability of the data filtering process described in the subsequent section.

3.2 MCTS-based Sample Selection

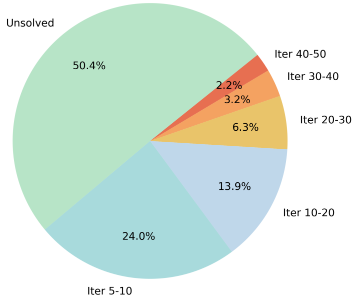


Figure 4: Data difficulty distribution of our 11k training set after 7B MCTS-based data filtration. Unsolved refers to data that VLM cannot solve after 50 MCTS iterations.

In our work, the collected data primarily originates from commonly used pretraining datasets for existing VLMs, which makes the model susceptible to overfitting on certain samples. Inspired by recent successes of data filtration in LLM SFT [53, 88] and conventional reinforcement learning [59, 74], we propose a MCTS-based sample selection mechanism. This approach leverages the VLM's own iterative reasoning process, using the number of iterations required to reach the correct answer as a metric to assess the difficulty of each data sample. Consequently, we can selectively filter for those samples that are more challenging for the model during RL training, rather than using the entire dataset.

Specifically, we define the state at step t, denoted as s_t, to represent the prefix of the reasoning chain. The introduction of a new reasoning step, a, transitions the state to s_{t+1}, which is formed by concatenating s_t with a. By leveraging VLM itself as policy model, π_θ, we sample candidate steps from the probability distribution $\pi_\theta(a|x, I, s_t)$, where x denotes the task's input prompt and I represents the input image. The MCTS process starts from the root node, s_0, representing the beginning of a sentence. It then iteratively proceeds through three key phases—selection, expansion and simulation—which are described in detail in the subsequent paragraphs. In contrast to previous studies, during the data filtering stage with MCTS, we prioritize computational efficiency and comprehensive exploration of the solution space, with our focus centered on self-rewarding setting. Consequently, throughout the MCTS process, we ***do not employ any pretrained or separately trained process reward models***, thereby simplifying and accelerating the procedure. The prompt used for MCTS is shown in Appendix A Table 6.

Selection. In our MCTS procedure, the selection process is only determined by the visitation frequency, denoted as $N(s_t)$, of the current state s_t. At node s_t, the subsequent node is selected according to the following formula: $s_{t+1} = \arg\max_{s_t} \left[\mathrm{c}_{\text{puct}} \cdot \frac{\sqrt{N(s_t)}}{1+N(s_{t+1})} \right]$

Expansion. Given a current step s_t, the VLM generates k distinct actions based on the prompt and image through temperature decoding. Each of these actions is then combined with the current step to form k candidates next steps. The diversity among these actions is regulated by temperature parameter, which is set to 0.5 in our experiments, with k configured as 3.

Simulation. After selecting a node , we directly utilize the policy π_θ to generate several reasoning steps until a final answer is produced or a preset reasoning step limit is reached. Subsequently, we employ the corresponding LLM (in our experiments, the Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-72B-Instruct are used, with Qwen2.5-7B-Instruct serving as the critic model) to compare the generated final answer with the ground truth answer, thereby determining the correctness of the response. If the answer is correct, the MCTS process is terminated and the current iteration number K is recorded; if the answer is incorrect, the visit count N of the selected node is updated and the next iteration commences. Appendix A Table 7 illustrates the prompt employed for the critic model.

Data filtration. We apply this MCTS procedure to the entire collection of 70k data samples and record the iteration number K required to solve each problem, using Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-72B-Instruct as the policy model. In this process, K served as a metric for assessing the difficulty of each sample: a higher K indicates that the VLM requires more extensive exploration to arrive at the correct answer, thereby reflecting a greater level of challenge. Ultimately, we select all samples with K greater than 5, as well as those that remained unsolved after 50 iterations, resulting in a final training set of 11k samples with 7B model and 7.5k samples with 72B model. The data difficulty distribution of 11k training set of 7B model is shown in Figure 4 as an example.

Table 1: Visual reasoning training data comparison between ThinkLite-VL-7B and other 7B-level VLM reasoning models. ALL these reasoning models have distilled knowledge from larger models or closed-source models except for MM-Eureka-Qwen-7B. MM-Eureka-Qwen-7B performs accuracy-based data filtering before training and uses more data (15k) than ours. Here the data size refers to the amount of extra visual reasoning data used to boost the base model for reasoning, via SFT or RFT.

Reasoning Models	Knowledge Distillation (KD)	RFT	Data size
LLaVA-Cot-11B [83]	GPT-4o	✗	100k
Mulberry-7B [86]	GPT-4o, Qwen2-VL-72B	✗	260k
Vision-R1-7B [22]	Deepseek-R1	✓	200k + 10k
OpenVLThinker-7B [12]	DeepSeek-R1-Distill-Qwen-14B	✓	59.2k
MM-EUREKA-Qwen-7B [50]	–	✓	15k
ThinkLite-VL-7B	–	✓	11k

3.3 Visual Reasoning Training

Unlike previous VLM reasoning studies, which heavily depend on large-scale Chain-of-Thought (CoT) data generated by external models and employ SFT for knowledge distillation to enhance reasoning capabilities (as shown in Table 1), we demonstrate that directly performing reinforcement fine-tuning (RFT) with a small amount of high-quality training data can significantly enhance the reasoning ability of VLMs, without the need for extensive external data generation.

After conducting MCTS-based sample selection and obtaining a filtered set of high-quality training data (11k for 7B and 7.5k for 72B), we then perform RL fine-tuning on the Qwen2.5-VL models using these selected data. Specifically, we employ Group Relative Policy Optimization (GRPO) loss function proposed by [62] for training, with the objective defined as follows:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_\theta^{\text{old}}(O|q)}$$
$$\left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left\{ \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta^{\text{old}}(o_{i,t} \mid q, o_{i,<t})} \hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_\theta^{\text{old}}(o_{i,t} \mid q, o_{i,<t})}, 1-\epsilon, 1+\epsilon \right) \hat{A}_{i,t} \right\} - \beta D_{\text{KL}}(\pi_\theta \parallel \pi_{\text{pre}}) \right]. \tag{1}$$

We provide the training prompt template during RFT in Appendix A Table 8.

4 Experiments

4.1 Benchmark Evaluation

We systematically evaluate ThinkLite-VL on several commonly used multimodal benchmark datasets and perform comprehensive comparisons with existing reasoning models. Through these experiments, we demonstrate the effectiveness and advantages of our model in multimodal reasoning tasks.

