
To Think or Not To Think: A Study of Thinking in Rule-Based Visual Reinforcement Fine-Tuning

Ming Li¹ Jike Zhong² Shitian Zhao¹ Yuxiang Lai³ Haoquan Zhang^{1,4} Wang Bill Zhu²
Kaipeng Zhang^{1*}

¹Shanghai AI Laboratory

²University of Southern California

³ Emory University

⁴ Chinese University of Hong Kong

lm1640362161@gmail.com, zhangkaipeng@pjlab.org.cn

Project code: <https://github.com/minglllli/CLS-RL>

Abstract

This paper investigates the role of explicit thinking process in rule-based reinforcement fine-tuning (RFT) for multi-modal large language models (MLLMs). We first extend *Thinking-RFT* to image classification task, using verifiable rewards for fine-tuning (FT). Experiments show Thinking-RFT significantly outperforms supervised FT and yields a cross-dataset generalization effect. We then rethink and question whether explicit thinking in RFT is always necessary and beneficial. Challenging the convention that explicit thinking is crucial for the success of RFT, we introduce *No-Thinking-RFT*, exploring RFT without thinking by introducing a simple equality accuracy reward. We evaluate No-Thinking-RFT on six diverse tasks across different model sizes and types. Experiment results reveal four key findings: **(1)**. Visual perception tasks do not require thinking during RFT, as No-Thinking-RFT consistently outperforms or matches Thinking-RFT across model sizes and types. **(2)**. Models with limited capabilities struggle to generate high-quality CoT for RFT, making Thinking-RFT less effective than No-Thinking-RFT. **(3)**. There are inconsistencies between the answers in the thinking tags and answer tags for some responses of Thinking-RFT, which show lower average accuracy than the overall accuracy. **(4)**. The performance gain of No-Thinking-RFT mainly stems from improved learning during no thinking FT and the avoidance of inference overthinking, as evidenced by the partial gains from appending empty thinking tags at inference time of Thinking-RFT. We hypothesize that explicit thinking before verifiable answers may hinder reward convergence and reduce performance in certain scenarios. To test this, we propose *Think-After-Answer*, which places thinking after the answer to mitigate this effect for experimental verification. Lastly, we conduct a pilot study to explore whether MLLMs can learn when to think during RFT, introducing an *Adaptive-Thinking* method. Experiments show that model converges to either thinking or not depending on model capability, achieving comparable or better performance than both Thinking and No-Thinking-RFT. Our findings suggest MLLMs can adaptively decide to think or not based on their capabilities and task complexity, offering insights into the thinking process in RFT.

1 Introduction

Recently, rule-based reinforcement fine-tuning (RFT) has made significant progress and achieved better performance than traditional supervised fine-tuning (SFT) [19, 68, 26]. RFT leverages verifiable

*Corresponding Author: zhangkaipeng@pjlab.org.cn

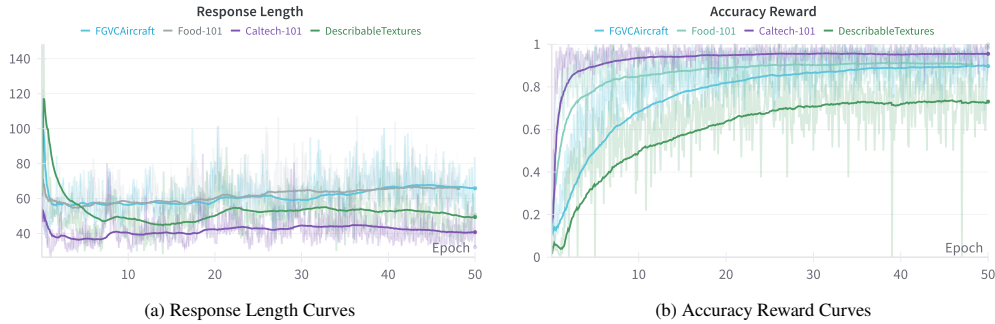


Figure 1: **Thinking is less essential:** Classification RFT exhibits sharp drops in response length at specific steps, accompanied by significant rises in accuracy reward.

rewards for training, encouraging models to engage in a thinking process before answering for solution exploration [19]. The explicit thinking is widely believed as a key factor in RFT’s success, and many works on multi-modal RFT [82, 24] aim to reproduce the length-increasing and ‘aha moment’ effects seen in Deepseek-R1 [19]. However, a critical question emerges as RFT being widely used: *Is explicit thinking always necessary and beneficial for small-sized-model RFT?* Recent studies [27, 64, 65] suggest that reasoning offers limited gains on commonsense tasks and overthinking may even harm inference performance. While these findings offer insight into model reasoning, they focus *solely on inference*, leaving the *impact of explicit thinking during RFT unclear and largely unexplored*. Furthermore, RFT typically requires significantly more fine-tuning time and GPU memory than SFT due to the generation of multiple lengthy responses [19]. Consequently, the role of explicit thinking during RFT warrants further exploration, considering both accuracy performance and computational efficiency.

In this paper, we investigate the thinking process in rule-based RFT for MLLMs across different tasks and model sizes. We begin with a case study exploring closed-form MLLM image classification. Motivated by the success of rule-based RFT [19, 68, 77] in LLM fine-tuning, we extend thinking-based RFT (Thinking-RFT) to few-shot classification fine-tuning. Thinking-RFT fine-tunes MLLMs using class labels as verifiable answers for reward calculation. Extensive experiments show that Thinking-RFT performs much better than SFT on both in-domain learning and new-class generalization. Additionally, we observe a *free-lunch phenomenon*: fine-tuning MLLMs on one dataset with Thinking-RFT improves performance on other datasets, despite shift in data distribution and entirely different class names. This phenomenon validates that rule-based RFT can effectively teach models the fundamentals of image classification rather than simply memorizing [8].

We then revisit the role of thinking process in rule-based RFT, a key factor in the success of Deepseek-R1 [19]. Unlike the gradual increase in response length of math problems observed in [19], classification RFT exhibits sharp drops in response length at specific steps, accompanied by significant rises in accuracy reward (Figure 1). Our finding that thinking is less essential for classification, prompting models to adopt minimal reasoning, aligns with recent studies [27, 64] showing that overthinking during inference can hinder performance on certain tasks. To this end, we propose No-Thinking-RFT, a rule-based RFT approach without explicit thinking process. No-Thinking RFT utilizes a direct-answer instruction prompt and removes the format reward. It implements a strict equality accuracy reward, granting a score of 1 solely when the output precisely matches the labels, thereby effectively discouraging reasoning during fine-tuning. Notably, in few-shot classification task, No-Thinking-RFT outperforms Thinking-RFT while achieving substantially shorter fine-tuning and inference time (Sec.4.4) and faster convergence (Sec.4.5) than Thinking-RFT.