Baseline VLMs. We compare our method with both 7B level and 72B level models as follows:

• For 7b-level VLMs, we use Qwen2.5-VL-7B-Instruct as the base model and perform RFT on the 11k high-quality data obtained through MCTS-based filtration, resulting in our reasoning model, named **ThinkLite-VL-7B**. We conduct training using Easy-R1 [101] code base and set GRPO rollout number as 32. Our main baselines are as follows: (1) Qwen2.5-VL-7B-Instruct [3], serving as our base model; (2) ThinkLite-VL-Random11k, trained using RFT on a randomly sampled subset of 11k instances from the full 70k dataset. Besides, we report the performance of several recent general and reasoning VLMs for comparison, including general opensourced models LLaVA-Onevision-7B [30] and InternVL2.5-8B [10], the SFT-based reasoning models LLaVA-Cot-11B [83] and Mulberry-7B [86], as well as the RFT-based reasoning models Vision-R1 [22], MM-Eureka-Qwen-7B [50], and OpenVLThinker-7B [12].

• For 72B-level VLMs, we use Qwen2.5-VL-72B-Instruct as the base model. We perform RFT on the 7.5k high-quality data obtained by Qwen2.5-VL-72B-Instruct through MCTS-based filtration and get 72B reasoning model **ThinkLite-VL-72B**. The 72B-level baselines include: (1) our base model Qwen2.5-VL-72B-Instruct [3]; (2) two opensourced general VLMs LLaVA-Onevision-72B [30] and InternVL2.5-78B [10]; (3) one opensouced reasoning model QvQ-72B [71]; (4) ThinkLite-VL-Random7.5k, trained using RFT on 7.5k randomly selected samples from the full 70k dataset.

Table 2: Comparison of different VLMs on 8 widely used visual benchmarks. Our model achieves SoTA performance at both 7B level and 72B level on 6 benchmarks and reaches a SoTA performance of 79.7 on MathVista among all VLMs. On average, our model improves performance by 7.5% and 6.5% compared with our base models Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-72B-Instruct. We do not evaluate Mulberry-7B on MathVision because Mulberry-7B uses MathVision as training dataset. We evaluate all models with same code using vLLM [27] inference. For reasoning models, we use thinking templates provided in their codebase to generate thoughts and get the final answer.

Models	Data size	MathVista testmini	MathVision mini	MathVerse mini	MMMU	MMStar	MMBench	MM-Vet	AI2D	Avg.
Proprietary Models										
OpenAI-GPT-4o	–	63.8	36.8	50.2	69.1	64.7	83.4	69.1	84.6	65.21
OpenAI-o1	–	73.9	58.2	57.0	77.6	–	–	–	–	–
7B-level General and Reasoning Vision-Language Models										
LLaVA-Onevision-7B	–	63.2	17.4	26.2	48.8	61.7	80.8	57.5	81.4	54.63
InternVL2.5-8B	–	64.4	22.0	39.5	54.9	62.8	**82.7**	**68.8**	83.3	59.80
Qwen2.5-VL-7B-Instruct	–	67.8	23.6	44.5	50.6	61.7	80.7	66.0	82.6	59.69
LLaVA-Cot-11B	100k	54.8	16.3	33.9	46.2	57.6	75.0	60.3	78.7	52.85
Mulberry-7B	260k	63.1	–	39.6	55.0	61.3	79.2	63.7	80.1	–
Vision-R1-7B	210k	73.5	30.7	51.9	50.5	60.2	78.9	65.6	80.4	61.46
OpenVLThinker-7B	59.2k	70.2	29.6	47.9	51.9	63.2	81.3	66.9	82.7	61.71
MM-EUREKA-Qwen-7B	15k	73.0	31.9	50.3	52.3	64.1	79.3	64.9	81.4	62.15
Our 7B-level Reasoning Model										
ThinkLite-VL-7B-Random11k	11k	71.9	26.1	47.3	51.7	62.7	81.1	65.5	80.9	60.89
ThinkLite-VL-7B	11k	**75.1**	**32.9**	**52.1**	**55.5**	**65.0**	81.4	67.8	**83.6**	**64.18**
Δ (Ours - Random selection)	–	+3.2	+6.8	+4.8	+3.8	+2.3	+0.3	+2.3	+2.7	+3.29
Δ (Ours - Open 7B SoTA)	–	+1.6	+1.0	+0.2	+0.5	+0.9	-1.3	-1.0	+0.3	+2.03
72B-level General and Reasoning Vision-Language Models										
LLaVA-Onevision-72B	–	67.5	29.3	39.1	56.8	66.1	85.9	63.7	85.6	61.75
InterVL2.5-78B	–	72.3	34.9	51.7	68.7	68.9	87.2	72.3	**87.9**	67.99
Qwen2.5-VL-72B-Instruct	–	74.8	35.2	53.3	63.4	68.4	87.4	76.3	87.2	68.25
QvQ-72B	–	71.4	32.7	48.6	**70.3**	67.2	86.3	75.9	86.6	67.37
Our 72B-level Reasoning Model										
ThinkLite-VL-72B-Random7.5k	7.5k	76.4	37.1	57.5	65.8	71.3	87.6	76.7	86.9	69.91
ThinkLite-VL-72B	7.5k	**79.7**	**43.8**	**64.3**	68.3	**72.0**	**88.2**	**77.3**	87.7	**72.67**
Δ (Ours - Random selection)	–	+3.3	+6.7	+6.8	+2.5	+0.7	+0.6	+0.6	+0.8	+3.06
Δ (Ours - Open 72B SoTA)	–	+4.9	+8.6	+11.0	-2.0	+3.1	+0.8	+1.0	-0.2	+4.42

We also include proprietary models as performance references which include OpenAI-GPT-4o and OpenAI-o1. For all models, we use 8×80G A100 GPUs for model training and evaluation.

Benchmarks. We select eight widely used VLM benchmarks for evaluation, namely MathVista [41], MathVison [69], MathVerse [96], MMMU [93], MMStar [8], MMBench [40], MMVet [91], and AI2D [26]. Among them, MathVista, MathVison, and MathVerse are widely used in VLM research to evaluate mathematical reasoning capabilities, while MMVet also includes a significant number of mathematical reasoning tasks. In contrast, MMMU, MMStar, MMBench, and AI2D are primarily utilized to assess VLM's visual perception reasoning and scientific reasoning abilities.