Next, we evaluate No-Thinking-RFT on general visual reasoning tasks, including math, spatial reasoning, puzzles, referring grounding, and detection, across different model sizes. Results reveal four important findings: **(1):** Thinking is unnecessary for visual perception tasks during RFT. Across all test model sizes, No-Thinking-RFT matches or outperforms Thinking-RFT for these tasks, showing that excluding thinking in RFT can preserve or improve performance while enhancing training and inference efficiency. **(2):** Models with limited capability (e.g., 2B) tend to converge to produce trivial reasoning under Thinking-RFT (refer to Figure 15 ~ Figure 20 for examples), leading to worse performance and longer fine-tuning time compared to No-Thinking-RFT. **(3):** For mid-sized models

(e.g., 7B), Thinking-RFT often produces inconsistencies between the content in thinking tags and answer tags (see Figure 9~Figure 14). We observe frequent mismatches comparing answers from the thinking tags and answer tags, with the average accuracy amongst inconsistent responses lower than the overall accuracy (Figure 2), which suggests that maintaining response consistency could improve performance. **(4):** The performance gain of No-Thinking-RFT over Thinking-RFT mainly stems from improved fine-tuning and the avoidance of inference overthinking. When we test Thinking-RFT in a no-thinking mode by appending empty thinking tags during inference, performance improves on perception tasks but still lags behind No-Thinking-RFT. This suggests that both enhanced learning during no thinking fine-tuning and reduced inference overthinking contribute to the gain.

We further study why No-Thinking-RFT outperforms Thinking-RFT in certain scenarios (e.g., 2B model on perception tasks), hypothesizing that the explicit thinking before verifiable answers may hinder the learning process during RFT, slowing accuracy reward convergence and reducing accuracy. To test this, we propose *Think-After-Answer*, a variant where reasoning occurs after generating verifiable answers. Experiment results support our hypothesis: in tasks where No-Thinking-RFT outperforms Thinking-RFT, Think-After-Answer converges faster and achieves higher accuracy than Thinking-RFT. However, it still underperforms No-Thinking-RFT, suggesting that reasoning, even when placed after answers, can remain detrimental during RFT in certain scenarios.

Finally, we conduct a pilot study to investigate whether MLLMs can learn to adaptively decide when to think during RFT. We introduce *Adaptive-Thinking*: models are prompted to first assess if a problem required reasoning or thinking. If reasoning was deemed necessary, the model would generate a thinking process before answering; otherwise, it would answer directly. Our experiments revealed that models consistently converged to a single response strategy (either always thinking or never thinking). Notably, Adaptive-Thinking achieves comparative or better performance than the better one of Thinking-RFT and No-Thinking-RFT, and the final response strategy always corresponded to one suited for that specific model size and task complexity. For example, on math tasks, the 2B model converged to the non-thinking answering response, whereas the 7B model consistently adopted the thinking process before answering. These findings suggest that MLLMs may possess the capability to learn whether to employ reasoning based on their inherent abilities and task complexity during RFT.

Our contributions can be summarized as follows:

- We extend Thinking-RFT to few-shot MLLM classification and reveal a cross-dataset transfer across datasets.
- We show several important findings about No-Thinking-RFT and Thinking-RFT via experiments on six tasks with 2B–7B models.
- Through Think-After-Answer and Adaptive-Thinking we confirm that deferring or omitting CoT speeds convergence without harming accuracy and explore adaptive thinking strategy.

2 RFT Methods: Think vs. No Think

2.1 Thinking-RFT

Optimization Algorithm. We follow Deepseek-R1 [19, 59] to employ Group Relative Policy Optimization (GRPO) as the RL algorithm for optimization in our study, since it is the most widely used one. We refer readers to Appendix A for a brief introduction of the technical details of GRPO.

Instruction prompt. Following Deepseek-R1 [59], we utilize a prompt that encourages models to first engage in a thinking process before generating the final answers. The prompt is designed as: *{Question} Please output the thinking process in <think> </think> and final answer in <answer> </answer> tags.* Here *{Question}* will be replaced by each specific question.

Reward Function. To clearly examine the RFT thinking process and promote the generalizability of our findings, we employ the simplest binary reward function for our study. The reward function is composed with two parts: format and accuracy reward. The format reward R_{format} is to check if the responses follow the format correctly. $R_{\text{format}} = 1$ if the response format is correct, and 0 if it is incorrect. The accuracy reward R_{accuracy} checks whether the answer in the answer tag matches the ground truth. For example, R_{accuracy} verifies the extracted answer against the correct choice in multi-choice problems, class labels in classification tasks, and numeric results in math problems. $R_{\text{accuracy}} = 1$ if the extracted answer matches correctly, and 0 otherwise.

Training Strategy. To study explicit thinking during RFT clearly, we adopt R1-zero [19] training for all methods in our study, i.e., we apply RL to all base models without any SFT, following [80].

2.2 No-Thinking-RFT

We explore rule-based RFT without a thinking process and propose a No-Thinking-RFT method. The instruction prompt and reward functions are designed as below:

Instruction prompt. Instead of the prompt in Thinking-RFT which encourages models to think before answering, the prompt in the No-Thinking-RFT method prohibits the model from thinking. The prompt is designed as: {Question} Please directly output the answer.

Reward Function. No-Thinking-RFT eliminates the format reward and relies solely on the accuracy reward rather than combining two rewards. The accuracy reward R_{accuracy} checks whether the model’s output matches the ground truth exactly. $R_{\text{accuracy}} = 1$ if the model response matches the ground truth and 0 otherwise. The equality-based matching reward forces the model to bypass any thinking process and output only the answers, which are significantly shorter than the typical reasoning responses in Thinking-RFT. As a result, training No-Thinking-RFT requires substantially less GPU memory, and its training and inference times are markedly shorter than those of Thinking-RFT (Sec. 4.4).

3 General Visual Reasoning

In this section, we introduce results on general visual reasoning, including spatial understanding, math, puzzle, referring grounding, and detection across different model sizes. We report the results of spatial understanding, math, and puzzle tasks with 2B and 7B Qwen2-VL-Instruct [72] models in main text, and leave the results of referring grounding, detection tasks and other different model types (e.g., InternVL2.5 [5] and Qwen2-5-VL-Instruct [1]) in Appendix G. We also discuss offline DPO, No-Thinking-RFT, and Thinking-RFT in Appendix H.2. We first introduce Think-After-Answer and Adaptive-Thinking methods and then report experimental results.

3.1 Think-After-Answer & Adaptive-Thinking

Think-After-Answer. We explore the reason that No-Thinking-RFT outperforms Thinking-RFT under specific tasks and model sizes. As shown in Sec. 4.5, Thinking-RFT converges slower than No-Thinking-RFT. Therefore, we hypothesize that for the 2B model, the explicit “think” process is detrimental, whereas for the 7B model, it is not universally required, as verifiable answers are already conditioned on thinking. To verify this hypothesis, we propose Think-After-Answer by letting MLLMs first answer the questions and then output a brief reasoning process, therefore alleviating the negative impact of explicit thinking on verifiable answers during RFT, as now the thinking is conditioned on answers. The prompt is provided in Appendix E.2 and the format and accuracy reward do not change. If the hypothesis holds, the convergence speed of Think-After-Answer should be faster and the final performance should be higher than Thinking-RFT in cases where No-Thinking-RFT outperforms Thinking-RFT.

Adaptive-Thinking. We propose an Adaptive-Thinking method that lets MLLMs decide whether to think by themselves during RFT. Specifically, we prompt MLLMs to first determine whether a problem requires thinking, and then either output a reasoning process if needed or directly output an answer if not (refer to Appendix E.3 for the prompt). A response in either thinking format or direct answering format will receive a format reward of 1. The accuracy reward is not changed as before.

3.2 Experimental Setup

We focus on three main visual or multi-modal tasks: visual perception (classification and spatial grounding), multi-modal math reasoning, and visual puzzle reasoning in the main text. For a fair comparison, we set hyperparameters the same for all methods. We provide a brief overview of the experimental settings below, with detailed implementation described in Appendix D.3.