SoTA performance over both 7B and 72B models. As shown in Table 2, ThinkLite-VL-7B and ThinkLite-VL-72B show a significant improvement in average performance across the eight benchmarks compared to the base model Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-72B-Instruct, with the average performance increasing from 59.69 to 63.89 and 68.25 to 72.67, respectively. ThinkLite-VL-7B also outperforms reasoning models that primarily achieve performance enhancement through extensive knowledge distillation (such as LLaVA-CoT-11B, Mulberry-7B, Vision-R1-7B, and OpenVLThinker-7B) with the closest average performance to GPT-4o. Compared to MM-EUREKA-Qwen-7B, which does not involve SFT knowledge distillation but adopts a larger RL training dataset, our model consistently outperforms across all benchmarks, highlighting the

Table 3: Comparison with models trained on data sampled using different selection strategies, ThinkLite-VL achieves significantly better performance, highlighting the effectiveness and superiority of our proposed MCTS-based sample selection method.

Models	Data size	MathVista testmini	MathVision mini	MathVerse mini	MMMU	MMStar	MMBench	MM-Vet	AI2D	Avg.
ThinkLite-VL-7B	11k	**75.1**	**32.9**	52.1	**55.5**	**65.0**	81.4	**67.8**	**83.6**	**64.18**
ThinkLite-VL-Unsolved	5.6k	73.6	26.9	49.4	52.1	62.7	81.1	67.0	83.5	62.04
ThinkLite-VL-Iter5Only	5.4k	73.5	27.5	50.2	52.5	64.2	80.9	66.9	83.3	62.38
ThinkLite-VL-Random11k	11k	71.9	26.1	47.3	51.7	62.7	81.1	65.5	80.9	60.89
ThinkLite-VL-SelfConsistency	23k	74.6	30.9	50.1	53.8	64.1	81.3	67.1	83.3	63.15
ThinkLite-VL-Fullset	70k	74.3	29.9	**52.2**	53.1	63.7	**81.6**	67.2	83.0	63.13

importance of high-quality data filtering before training, and the effectiveness of the proposed MCTS-based filtering. For more discussion between offline and online data filtration, please refer to Section 4.3. Analyzing individual benchmarks, ThinkLite-VL-7B achieves best performance among all 7B-scale models on six out of eight benchmarks, with only marginal gaps behind InternVL2.5-7B on MMBench and MM-Vet. In addition, ThinkLite-VL-72B outperforms all existing open-source vision-language models across six benchmarks. Notably, ThinkLite-VL-7B attains SoTA accuracy of **75.1** on MathVista, exceeding both GPT-4o and o1. ThinkLite-VL-72B further advances the frontier, reaching **79.7** on MathVista and **64.3** on MathVerse, establishing new SoTA on both benchmarks.

Effectiveness of MCTS-based sample selection. Compared to training on an equal number of randomly selected samples from the full 70K dataset (ThinkLite-VL-7B-Random11k and ThinkLite-VL-72B-Random7.5k), ThinkLite-VL-7B and ThinkLite-VL-72B demonstrate a clear advantage across eight benchmarks, with average performance improvements of 5.4% at the 7B scale and 4.4% at the 72B scale. These results further show the importance of MCTS-based sample selection.

4.2 Importance of MCTS-based Sample Selection

We conduct ablation studies to demonstrate the importance of MCTS-based sample selection. We compare five different training settings of ThinkLite-VL: (1) ThinkLite-VL-Unsolved: Trained using only the 5.6k samples that could not be solved by MCTS, representing the most difficult subset. (2) ThinkLite-VL-Iter5Only: Trained on the subset of data that VLM is able to solve via MCTS, but required more than 5 iterations. This set, combined with the unsolved samples, forms the full 11k training set used in ThinkLite-VL. (3) ThinkLite-VL-Random11k: Trained on a randomly sampled 11k subset from the full 70k dataset, matching the size of the ThinkLite-VL training set. (4) ThinkLite-VL-SelfConsistency: Trained on 23k samples selected based on a self-consistency difficulty measure. Specifically, for each prompt, we perform 50 rollouts using Qwen2.5-VL-7B-Instruct and compute answer accuracy using Qwen2.5-7B-Instruct. Samples with accuracy lower than 0.2 are selected for RFT. (5) ThinkLite-VL-Fullset: Trained on the complete 70k dataset without any filtering. We report the evaluation results of all five settings across the eight VLM benchmarks, as shown in Table 3.

We observe that ThinkLite-VL-7B, trained using 11k samples via MCTS-guided sample selection, achieves the highest average performance among all settings. It outperforms not only the random sampling baseline but also models trained on the full dataset and self-consistency-based filtering, despite using significantly fewer training samples. This highlights the effectiveness of our difficulty-aware data selection strategy. Further analysis reveals that models trained on subsets derived solely from unsolved samples or samples requiring more than five iterations also show decent performance, suggesting that hard and medium-difficulty samples contribute meaningfully to reasoning ability. However, neither subset alone is sufficient. The combination of both unsolved and medium-difficulty samples yields the strongest and most effective training signal. Additional analyses are in Appendix B.

4.3 Comparison with Online Data Selection

In this section, we compare our offline data-selection strategy with an online alternative and evaluate their impact on model performance. We adopt an online baseline based on self-consistency filtering: during training we keep only those samples whose rollout accuracy is greater than 0 but below 0.9,

Table 4: Comparison between ThinkLite-VL and model trained with offline and online self-consistency based sample selection. Our method demonstrates significant advantages.

Model Size	Training type	Selection method	MathVista testmini	MathVision mini	MathVerse mini	MMMU	MMStar	MMBench	MM-Vet	AI2D	Avg.
7B	Offline	MCTS (Ours)	**75.1**	**32.9**	**52.1**	**55.5**	**65.0**	81.4	**67.8**	**83.6**	**64.18**
		SelfConsistency	74.6	30.9	50.1	53.8	64.1	81.3	67.1	83.3	63.15
	Online	SelfConsistency	74.2	26.9	50.1	50.6	64.8	**82.0**	67.1	83.0	62.34
72B	Offline	MCTS (Ours)	**79.7**	**43.8**	**64.3**	**68.3**	**72.0**	**88.2**	**77.3**	**87.7**	**72.67**
		SelfConsistency	77.3	39.1	62.0	66.3	71.6	87.7	77.0	87.1	71.01
	Online	SelfConsistency	76.9	38.5	58.2	66.0	71.7	87.5	77.1	87.4	70.12

drawing additional samples until the training batch is full. Table 4 compares this online variant with our MCTS-based offline selector and a plain offline self-consistency baseline. Similar to the findings in other RL studies [90], the online filter offers negligible improvement except converges faster. The decisive factor is still the ability to identify examples that are truly challenging for the current model, a task at which our MCTS selector excels due to its explicit tree search.

4.4 Data Difficulty Analysis between 7B and 72B Models

We analyze the 11k and 7.5k sample sets selected by 7B and 72B models, to examine how models of different capacity agree on the sample difficulty. We find that there is an overlap of 5.4k samples, where 3.6k of them are instances that neither model is able to solve within 50 MCTS iterations. The real divergence lies in the mid-difficulty stratum. We observe that for this subset, the two models often behave asymmetrically: problems easily solved by the 7B model may require many more iterations for the 72B model, and vice versa, exposing distinct reasoning heuristics across models.

We validate this model-specific preference through cross-sample training: the 11k samples selected by the 7B model are used to RFT the 72B model, and vice versa. Table 5 shows that the gains in both settings were markedly smaller than when each model trains on its own curated set. These results suggest that a sample set tailored to one model transfers poorly to another, even in a strong-to-weak setting. Instead, it is more effective to scale extra compute to find *appropriately* difficult samples that best fit the model itself, as the approach proposed in ThinkLite-VL.

Table 5: Comparison between the 7B and 72B models which trained on each other's selected samples, the resulting performance improvements drops significantly.

Models	Data size	MathVista testmini	MathVision mini	MathVerse mini	MMMU	MMStar	MMBench	MM-Vet	AI2D	Avg.
ThinkLite-VL-7B	7.5k-72B	70.2	26.3	49.2	51.6	61.7	81.1	66.9	82.9	61.24
	11k-7B	**75.1**	**32.9**	**52.1**	**55.5**	**65.0**	81.4	**67.8**	**83.6**	**64.18**
ThinkLite-VL-72B	11k-7B	76.4	38.5	58.4	67.2	70.2	87.3	76.6	87.4	70.24
	7.5k-72B	**79.7**	**43.8**	**64.3**	**68.3**	**72.0**	**88.2**	**77.3**	**87.7**	**72.67**

5 Conclusion

We have introduced an effective self-improvement approach to enhance the reasoning capabilities of VLMs, eliminating the need for external supervision or knowledge distillation. Our key insight highlights the critical importance of selecting appropriately challenging examples for RFT. We find that when training data quality is sufficiently high, even a small dataset can substantially enhance visual reasoning performance without knowledge distillation. Building on this insight, we propose a novel data selection technique, MCTS-based sample selection, which identifies and retains challenging samples by quantifying the number of MCTS reasoning iterations. Starting from 70k initial samples, we obtain a high-quality subset comprising 11k and 7.5k challenging samples for 7B-level and 72B-level models, respectively. These curated datasets are then used to fine-tune the Qwen2.5-VL-7B-Instruct and Qwen2.5-VL-72B-Instruct via RFT, resulting in the reasoning VLMs named ThinkLite-VL-7B and ThinkLite-VL-72B. Our models demonstrate significant improvements across multiple visual reasoning benchmarks, and notably achieves a new SoTA accuracy of 79.7 on MathVista and 64.3 on MathVerse. We hope that our findings on the difficulty-based selection of RFT training data can provide insights for training more effective reasoning VLMs.

References

[1] Gpt-4v(ision) system card. 2023.

[2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. *ArXiv*, abs/2204.14198, 2022.

[3] 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.

[4] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690, 2024.

[5] Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: process supervision without process. *arXiv preprint arXiv:2405.03553*, 2024.

[6] Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P. Xing, and Liang Lin. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning, 2022.

[7] Lichang Chen, Shiyang Li, Jun Yan, Hai Wang, Kalpa Gunaratna, Vikas Yadav, Zheng Tang, Vijay Srinivasan, Tianyi Zhou, Heng Huang, and Hongxia Jin. Alpagasus: Training a better alpaca with fewer data. In *The Twelfth International Conference on Learning Representations*, 2024.

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

[9] Ruibo Chen, Yihan Wu, Lichang Chen, Guodong Liu, Qi He, Tianyi Xiong, Chenxi Liu, Junfeng Guo, and Heng Huang. Your vision-language model itself is a strong filter: Towards high-quality instruction tuning with data selection. *ArXiv*, abs/2402.12501, 2024.

[10] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[11] Zhenfang Chen, Qinhong Zhou, Yikang Shen, Yining Hong, Hao Zhang, and Chuang Gan. See, think, confirm: Interactive prompting between vision and language models for knowledge-based visual reasoning. *arXiv preprint arXiv:2301.05226*, 2023.

[12] Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement, 2025.

[13] 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.

[14] Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, Shawn Presser, and Connor Leahy. The pile: An 800gb dataset of diverse text for language modeling. *ArXiv*, abs/2101.00027, 2020.

[15] Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. Interpretable contrastive monte carlo tree search reasoning. *arXiv preprint arXiv:2410.01707*, 2024.

[16] 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.

[17] 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.

[18] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[19] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

[20] Yunzhuo Hao, Jiawei Gu, Huichen Will Wang, Linjie Li, Zhengyuan Yang, Lijuan Wang, and Yu Cheng. Can mllms reason in multimodality? emma: An enhanced multimodal reasoning benchmark. *arXiv preprint arXiv:2501.05444*, 2025.

[21] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*, 2024.

[22] Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models, 2025.

[23] 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.

[24] 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.

[25] Samira Ebrahimi Kahou, Vincent Michalski, Adam Atkinson, Akos Kadar, Adam Trischler, and Yoshua Bengio. Figureqa: An annotated figure dataset for visual reasoning, 2018.

[26] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images, 2016.

[27] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

[28] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv preprint arXiv:2406.18629*, 2024.

[29] Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. Deduplicating training data makes language models better. In *Annual Meeting of the Association for Computational Linguistics*, 2021.

[30] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[31] Chunyuan Li, Zhe Gan, Zhengyuan Yang, Jianwei Yang, Linjie Li, Lijuan Wang, Jianfeng Gao, et al. Multimodal foundation models: From specialists to general-purpose assistants. *Foundations and Trends® in Computer Graphics and Vision*, 16(1-2):1–214, 2024.

[32] Ming Li, Ruiyi Zhang, Jian Chen, Jiuxiang Gu, Yufan Zhou, Franck Dernoncourt, Wanrong Zhu, Tianyi Zhou, and Tong Sun. Towards visual text grounding of multimodal large language model, 2025.

[33] Ming Li, Yong Zhang, Shwai He, Zhitao Li, Hongyu Zhao, Jianzong Wang, Ning Cheng, and Tianyi Zhou. Superfiltering: Weak-to-strong data filtering for fast instruction-tuning. *ArXiv*, abs/2402.00530, 2024.