Visual Perception. We follow [82] to fine-tune models on SAT dataset [57] 2 epochs and then test on CVBench dataset [69]. We also include classification results for comparison. The number of rollout

Table 1: Results of Thinking-RFT, Think-After-Answer, No-Thinking-RFT, and Adaptive-Thinking on CVBench.

Model	Method	Overall	Count	Relation	Depth	Distance	FT Time
2B	Thinking-RFT	70.36	66.12	83.38	68.50	63.67	599 m
	Think-After-Answer	73.65	68.65	82.15	74.83	69.83	408 m
	No-Thinking-RFT	76.76	69.67	84.46	80.67	73.83	139 m
	Adaptive-Thinking	77.03	69.92	86.31	76.50	76.83	208 m
7B	Thinking-RFT	80.36	66.24	90.92	87.33	80.00	651 m
	Think-After-Answer	81.61	66.75	90.46	87.33	85.83	428 m
	No-Thinking-RFT	80.67	66.50	92.15	83.83	83.67	155 m
	Adaptive-Thinking	81.65	67.38	90.31	88.00	84.67	639 m

Table 2: Performance comparison of PuzzleVQA and AlgoPuzzleVQA across 2B and 7B models.

Method	2B		7B	
	PuzzleVQA	AlgoPuzzleVQA	PuzzleVQA	AlgoPuzzleVQA
Thinking-RFT	52.50	27.72	66.60	24.78
Think-After-Answer	64.70	26.94	80.45	28.11
No-Thinking-RFT	70.85	29.17	80.65	29.39
Adaptive-Thinking	75.45	27.94	85.05	29.00

is set to 4 for both 2B and 7B models. The number of rollout is set to 4 for both 2B and 7B models. β is set to 0.04, learning rate is set to 1×10^{-6} . The temperature is set to 1 for all experiments.

Multi-Modal Math Reasoning. We utilize the Math-40K [61] as the find-tuning data and fine-tune models 1 epoch. Then we test the fine-tuned models on both MathVista [43] and MathVision [71]. The maximum length of input prompt and response are set to 4096 and 512 respectively. The number of rollout is set to 8 for 2B model, and 4 for 7B model. β is set to 0.04, learning rate is set to 1×10^{-6} .

Visual Puzzle Reasoning. We follow the code of [7] to generate a training dataset with 6.5k data and fine-tune models 2 epochs. We then test fine-tuned models on PuzzleVQA [7] as in-domain testing, and on AlgoPuzzleVQA [17] as out-of-domain testing. β is set to 0.04, learning rate is set to 1×10^{-6} . The batch size is set to 1 per GPU and we use 2-step gradient accumulation during training.

3.3 Results & Findings

We compare the results of four different thinking strategies here and show examples of responses in Figure 21~Figure 25. We report the results of visual perception on Table 1, math reasoning on Table 3, and puzzle reasoning on Table 2. We analyze these results and discuss several findings below.

Finding 1: For small models with weak ability (e.g., 2B size), Thinking-RFT may lead to trivial reasoning and inferior performance compared with No-Thinking. We observe that for 2B models, by using much less fine-tuning time, No-Thinking-RFT outperforms Thinking-RFT in most benchmarks by a large margin, including even Mathvista and except only MathVision. We find that for Mathvista, No-Thinking-RFT outperforms Thinking-RFT on all tasks except math world problems (MWP). It is reasonable that Thinking-RFT outperforms No-Thinking-RFT in MWP and MathVision since these problems need much computation for obtaining final answers and it is hard to directly output correct answers. When looking deeper into model responses under Thinking-RFT, we find that most reasoning is trivial, i.e., it does not contribute meaningfully to the final answer, especially for complex reasoning tasks (refer to Figures 15~Figure 20 for qualitative examples). These results suggest that when the capability of models is poor, the thinking process during RFT may converge to trivial solutions and finally lead to inferior performance compared with No-Thinking.

Finding 2: Visual perception and puzzle tasks do not need thinking. We find that No-Thinking-RFT outperforms Thinking-RFT on spatial reasoning, classification (7B results in Table 12), and puzzle tasks across both 2B and 7B models, while Thinking-RFT performs better on math reasoning with the 7B model. We notice that although puzzle tasks typically require complex reasoning, No-Thinking-RFT outperforms Thinking-RFT on both 2B and 7B models. This may be because these tasks rely on visual rather than linguistic reasoning, and language-based reasoning might cause

Table 3: Accuracy (%) of 2B and 7B models on MathVista sub-tasks and MathVision overall. TQA: Textbook QA, VQA: Visual QA, Geo: Geometry, MWP: Math Word Problem, FQA: Figure QA.

Model	Method	MathVista					MathVision	
		Overall	TQA	VQA	Geo	MWP	FQA	
2B	Thinking-RFT	44.90	46.84	36.87	37.02	52.15	50.19	16.45
	Think-After-Answer	48.50	51.27	44.69	36.54	50.54	57.25	14.80
	No-Thinking-RFT	48.80	52.53	46.37	38.94	49.46	55.39	13.49
	Adaptive-Thinking	50.20	50.00	51.96	37.98	52.69	56.88	13.16
7B	Thinking-RFT	64.60	65.19	59.22	60.58	68.28	68.40	21.71
	Think-After-Answer	62.00	63.92	58.10	58.65	63.98	64.68	21.71
	No-Thinking-RFT	59.10	65.19	58.66	55.77	56.45	60.22	18.09
	Adaptive-Thinking	64.70	64.56	56.42	59.13	74.73	67.66	21.05

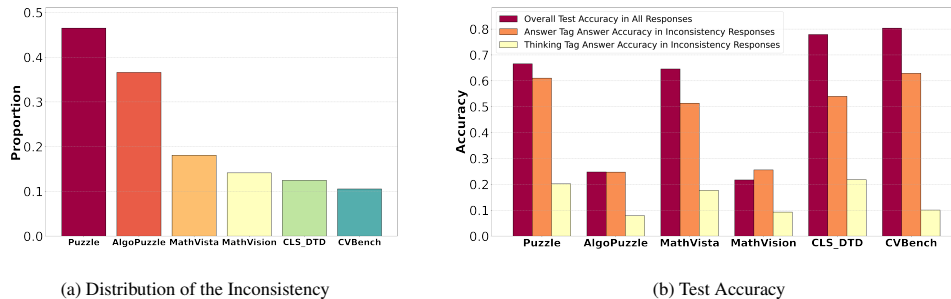


Figure 2: Comparison of inconsistency proportion and test accuracy among different datasets.

hallucinations, or the 7B model’s puzzle-solving capability may still be limited. In summary, visual perception and puzzle reasoning tasks do not need reasoning among test model sizes, and RFT without thinking is beneficial to both accuracy performance and computational efficiency.

Finding 3: There are inconsistencies between thinking and answer content of some responses.

We use GPT-4o [25] to extract answers from the thinking tag and compare them with those in the answer tag (see Appendix E.1 for details). We find inconsistencies in some responses between the thinking and answer tags. Figure 2 visualizes the proportion of inconsistent responses per dataset and their respective accuracies in both tags. We also show examples of inconsistent responses in Figure 9~Figure 14. We found that **1).** Inconsistencies are more common in reasoning tasks such as puzzles than in perception tasks like classification. **2).** For inconsistent responses, accuracy of answers in answer tags is much higher than in thinking tags, and both are below the overall accuracy. These results suggest that encouraging more consistent responses could improve performance, particularly on reasoning tasks. We leave strategies for enhancing consistency to future work.

Finding 4: Performance gain of No-Thinking-RFT over Thinking-RFT on certain tasks stems from improved learning during RFT and avoidance of inference overthinking.