[34] Ming Li, Yong Zhang, Zhitao Li, Jiuhai Chen, Lichang Chen, Ning Cheng, Jianzong Wang, Tianyi Zhou, and Jing Xiao. From quantity to quality: Boosting llm performance with self-guided data selection for instruction tuning. In *North American Chapter of the Association for Computational Linguistics*, 2023.

[35] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

[36] Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

[37] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

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

[39] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[40] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer, 2024.

[41] 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. In *International Conference on Learning Representations (ICLR)*, 2024.

[42] 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, 2021.

[43] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022.

[44] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning, 2023.

[45] 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, 2022.

[46] Bozhi Luan, Hao Feng, Hong Chen, Yonghui Wang, Wengang Zhou, and Houqiang Li. Textcot: Zoom in for enhanced multimodal text-rich image understanding. *arXiv preprint arXiv:2404.09797*, 2024.

[47] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2, 2024.

[48] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Ok-vqa: A visual question answering benchmark requiring external knowledge, 2019.

[49] 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.

[50] Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.

[51] Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14420–14431, 2024.

[52] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

[53] Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025.

[54] Guilherme Penedo, Hynek Kydlícek, Loubna Ben Allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro von Werra, and Thomas Wolf. The fineweb datasets: Decanting the web for the finest text data at scale. *ArXiv*, abs/2406.17557, 2024.

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

[56] Filip Radenovic, Abhimanyu Dubey, Abhishek Kadian, Todor Mihaylov, Simon Vandenhende, Yash J. Patel, Yi Wen, Vignesh Ramanathan, and Dhruv Kumar Mahajan. Filtering, distillation, and hard negatives for vision-language pre-training. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6967–6977, 2023.

[57] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.

[58] Laura Ruis, Maximilian Mozes, Juhan Bae, Siddhartha Rao Kamalakara, Dwarak Talupuru, Acyr F. Locatelli, Robert Kirk, Tim Rocktaschel, Edward Grefenstette, and Max Bartolo. Procedural knowledge in pretraining drives reasoning in large language models. *ArXiv*, abs/2411.12580, 2024.

[59] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay, 2016.

[60] Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1476, Lisbon, Portugal, September 2015. Association for Computational Linguistics.

[61] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *Advances in Neural Information Processing Systems*, 37:8612–8642, 2024.

[62] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024.

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

[64] Peter Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Adithya Jairam Vedagiri IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.

[65] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

[66] Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

[67] Alex Wang, Kevin Lin, David Junhao Zhang, Stan Weixian Lei, and Mike Zheng Shou. Too large; data reduction for vision-language pre-training. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3124–3134, 2023.

[68] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022.

[69] 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. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

[70] Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.

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

[72] Xiyao Wang, Jiuhai Chen, Zhaoyang Wang, Yuhang Zhou, Yiyang Zhou, Huaxiu Yao, Tianyi Zhou, Tom Goldstein, Parminder Bhatia, Furong Huang, et al. Enhancing visual-language modality alignment in large vision language models via self-improvement. *arXiv preprint arXiv:2405.15973*, 2024.

[73] Xiyao Wang, Linfeng Song, Ye Tian, Dian Yu, Baolin Peng, Haitao Mi, Furong Huang, and Dong Yu. Towards self-improvement of llms via mcts: Leveraging stepwise knowledge with curriculum preference learning. *arXiv preprint arXiv:2410.06508*, 2024.

[74] Xiyao Wang, Wichayaporn Wongkamjan, Ruonan Jia, and Furong Huang. Live in the moment: Learning dynamics model adapted to evolving policy. In *International Conference on Machine Learning*, pages 36470–36493. PMLR, 2023.

[75] Xiyao Wang, Zhengyuan Yang, Linjie Li, Hongjin Lu, Yuancheng Xu, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Scaling inference-time search with vision value model for improved visual comprehension. *arXiv preprint arXiv:2412.03704*, 2024.

[76] Xiyao Wang, Yuhang Zhou, Xiaoyu Liu, Hongjin Lu, Yuancheng Xu, Feihong He, Jaehong Yoon, Taixi Lu, Gedas Bertasius, Mohit Bansal, et al. Mementos: A comprehensive benchmark for multimodal large language model reasoning over image sequences. *arXiv preprint arXiv:2401.10529*, 2024.

[77] 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.

[78] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[79] Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy Liang. Data selection for language models via importance resampling. *ArXiv*, abs/2302.03169, 2023.

[80] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

[81] Huajian Xin, ZZ Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, et al. Deepseek-prover-v1. 5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv preprint arXiv:2408.08152*, 2024.

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

[83] Guowei Xu, Peng Jin, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. Llava-cot: Let vision language models reason step-by-step, 2025.

[84] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, Haoyu Lu, Dacheng Yin, Fengyun Rao, Minfeng Zhu, Bo Zhang, and Wei Chen. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. *arXiv preprint arXiv:2503.10615*, 2025.

[85] Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1, 2023.

[86] Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and Dacheng Tao. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search, 2024.

[87] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

[88] Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025.

[89] Qifan Yu, Zhebei Shen, Zhongqi Yue, Yang Wu, Wenqiao Zhang, Yunfei Li, Juncheng Li, Siliang Tang, and Yueting Zhuang. Mastering collaborative multi-modal data selection: A focus on informativeness, uniqueness, and representativeness. *ArXiv*, abs/2412.06293, 2024.

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

[91] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities, 2024.

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

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

[94] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[95] Lei Zhang, Fangxun Shu, Tianyang Liu, Sucheng Ren, Hao Jiang, and Cihang Xie. Filter&align: Leveraging human knowledge to curate image-text data. 2023.

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

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

[98] Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. *arXiv preprint arXiv:2501.07301*, 2025.

[99] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[100] Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191, 2023.

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

[102] 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.

[103] 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.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

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

   Answer: [Yes] .

   Justification: See Abstract and Introduction sections.

   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: Section E

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: [NA] .

   Justification: [NA]

   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: See contributions in introduction and Section 4.

   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 all training data and code.

   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: Section 4

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes] .

   Justification: Section 4

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - 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: Section 4

   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:

   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: [Yes]

    Justification: We discuss both potential positive societal impacts and negative societal impacts in the conclusion section.

    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 high misuse 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: The paper properly credits the creators of existing assets (e.g., VLLM, Qwen) with citations, explicitly stating their licenses and confirming compliance with their terms of use.

    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: This paper includes all training code in supplementary materials.

    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: [NA] .

Justification: The paper does not involve crowdsourcing or 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: [NA] .