We test the No-Thinking inference mode of Thinking-RFT by appending an empty thinking tag during inference and show the results on Table 17. We find that while appending empty thinking tags during inference could improve performance on CVBench and puzzle tasks, its performance is still far behind No-Thinking-RFT. This suggests that the gain of No-Thinking-RFT primarily stems from two factors: improved learning during fine-tuning and the avoidance of overthinking during inference.

Finding 5: Explicit thinking before verifiable answers may leads to slower reward convergence and inferior performance.

We observe that Think-After-Answer outperforms Thinking-RFT in perception and puzzle tasks. We also visualize the accuracy-reward curves on the SAT and Puzzle datasets in Figure 3 (2B) and Figure 8 (7B), where Think-After-Answer achieves faster reward convergence and higher final accuracy than Thinking-RFT. These results indicate that placing explicit CoT after verifiable answers during RFT can alleviate its negative impact, improving both convergence speed and performance, thus validating our hypothesis that explicit thinking before answers may lead to slower reward convergence and reduced performance on perception and puzzle tasks.

Finding 6: MLLMs can adaptively learn whether to think on task level during RFT on task level.

We find that Adaptive-Thinking performs comparably to or better than the stronger of Thinking-RFT

Table 4: Parameter Difference (L_1 Norm)

Dataset	Method	Visual	Language	MLP_V	norm_V	attn_V	MLP_L	norm_L	attn_L
DTD	Thinking	0.267	0.434	0.206	0.001	0.158	0.372	0.000	0.153
	No-Thinking	0.237	0.402	0.184	0.001	0.143	0.379	0.000	0.154
SAT	Thinking	0.503	0.712	0.398	0.003	0.311	0.614	0.000	0.256
	No-Thinking	0.518	0.718	0.398	0.003	0.327	0.704	0.000	0.281
Math	Thinking	0.415	0.735	0.364	0.002	0.210	0.691	0.000	0.270
	No-Thinking	0.731	0.931	0.581	0.004	0.455	0.880	0.000	0.388

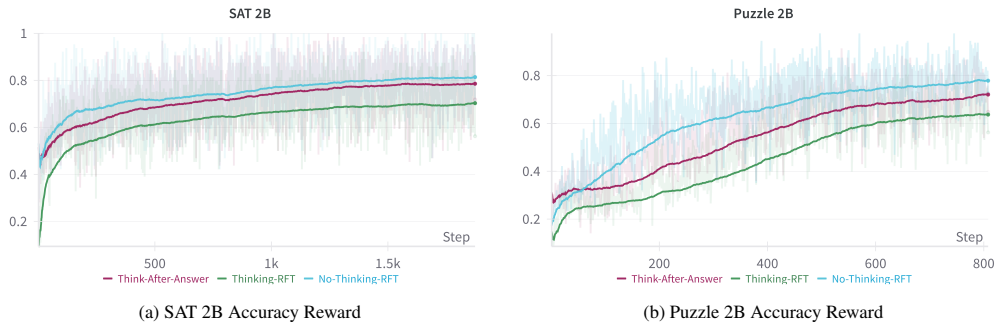


Figure 3: Comparison of accuracy reward convergence speed on SAT dataset over steps of Thinking-RFT, Think-After-Answer, and No-Thinking-RFT across 2B and 7B models.

and No-Thinking-RFT. Figure 6 shows the response length trend curves during RFT, and Table 10 summarizes the statistics of test response types. We observe that models initially produce a mix of thinking and No-Thinking responses, but gradually converge to a single response type—either think or not think and matching the better-performing strategy. For instance, the 2B model converges to No-Thinking response, while the 7B model converges to thinking response for math tasks. These results suggest that MLLMs can learn whether to think during RFT based on model capacity and task complexity. Despite promising results, the same task can present problems requiring varied responses. Ideally, Adaptive-Thinking would operate at the problem level, a refinement for future work.

3.4 Parameter Change during RL

We compare the changes in parameters by computing the L_1 norm of parameter difference. Our analysis focuses on three aspects: **1. Modality**: changes between visual and language components, **2. Module**: changes in different modules, **3. Layer**: changes across different layers.

We analyze Qwen2VL-2B on SAT, DTD, and Math datasets using Thinking and No-Thinking-RFT. Modality- and Module-level results are given in the table below. We directly discuss layer-level findings as the table is too large.

1. Modality-Level: Language weights drift more than visual weights for all dataset—and the No-Thinking strategy raises visual drift slightly. This implies that the reward gradient mainly applies to the language modality.

2. Module-Level: MLP is the dominant changed component for both visual and language parts, and attention blocks change less but still contribute near 15-20%.

3. Layer-Level: With Thinking, weight drift grows toward deeper layers as reward back-propagates through the whole reasoning chain to high-level semantics. With No-Thinking, drift peaks in early-mid layers and then declines, indicating that low-level features are reshaped so a shallow forward path already produces the reward token.

4 Case Study on Image Classification Tasks

In this section, we introduce the experiment results on classification. Recent works [81] show that the classification abilities of MLLMs [42, 36, 12, 13, 49, 6] are poor due to pre-training data limitations

Table 5: No-Thinking-RFT outperforms Thinking-RFT in 10 out of 11 datasets under few-shot learning results. S.C.: StanfordCars dataset. F.A.: FGVC Aircraft dataset.

	Caltech101	DTD	EuroSAT	Food101	ImageNet	Flowers102	OxfordPets	S.C.	SUN397	UCF101	F.A.	Average
Qwen2VL	88.56	54.79	45.68	77.54	70.8	64.43	73.89	35.77	63.83	66.22	42.75	62.21
SFT	93.91	71.336	75.16	75.75	41.60	96.87	85.80	71.13	41.66	63.81	60.15	70.65
Thinking-RFT	98.09	69.92	49.46	88.94	92.24	86.56	87.24	80.24	84.57	82.1	74.41	81.25
No-Thinking-RFT	98.46	73.52	58.02	90.78	92.31	91.6	86.13	92.5	86.72	83.82	74.41	84.39

and SFT with large-scale data could improve performance [81]. However, acquiring large-scale datasets incurs substantial cost and computational overhead [84]. Few-shot learning thus emerges as a compelling alternative, previously validated in contrastive vision-language models (VLMs) [54, 83, 84, 38, 30], but its application to MLLMs remains underexplored. We report the results of few-shot learning, free-lunch phenomenon, convergence and efficiency comparison in this section, the results of base-to-new setting in Appendix F.1, open-set classification in Appendix F.2, the results of 7B Qwen2-VL-Instruct [72] in Appendix F.3, and examples of model response in Appendix I.

4.1 Experimental Setup

Setting. In this paper, we mainly focus on closed-form classification for MLLMs, where a subset of class names is provided for selection. The question format is {Question} {Instruction prompt}, where {Question} and {Instruction prompt} will be replaced by the specific questions and instruction prompt of each dataset and method respectively.

Datasets. Following [83, 84], we conduct experiments on 11 public classification benchmarks. The names, questions, statistics of each dataset are provided in Appendix C. Due to computational resource and model input limit, we select a subset of class labels as the choice list. For few-shot classification, we sample 40% of labels including the gt labels to form the choice list in question.