    Justification: The paper does not involve research with human subjects or crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

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

    Answer: [Yes] .

    Justification: We used the Qwen2.5-VL series to filter and select training samples. This LLM-based selection process is a critical step in constructing our training dataset and constitutes an essential, non-standard component of our core methodology.

    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.

A Prompts used in experiments

A.1 Prompt for MCTS

The prompt used for MCTS is shown in Table 6.

Table 6: Prompt used for VLM during MCTS procedure. We provide two examples of in-context learning to force VLM to follow the reasoning format.

MCTS Prompt Template:

Answer the question **step by step** and provide the final answer at the end, each step should end with **<end>** and put your final answer within $\boxed{}$. Below are two examples:
Question: BoatsRUs built 7 canoes in January of this year and then each subsequent calendar month they built twice the number of canoes they had built the previous month. How many total canoes were built by BoatsRUs by the end of May of this year?
### Step1: To find the result of the total number of canoes built by BoatsRUs by the end of May, I need to find the number of canoes built in each month from January to May and then add them up. <end>
### Step2: To find the number of canoes built in each month, I need to use the formula for the number of canoes built in a given month, which is the number of canoes built in the previous month times 2. <end>
### Step3: So, the number of canoes built in January is 7, the number of canoes built in February is 7 times 2, which is 14, the number of canoes built in March is 14 times 2, which is 28, the number of canoes built in April is 28 times 2, which is 56, and the number of canoes built in May is 56 times 2, which is 112. <end>
### Step4: Now, I can add up these numbers to get the total number of canoes built by BoatsRUs by the end of May: 7 plus 14 plus 28 plus 56 plus 112, which is 217. <end>
### Final Answer: The answer is: $\boxed{217}$.
Question: Find the number of blue circles in the figure.
### Step 1: To find the result of the number of blue circles, I need to interpret the figure. The figure is a Venn diagram with two labeled sets: - One set labeled "blue" contains all the shapes that are blue in color. - The other set labeled "circle" contains all the shapes that are circular in shape. The overlapping region of the Venn diagram contains shapes that are both blue and circular. <end>
### Step 2: The overlapping region contains shapes that meet both criteria: Blue color and Circle shape. From the diagram: - There is **one blue circle** in the overlapping region. <end>
### Final Answer: The answer is: $\boxed{1}$.
Remember to answer the question **step by step**! Here is your question:
Question: {**QUESTION**}

A.2 Prompt for Critic Model

The prompt used for critic model during MCTS is shown in Table 7.

A.3 Prompt for RFT

The prompt used for RFT is shown in Table 8.

B More experiments

B.1 Reward curves of VLM with different training data

We compare the reward curves during RFT of ThinkLite-VL-Random11k, ThinkLite-VL-Fullset, ThinkLite-VL-Iter5Only, and ThinkLite-VL, as shown in Figure 5. Although ThinkLite-VL-

Table 7: Critic prompt for MCTS simulation results evaluation.

Critic Prompt Template:

Please help me judge the correctness of the generated answer and the corresponding rationale.
Question: { }
Ground truth answer: { }
Generated rationale and answer: { }
Your output should only be one sentence: the generated answer is true or false.

Table 8: Prompt template used for reinforcement learning fine-tuning.

Prompt Template:

You FIRST think about the reasoning process as an internal monologue and then provide the final answer. The reasoning process MUST BE enclosed within <think> </think> tags. The final answer MUST BE put in $\boxed{}$.

Random11k and ThinkLite-VL-Fullset achieve higher rewards during training, their actual benchmark performances are inferior to ThinkLite-VL. This observation suggests that incorporating a large number of easy samples into training rapidly improves rewards but fails to enhance the model's reasoning ability. Moreover, ThinkLite-VL exhibits notably lower rewards compared to ThinkLite-VL-Iter5Only, indicating that the unsolved data identified by our MCTS-based sample selection strategy indeed pose significant challenges to the VLM. By progressively learning to solve these challenging problems during training—even if not all are solved completely—the reasoning capabilities of VLMs can be substantially improved.

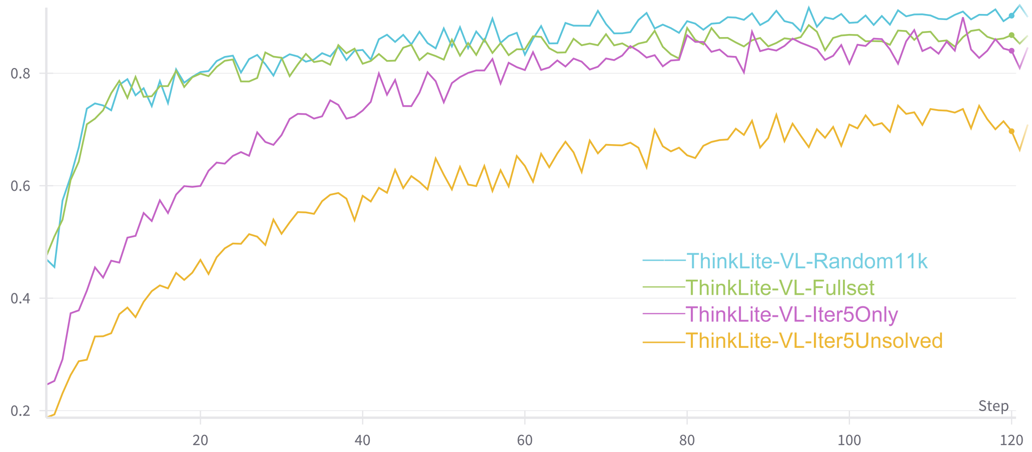


Figure 5: Comparison of reward curves of 7B models trained with different data during RFT. Iter5+Unsolved 11k dataset presents the most challenging learning setting for VLM, highlighting the difficulty of the samples selected by MCTS-based sample selection.

B.2 Ablation Study of Data Difficulty

In this section, we investigate how training data difficulty affects model performance. We present the average performance of models trained using different difficulty data in Table 9. Notably, the model trained with the Iter5+Unsolved subset achieves the highest average score of 63.89, outperforming all other settings. When expanding the difficulty threshold (e.g., Iter10, Iter20, Iter30, and Iter40), the model performance consistently declines, suggesting that medium-difficulty samples are important for improving model reasoning ability. As the difficulty of the training data decreases, the model's performance also declines. This trend suggests that the inclusion of an excessive number of easy samples may weaken the training signal during RFT and ultimately hurt the model's reasoning ability.

Table 9: ThinkLite-VL-7B performance under different training data difficulty settings. Iter5+Unsolved achieves the best performance.