Implementation Details. We utilize Qwen2-VL-2B-Instruct [72] as the base model, and fine-tune all parameters, following [82, 3]. All training is conducted in 8 A100 GPUs. The batch size is set to 1 per GPU and we use 2-step gradient accumulation during training [3, 60]. All images are resized to 328×328 resolution and no data augmentation is used. For inference test, we first extract answers from the answer tag and then verify if class names are in answers. If the answer tag does not exist in model responses we directly verify if class names are in model responses, following [81]. The temperature is set to 1.0 for all experiments. The random seed is set to 100 for all experiments.

More details about dataset construction and implementation details are provided in Appendix D.

4.2 Few-Shot Learning Results

In this subsection, we present the results of few-shot learning. We train models on 4-shot setting and report the accuracy results in Table 5 for 2B models and Table 7 for 7B models. For 2B models, we observe that Thinking-RFT markedly surpasses SFT in most datasets, resulting in a notably higher average accuracy than SFT, which implies that rule-base RFT can let model learn better downstream knowledge than SFT. We further discover that No-Thinking-RFT outperforms Thinking-RFT in 10 out of 11 datasets, ultimately achieving a 3.14% higher average accuracy compared to Thinking-RFT. For 7B models, the performance gap between Thinking-RFT and No-Thinking-RFT narrows but remains. These results indicate that RFT without the thinking process can effectively enhance model’s performance on classification than Thinking-RFT.

4.3 Free-Lunch Phenomenon

In this section, we discuss the free-lunch phenomenon. Previous work in few-shot contrastive VLM fine-tuning has demonstrated that, when fine-tuned on a specific dataset, the performance of VLMs on other datasets is drastically degraded, a phenomenon known as catastrophic forgetting [83, 30]. However, we will show that Thinking-RFT and No-Thinking-RFT can enhance the performance of MLLMs on other datasets when fine-tuned on one specific dataset. We visualize the improvement of

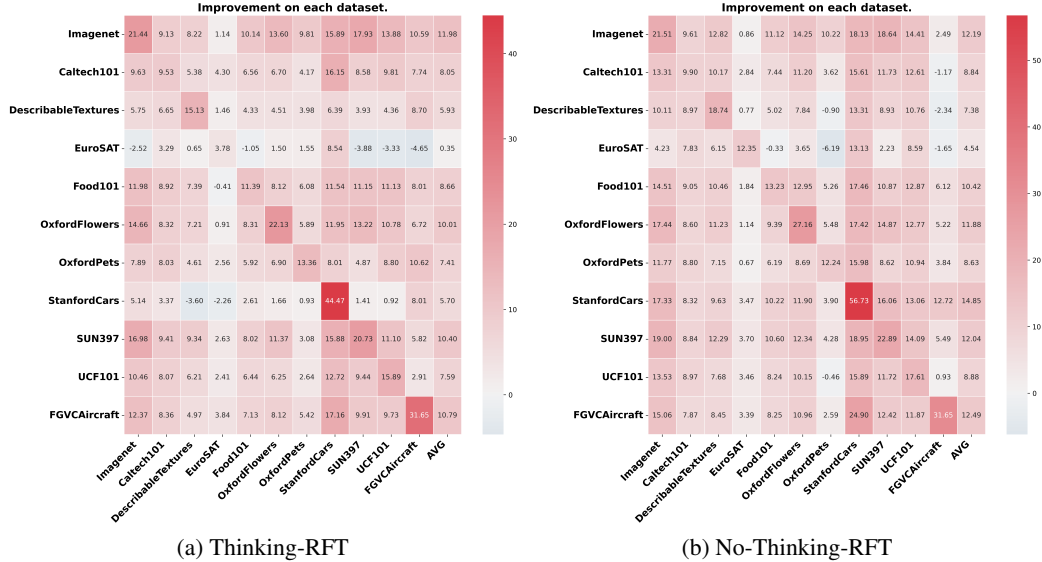


Figure 4: Free-lunch phenomenon: Both Thinking-RFT and No-Thinking-RFT lead to cross-dataset accuracy gains. Red: increase; blue: decrease; x-axis: test dataset; y-axis: fine-tuning dataset.

Table 7: Comparison of Thinking-RFT and No-Thinking-RFT with 2B and 7B models on fewshot classification. DTD: DescribableTextures.

Model	Method	DTD	EuroSAT	OxfordFlowers	StanfordCars	Average
7B	Thinking-RFT	77.90	53.17	93.91	84.19	77.29
	Think-After-Answer	76.29	62.95	94.84	89.32	80.85
	Adaptive-Thinking	79.60	56.62	96.86	89.92	80.75
	No-Thinking-RFT	80.56	58.91	94.24	94.02	81.93

Thinking-RFT and No-Thinking-RFT compared with zero-shot Qwen2VL-instruct-2B in Figure 4. We find that when fine-tuned on a specific dataset, both Thinking-RFT and No-Thinking-RFT yield improvements on other datasets in most instances, despite variations in data distribution and even completely different class lists. These results indicate that the application of rule-based verifiable signals and reward loss for model fine-tuning can effectively compel models to acquire essential classification knowledge instead of memorizing. This equips them to achieve superior performance on entirely new datasets. We discuss more about the free lunch phenomenon in Appendix F.4.

4.4 Efficiency Comparison

In this subsection, we compare the training and inference efficiency of SFT, Thinking-RFT and No-Thinking-RFT, using the ImageNet dataset as a case study. The results are presented in Table 6. We find that Thinking-RFT requires significantly more time for both training and inference compared to SFT and No-Thinking-RFT, attributable to the necessity of generating multiple lengthy responses during fine-tuning and long reasoning response before answers during inference. In contrast, SFT optimizes only the label tokens during fine-tuning, and No-Thinking-RFT compels the model to output only the ground truth labels during this phase, which significantly reduces the time required. During the inference phase, both methods are designed to output solely class labels, resulting in considerably reduced inference time.

Table 6: Thinking-RFT requires much more time (in mins) for finetuning and inference v.s. SFT and No-Thinking-RFT.

Method	FT	Infer
SFT	35	20
Thinking-RFT	1587	30
No-Thinking-RFT	94	26

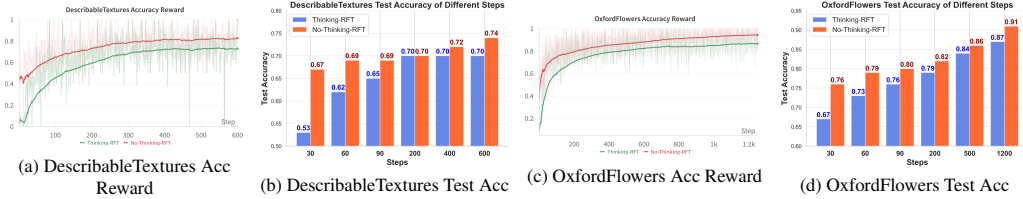


Figure 5: Comparison of accuracy reward convergence speed and test accuracy over steps between Thinking-RFT and No-Thinking-RFT. No-Thinking-RFT converges faster than Thinking-RFT.

4.5 Convergence Comparison

In this subsection, we discuss the comparison in convergence rates between Thinking-RFT and No-Thinking-RFT. We illustrate the accuracy reward at each training step and examine the test accuracy at select intervals on the DescribableTextures and OxfordFlowers datasets. The results are shown in Figure 5. We observe that No-Thinking-RFT exhibits a faster convergence speed compared to Thinking-RFT, as evidenced by a higher accuracy reward and significantly higher test accuracy in the early stages of training (within the first 30 steps). These findings imply that slower convergence speed and lower accuracy reward of Thinking-RFT leads to inferior performance than No-Thinking-RFT.

5 Related Works

Rule-based RFT has recently achieved significant progress in LLMs [19, 26, 68] and show superior performance compared with SFT. To extend this success to MLLMs, numerous studies have been proposed [41, 60, 82, 24, 46, 3], aiming to replicate phenomena observed in Deepseek-R1, such as increasing response length and emergence of ‘aha’ moments. Although these initial efforts offer encouraging outcomes, none have thoroughly investigated the role of thinking processes during RFT. Recent research into LLM and MLLM inference suggests that overthinking can degrade performance [27, 64], providing valuable insights into model reasoning. However, these studies exclusively examine the inference stage and do not explore the impact of thinking during the fine-tuning process. As a result, the role of the explicit thinking process in RFT remains unclear. In this paper, we study the effect of explicit thinking during RFT and conduct extensive experiments across different model sizes and tasks. More detailed related works are provided in Appendix B.

6 Conclusion

In this paper, we conduct a comprehensive study on the explicit thinking in RFT. We begin by extending Thinking-RFT to classification. Experiments show that Thinking-RFT performs much better than SFT on both base-to-new and few-shot settings. Furthermore, we observed a free-lunch phenomenon with classification RFT, wherein the performance of a model fine-tuned on one dataset improves on other datasets. We then delve into the thinking process of RFT. Inspired by recent research on inference overthinking and observations regarding the decreasing response length of classification RFT, we introduce No-Thinking-RFT, which compels the model to bypass thinking. We conduct extensive experiments on various visual reasoning tasks across different model sizes to evaluate Thinking and No-Thinking-RFT. The results reveal four key findings about the inconsistency in model responses, the impact and the necessity of thinking across different tasks and model sizes, and the stem of No-Thinking-RFT’s performance gain. Finally, we propose Thinking-After-Answer and Adaptive-Thinking to explore whether MLLMs can learn when to think or not during RFT.

Limitations This study has two main limitations. First, computational constraints restricted our experiments to models 7B and smaller; thus, the applicability of our findings to larger models remains unverified. Second, our Adaptive-Thinking method converges to a single response mode (thinking or No-Thinking) at the task level, rather than adapting at the more granular problem level. Designing methods to achieve this problem-level adaptation is a direction for future work.

Acknowledgments

This paper is supported by the National Key R&D Program of China No.2022ZD0160101.

References

- [1] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- [2] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101—mining discriminative components with random forests. In *Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part VI 13*, pages 446–461. Springer, 2014.
- [3] L. Chen, L. Li, H. Zhao, Y. Song, and Vinci. R1-v: Reinforcing super generalization ability in vision-language models with less than \$3. <https://github.com/Deep-Agent/R1-V>, 2025. Accessed: 2025-02-02.
- [4] X. Chen, J. Xu, T. Liang, Z. He, J. Pang, D. Yu, L. Song, Q. Liu, M. Zhou, Z. Zhang, et al. Do not think that much for $2+3=?$ on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.
- [5] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
- [6] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.
- [7] Y. K. Chia, V. T. Y. Han, D. Ghosal, L. Bing, and S. Poria. Puzzlevqa: Diagnosing multi-modal reasoning challenges of language models with abstract visual patterns. *arXiv preprint arXiv:2403.13315*, 2024.
- [8] T. Chu, Y. Zhai, J. Yang, S. Tong, S. Xie, D. Schuurmans, Q. V. Le, S. Levine, and Y. Ma. Sft memorizes, rl generalizes: A comparative study of foundation model post-training. *arXiv preprint arXiv:2501.17161*, 2025.
- [9] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3606–3613, 2014.
- [10] A. Cuadron, D. Li, W. Ma, X. Wang, Y. Wang, S. Zhuang, S. Liu, L. G. Schroeder, T. Xia, H. Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.
- [11] C. N. De Sabbata, T. R. Sumers, B. AlKhamissi, A. Bosselut, and T. L. Griffiths. Rational metareasoning for large language models. *arXiv preprint arXiv:2410.05563*, 2024.
- [12] DeepMind. Gemini 2.0 flash experimental. <https://deepmind.google/technologies/gemini/flash/>, 2024. Accessed: 2024-12-25.
- [13] DeepMind. Gemini 2.0 flash thinking. <https://deepmind.google/technologies/gemini/flash-thinking/>, 2025. Accessed: 2025-01-21.
- [14] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

- [16] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.
- [17] D. Ghosal, V. T. Y. Han, C. Y. Ken, and S. Poria. Are language models puzzle prodigies? algorithmic puzzles unveil serious challenges in multimodal reasoning. *arXiv preprint arXiv:2403.03864*, 2024.
- [18] X. Guan, L. L. Zhang, Y. Liu, N. Shang, Y. Sun, Y. Zhu, F. Yang, and M. Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.
- [19] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- [20] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [21] P. Helber, B. Bischke, A. Dengel, and D. Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.
- [22] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *arXiv preprint arXiv:1704.04861*, 2017.
- [23] J. Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 2025.
- [24] W. Huang, B. Jia, Z. Zhai, S. Cao, Z. Ye, F. Zhao, Z. Xu, Y. Hu, and S. Lin. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*, 2025.
- [25] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.
- [26] A. Jaech, A. Kalai, A. Lerer, A. Richardson, A. El-Kishky, A. Low, A. Helyar, A. Madry, A. Beutel, A. Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- [27] D. Jiang, R. Zhang, Z. Guo, Y. Li, Y. Qi, X. Chen, L. Wang, J. Jin, C. Guo, S. Yan, et al. Mme-cot: Benchmarking chain-of-thought in large multimodal models for reasoning quality, robustness, and efficiency. *arXiv preprint arXiv:2502.09621*, 2025.
- [28] B. Joshi, Z. Liu, S. Ramnath, A. Chan, Z. Tong, S. Nie, Q. Wang, Y. Choi, and X. Ren. Are machine rationales (not) useful to humans? measuring and improving human utility of free-text rationales. *arXiv preprint arXiv:2305.07095*, 2023.
- [29] Y. Kang, X. Sun, L. Chen, and W. Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 24312–24320, 2025.
- [30] M. U. Khattak, H. Rasheed, M. Maaz, S. Khan, and F. S. Khan. Maple: Multi-modal prompt learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19113–19122, 2023.
- [31] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.
- [32] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1097–1105, 2012.

- [33] X. Lai, Z. Tian, Y. Chen, Y. Li, Y. Yuan, S. Liu, and J. Jia. Lisa: Reasoning segmentation via large language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9579–9589, 2024.
- [34] Y. Lai, J. Zhong, M. Li, S. Zhao, and X. Yang. Med-r1: Reinforcement learning for generalizable medical reasoning in vision-language models. *arXiv preprint arXiv:2503.13939*, 2025.
- [35] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [36] J. Li, D. Li, C. Xiong, and S. Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.
- [37] M. Li, Q. Li, and Y. Wang. Class balanced adaptive pseudo labeling for federated semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16292–16301, 2023.
- [38] M. Li, J. Zhong, C. Li, L. Li, N. Lin, and M. Sugiyama. Vision-language model fine-tuning via simple parameter-efficient modification. *arXiv preprint arXiv:2409.16718*, 2024.
- [39] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.
- [40] R. Liu, J. Geng, A. J. Wu, I. Sucholutsky, T. Lombrozo, and T. L. Griffiths. Mind your step (by step): Chain-of-thought can reduce performance on tasks where thinking makes humans worse. *arXiv preprint arXiv:2410.21333*, 2024.
- [41] Z. Liu, Z. Sun, Y. Zang, X. Dong, Y. Cao, H. Duan, D. Lin, and J. Wang. Visual-rft: Visual reinforcement fine-tuning. *arXiv preprint arXiv:2503.01785*, 2025.
- [42] H. Lu, W. Liu, B. Zhang, B. Wang, K. Dong, B. Liu, J. Sun, T. Ren, Z. Li, H. Yang, et al. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*, 2024.
- [43] P. Lu, H. Bansal, T. Xia, J. Liu, C. Li, H. Hajishirzi, H. Cheng, K.-W. Chang, M. Galley, and J. Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.
- [44] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.
- [45] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20, 2016.
- [46] F. Meng, L. Du, Z. Liu, Z. Zhou, Q. Lu, D. Fu, B. Shi, W. Wang, J. He, K. Zhang, et al. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *arXiv preprint arXiv:2503.07365*, 2025.
- [47] Y. Meng, M. Xia, and D. Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.
- [48] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.
- [49] OpenAI. Gpt-4o system card. <https://cdn.openai.com/gpt-4o-system-card.pdf>, 2024. Accessed: 2024-09-26.
- [50] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