Difficulty level	Data size	Avg. score
Fullset	70k	63.13
Iter1+Unsolved	18k	63.29
Iter5+Unsolved	11k	63.89
Iter10+Unsolved	8k	62.65
Iter20+Unsolved	6.8k	62.61
Iter30+Unsolved	6.1k	62.39
Iter40+Unsolved	5.8k	62.26
Unsolved	5.6k	62.04

C Case Studies

In this section, we present samples of varying difficulty levels selected by the MCTS-based sample selection method across different datasets, as shown in Tables 15 through 14. The difficulty levels are determined based on the number of reasoning iterations required by the VLM to arrive at the correct answer during the MCTS process, providing reference examples for understanding how the method distinguishes between easy and challenging samples.

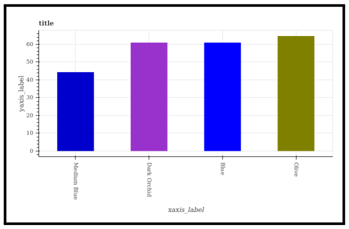
D Limitations

While sample selection has effectively enhanced the reasoning capabilities of vision-language models (VLMs), the overall training efficiency remains a key limitation, both in data filtering and reinforcement learning stages. Common approaches such as self-consistency-based selection and our proposed MCTS-based strategy require substantial time for sample filtering. Additionally, the GRPO training process incurs significant computational overhead due to the large number of rollout samples needed for target value estimation. These efficiency challenges can potentially be mitigated through parallelized sample generation and processing, which we leave as an avenue for future work.

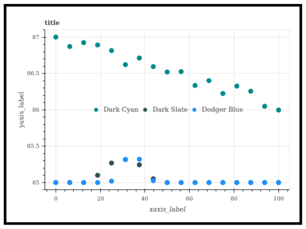
E Societal impacts

ThinkLite-VL can positively impact society by enabling more data-efficient and accessible development of advanced visual reasoning in vision-language models (VLMs). However, negative societal risks include the potential misuse of its MCTS-based sample selection insights for crafting sophisticated disinformation. Responsible development must also guard against over-reliance on these enhanced, yet still fallible, reasoning systems.

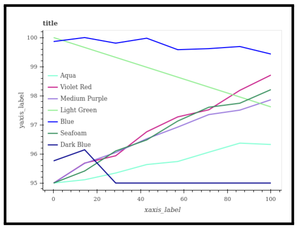
Example 3: Different difficulty samples from FigureQA



Iter0 **Question:** Is Medium Blue less than Dark Orchid?
Ground Truth Answer: Yes.



Iter29 **Question:** Does Dodger Blue intersect Dark Slate?
Ground Truth Answer: Yes.



Unsolved **Question:** Does Violet Red have the maximum area under the curve?
Ground Truth Answer: No.

Table 10: Example of samples with different difficulties decided by MCTS-based sample selection from FigureQA.

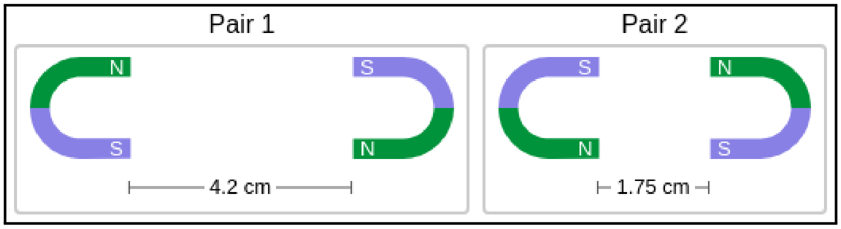
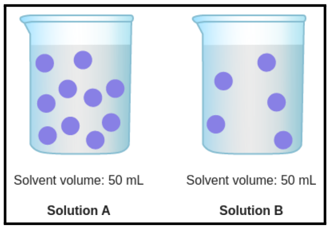
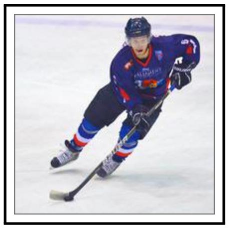
Example 4: Different difficulty samples from ScienceQA	
Iter0	 Pair 1 N S S N 4.2 cm Pair 2 S N N S 1.75 cm **Question:** Think about the magnetic force between the magnets in each pair. Which of the following statements is true? Choices: (A) The magnitude of the magnetic force is greater in Pair 2. (B) The magnitude of the magnetic force is greater in Pair 1. (C) The magnitude of the magnetic force is the same in both pairs. **Ground Truth Answer**: A.
Iter13	 Solvent volume: 50 mL Solution A Solvent volume: 50 mL Solution B **Question:** Which solution has a higher concentration of purple particles? Choices: (A) neither; their concentrations are the same (B) Solution A (C) Solution B **Ground Truth Answer**: B.
Unsolved	 **Question:** What is the direction of this push? Choices: (A) away from the hockey stick (B) toward the hockey stick **Ground Truth Answer**: A.

Table 11: Example of samples with different difficulties decided by MCTS-based sample selection from ScienceQA.

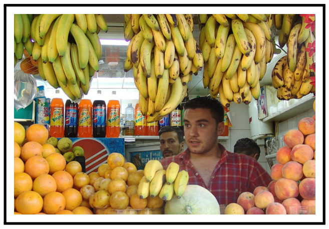
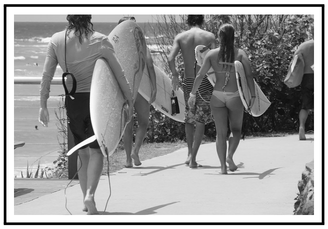
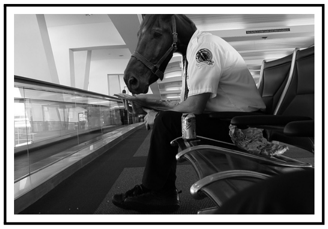
Example 5: Different difficulty samples from OK-VQA	
Iter0	 **Question:** What food group is pictured here? **Ground Truth Answer**: fruit.
Iter20	 **Question:** What is the length of the surfboard the man in the black shorts at the back of the line of people is holding? **Ground Truth Answer**: 7 feet.
Unsolved	 **Question:** What is this guy's profession? **Ground Truth Answer**: security.

Table 12: Example of samples with different difficulties decided by MCTS-based sample selection from OK-VQA.