- [51] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012.
- [52] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.
- [53] Y. Peng, G. Zhang, M. Zhang, Z. You, J. Liu, Q. Zhu, K. Yang, X. Xu, X. Geng, and X. Yang. Lmm-r1: Empowering 3b llms with strong reasoning abilities through two-stage rule-based rl. *arXiv preprint arXiv:2503.07536*, 2025.
- [54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021.
- [55] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021.
- [56] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.
- [57] A. Ray, J. Duan, R. Tan, D. Bashkurova, R. Hendrix, K. Ehsani, A. Kembhavi, B. A. Plummer, R. Krishna, K.-H. Zeng, et al. Sat: Spatial aptitude training for multimodal language models. *arXiv preprint arXiv:2412.07755*, 2024.
- [58] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- [59] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, X. Bi, H. Zhang, M. Zhang, Y. Li, Y. Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- [60] H. Shen, P. Liu, J. Li, C. Fang, Y. Ma, J. Liao, Q. Shen, Z. Zhang, K. Zhao, Q. Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.
- [61] W. Shi, Z. Hu, Y. Bin, J. Liu, Y. Yang, S.-K. Ng, L. Bing, and R. K.-W. Lee. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*, 2024.
- [62] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [63] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [64] Z. Sprague, F. Yin, J. D. Rodriguez, D. Jiang, M. Wadhwa, P. Singhal, X. Zhao, X. Ye, K. Mahowald, and G. Durrett. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *arXiv preprint arXiv:2409.12183*, 2024.
- [65] Y. Sui, Y.-N. Chuang, G. Wang, J. Zhang, T. Zhang, J. Yuan, H. Liu, A. Wen, H. Chen, X. Hu, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.
- [66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.
- [67] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning (ICML)*, pages 6105–6114, 2019.
- [68] K. Team, A. Du, B. Gao, B. Xing, C. Jiang, C. Chen, C. Li, C. Xiao, C. Du, C. Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

- [69] P. Tong, E. Brown, P. Wu, S. Woo, A. J. V. IYER, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *Advances in Neural Information Processing Systems*, 37:87310–87356, 2024.
- [70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need, 2023.
- [71] K. Wang, J. Pan, W. Shi, Z. Lu, H. Ren, A. Zhou, M. Zhan, and H. Li. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169, 2024.
- [72] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.
- [73] P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *arXiv preprint arXiv:2312.08935*, 2023.
- [74] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.
- [75] Y. Wu, Y. Wang, Z. Ye, T. Du, S. Jegelka, and Y. Wang. When more is less: Understanding chain-of-thought length in llms. *arXiv preprint arXiv:2502.07266*, 2025.
- [76] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.
- [77] T. Xie, Z. Gao, Q. Ren, H. Luo, Y. Hong, B. Dai, J. Zhou, K. Qiu, Z. Wu, and C. Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.
- [78] Y. Xu, X. Guo, Z. Zeng, and C. Miao. Softcot: Soft chain-of-thought for efficient reasoning with llms. *arXiv preprint arXiv:2502.12134*, 2025.
- [79] L. Yu, P. Poirson, S. Yang, A. C. Berg, and T. L. Berg. Modeling context in referring expressions. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 69–85. Springer, 2016.
- [80] Y. Yue, Z. Chen, R. Lu, A. Zhao, Z. Wang, S. Song, and G. Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.
- [81] Y. Zhang, A. Unell, X. Wang, D. Ghosh, Y. Su, L. Schmidt, and S. Yeung-Levy. Why are visually-grounded language models bad at image classification? *arXiv preprint arXiv:2405.18415*, 2024.
- [82] H. Zhou, X. Li, R. Wang, M. Cheng, T. Zhou, and C.-J. Hsieh. R1-zero’s" aha moment" in visual reasoning on a 2b non-sft model. *arXiv preprint arXiv:2503.05132*, 2025.
- [83] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Conditional prompt learning for vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16816–16825, 2022.
- [84] K. Zhou, J. Yang, C. C. Loy, and Z. Liu. Learning to prompt for vision-language models. *International Journal of Computer Vision*, 130(9):2337–2348, 2022.

Contents

1	Introduction	1
2	RFT Methods: Think vs. No Think	3
2.1	Thinking-RFT	3
2.2	No-Thinking-RFT	4
3	General Visual Reasoning	4
3.1	Think-After-Answer & Adaptive-Thinking	4
3.2	Experimental Setup	4
3.3	Results & Findings	5
3.4	Parameter Change during RL	7
4	Case Study on Image Classification Tasks	7
4.1	Experimental Setup	8
4.2	Few-Shot Learning Results	8
4.3	Free-Lunch Phenomenon	8
4.4	Efficiency Comparison	9
4.5	Convergence Comparison	10
5	Related Works	10
6	Conclusion	10
A	Group Relative Policy Optimization	18
B	Detailed Related Works	18
B.1	MLLM for Image Classification	18
B.2	RL for Post Training	19
B.3	Rule-Based Reinforcement Fine-tuning	19
B.4	Overthinking in LLMs and MLLMs	19
C	Dataset Statistics and Details	19
D	Implementation Details	20
D.1	Dataset Construction for Classification	20
D.2	Implementation Details for Classification	20
D.3	Implementation Details for More Diverse Tasks	21
E	Prompt	21
E.1	Few-shot Prompt	21
E.2	Thinking-After-Answer	22
E.3	Adaptive-Thinking	22

F	More Experimental Results on Classification	22
F.1	Base-to-New Results	22
F.2	Open-set Classification Comparison	23
F.3	Results of Classification with 7B Models.	24
F.4	More Discussion About Free-Lunch Phenomenon	24
G	More Experimental Results on More Diverse Tasks and Model Types	25
G.1	Experiments on More different Models	25
G.2	Referring Expression Comprehension Results	25
G.3	Experiments on Few-Shot Detection	26
G.4	Experiments of Thinking-RFT inference with no-thinking tags.	26
H	More Discussion	26
H.1	Limited Gains in Puzzle Tasks	26
H.2	Discussion between offline DPO, No-Thinking-RFT, and Thinking-RFT	27
I	Examples of the Model Response	28

Table 8: Statistics and questions for each Dataset.