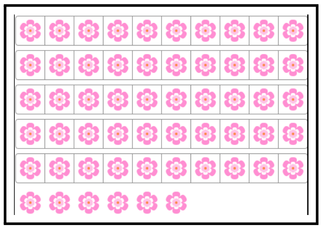
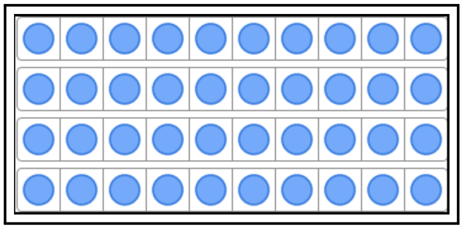
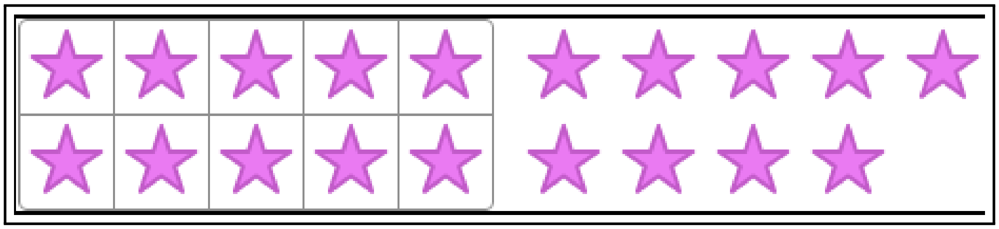
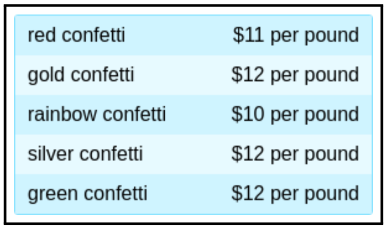
Example 6: Different difficulty samples from IconQA	
Iter0	 **Question:** How many flowers are there? **Ground Truth Answer**: 56.
Iter10	 **Question:** How many dots are there? **Ground Truth Answer**: 40.
Unsolved	 **Question:** How many stars are there? **Ground Truth Answer**: 19.

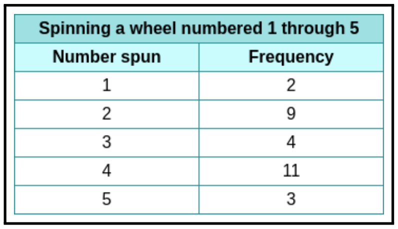
Table 13: Example of samples with different difficulties decided by MCTS-based sample selection from IconQA.

Example 7: Different difficulty samples from TabMWP



red confetti	$11 per pound
gold confetti	$12 per pound
rainbow confetti	$10 per pound
silver confetti	$12 per pound
green confetti	$12 per pound

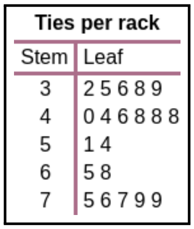
Iter0 **Question:** Adriana wants to buy 3 pounds of silver confetti. How much will she spend?
Ground Truth Answer: 36.



Spinning a wheel numbered 1 through 5

Number spun	Frequency
1	2
2	9
3	4
4	11
5	3

Iter22 **Question:** A game show viewer monitors how often a wheel numbered 1 through 5 stops at each number. How many people are there in all?
Ground Truth Answer: 29.

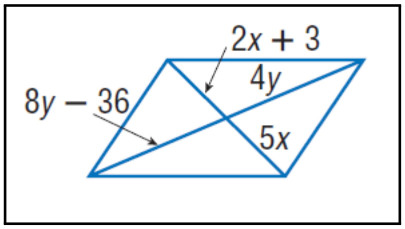


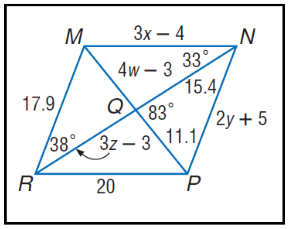
Ties per rack

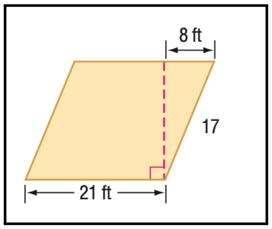
Stem	Leaf
3	2 5 6 8 9
4	0 4 6 8 8 8
5	1 4
6	5 8
7	5 6 7 9 9

Unsolved **Question:** The employee at the department store counted the number of ties on each tie rack. How many racks have at least 30 ties but fewer than 70 ties?
Ground Truth Answer: 15.

Table 14: Example of samples with different difficulties decided by MCTS-based sample selection from TabMWP.

<table>
<tr><th colspan="2">Example 1: Different difficulty samples from Geometry3K</th></tr>
<tr><td>Iter0</td><td>


Question: Find y so that the quadrilateral is a parallelogram.
Ground Truth Answer: 9.
</td></tr>
<tr><td>Iter16</td><td>


Question: Use parallelogram M N P R to find y.
Ground Truth Answer: 6.45.
</td></tr>
<tr><td>Unsolved</td><td>


Question: Find the area of the parallelogram. Round to the nearest tenth if necessary.
Ground Truth Answer: 315.
</td></tr>
</table>

Table 15: Example of samples with different difficulties decided by MCTS-based sample selection from GeoQA.

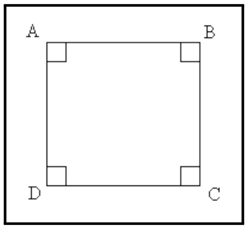
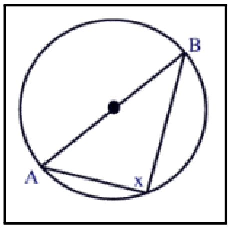
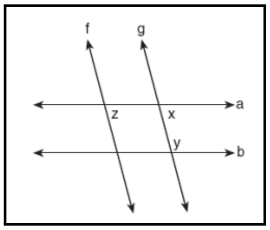
Example 2: Different difficulty samples from Geos	
Iter0	 **Question:** What is the area of the following square, if the length of BD is $2 * \sqrt{2}$? Choices: (A) 1 (B) 2 (C) 3 (D) 4 (E) 5. **Ground Truth Answer**: D.
Iter7	 **Question:** Given the circle at the right with diameter AB, find x. Choices: (A) 30 degrees (B) 45 degrees (C) 60 degrees (D) 90 degrees (E) None **Ground Truth Answer**: D.
Unsolved	 **Question:** In the diagram at the right, lines f and g are parallel, and lines a and b are parallel. x = 75. What is the value of y + z? Choices: (A) 75 (B) 105 (C) 150 (D) 180 (E) None **Ground Truth Answer**: D.

Table 16: Example of samples with different difficulties decided by MCTS-based sample selection from Geos.