Dataset	Classes	Train	Val	Test	Question
ImageNet	1,000	1.28M	N/A	50,000	“What type of object is in the photo?”
Caltech101	100	4,128	1,649	2,465	“What type of object is in the photo?”
OxfordPets	37	2,944	736	3,669	“What type of object is in the photo?”
StanfordCars	196	6,509	1,635	8,041	“What type of object is in the photo?”
Flowers	102	4,093	1,633	2,463	“What type of object is in the photo?”
Food101	101	50,500	20,200	30,300	“What type of object is in the photo?”
FGVCAircraft	100	3,334	3,333	3,333	“What type of aircraft is in the photo?”
SUN397	397	15,880	3,970	19,850	“What type of object is in the photo?”
DTD	47	2,820	1,128	1,692	“What type of texture is in the photo?”
EuroSAT	10	13,500	5,400	8,100	“What type of object is in the centered satellite photo?”
UCF101	101	7,639	1,898	3,783	“What actions is the person performing in the photo?”

A Group Relative Policy Optimization

We follow Deepseek-R1 [19, 59] to employ Group Relative Policy Optimization (GRPO) as the RL algorithm for optimization. Unlike SFT, which optimizes models through token-level losses, RL-based methods like GRPO utilize policy gradients, calculated from reward loss, for optimization. This encourages reasoning by exploring a much larger solution space [19].

Let Q be the question set, $\pi_{\theta_{\text{old}}}$ be the policy model and $\{o_1, o_2, \dots, o_G\}$ be a group of response from $\pi_{\theta_{\text{old}}}$ for a question q . Let $\pi_{\theta_{\text{ref}}}$ denote the frozen reference model. The GRPO algorithms aim to optimize model π_{θ} by the following objective:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim Q, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \min \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)} A_i, \text{clip} \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{\text{old}}}(o_i|q)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right],$$

where ϵ and β are clipping hyper-parameter and the coefficient controlling the Kullback–Leibler (KL) penalty, respectively. Here, $A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}$ is the advantage using the group reward $\{r_1, r_2, \dots, r_G\}$, and $D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_{\theta}(o_i|q)} - \log \left(\frac{\pi_{\text{ref}}(o_i|q)}{\pi_{\theta}(o_i|q)} \right) - 1$ is the KL divergence loss to prevent current model π_{θ} deviating too much from reference model $\pi_{\theta_{\text{ref}}}$. GRPO eliminates the critic model in PPO by estimating the relative advantage by sampling a group of responses $\{o_i\}_{i=1}^G$ and normalizing their rewards within the group to compute a relative advantage, which is more computationally efficient [59].

B Detailed Related Works

B.1 MLLM for Image Classification

Image classification is at the core of machine learning research, driving many fundamental advancements in theoretical understanding and practical applications. Early CNN-based [35, 32, 62, 66, 20, 37, 22, 67] models perform classification by hierarchically extracting spatial features through convolutional layers and mapping them to class probabilities. Vision transformers [15] largely follow the same pipeline but replace convolution with self-attention [70] for global feature modeling. CLIP [55] classifies images by computing the similarity between the image embedding and text embeddings of category descriptions and then selecting the closest match. More recently, the advent of MLLM enabled a new paradigm for image classification that leverages both vision encoders and LLMs. Unlike traditional classifiers, MLLMs take images and natural language prompts as input and generate text-based outputs, making them more interpretable, flexible, and user-friendly. However, MLLM for image classification is still an underexplored area. Recent work [81] shows that MLLMs perform poorly at classification but can be improved with SFT. In this work, we take a different angle and investigate a novel approach: using rule-based RL to fine-tune MLLM for classification. We systematically study its effectiveness compared to SFT and show its advantage and potential improvements.

B.2 RL for Post Training

Reinforcement learning (RL) has become an important technique used in LLM and MLLM post-training. Introduced by InstructGPT [50] as Reinforcement Learning from Human Feedback (RLHF), RL was used to align LLM and MLLM output with human’s preference. Due to the computation inefficiency of PPO [58] used in RLHF, some offline RL algorithms [47, 56] and value model free RL algorithms [59, 23] were introduced in the post training stage. Besides, researchers also explored how to build a good reward model in terms of robustness and reward density [73, 18, 39]. Recently, DeepSeek-R1 [19] applied the rule-based reward to the reinforcement training of LLM, proving the huge potential of RL in terms of incentivizing LLM’s reasoning ability. Inspired by DeepSeek-R1, we applied reinforcement learning to the classification task with a rule-based reward function, and analyzed its generalizability and learning efficiency compared to supervised fine-tuning.

B.3 Rule-Based Reinforcement Fine-tuning

Rule-based reinforcement fine-tuning (RFT) has recently achieved much process in large language models [19, 26, 68]. To transfer this success to MLLMs, numerous studies have been proposed [41, 60, 82, 24, 46, 3, 34, 53]. These works aim to replicate phenomena observed in Deepseek-R1, such as increased response length and the emergence of ‘aha’ moments. However, the role of the thinking process in RFT has not been explored in depth in these works. In this paper, we investigate its impact across different model sizes and tasks.

B.4 Overthinking in LLMs and MLLMs

Recent advancements in sophisticated reasoning abilities enabled by techniques such as Chain-of-Thought (CoT) prompting [74] have marked a significant milestone in the development of large language models (LLMs) and multimodal LLMs (MLLMs). CoT allows models to generate intermediate reasoning steps when solving complex problems, thereby improving both transparency and performance. However, this capability has also introduced a notable challenge referred to as the "overthinking phenomenon"[65].

Overthinking describes the tendency of LLMs and MLLMs to produce unnecessarily verbose, redundant, and computationally expensive reasoning chains, even for simple queries. This behavior can hinder practical deployment and, in some cases, degrade performance [40, 4, 10, 75, 28, 27, 29, 78, 11]. For example, [40] shows that CoT can harm accuracy on tasks where extra deliberation impairs human performance. [4] quantifies overthinking in powerful LLMs and proposes pruning strategies, [10] shows excessive internal reasoning degrades success of LLM-based agents and [27] demonstrates that reasoning offers limited benefits on commonsense tasks. Additionally, [28] shows that overthinking can negatively impact MLLM inference accuracy and [78] enhances CoT reasoning by representing its steps in a continuous space which makes CoT more efficient and improve performance.

While these studies provide valuable insights into model reasoning behavior, they focus exclusively on inference. The effect of explicit reasoning during reinforcement fine-tuning (RFT), however, remains largely unexplored.

C Dataset Statistics and Details

Following CoOp [84], we conducted extensive experiments on 11 public classification benchmark datasets to evaluate the effectiveness of the proposed CLIPFit. The datasets are ImageNet [14], Caltech101 [16], OxfordPets [51], StanfordCars [31], Flowers102 [48], Food101 [2], FGVC Aircraft [44], SUN397 [76], DTD [9], EuroSAT [21], and UCF101 [63]. The dataset statistics and the questions for each dataset are shown in Table 8.

For general reasoning dataset, CVBench is a benchmark for spatial reasoning and SAT is a training dataset for spatial reasoning. PuzzleVQA is a dataset for visual puzzle reasoning. MathVista is for general math reasoning and MathVision is for competition math reasoning.

The examples of these datasets can be found in Appendix I.

Table 9: Comparison of Qwen2VL instruct, SFT, Thinking-RFT, and No-Thinking-RFT in the base-to-new generalization setting. No-Thinking: No-Thinking-RFT. Base: base class accuracy. New: new class accuracy. H: harmonic mean accuracy. No-Thinking: No-Thinking-RFT.

(a) Average over 11 datasets.				(b) ImageNet.				(c) Caltech101.			
	Base	New	H		Base	New	H		Base	New	H
Qwen2VL	62.1	66.27	64.12	Qwen2VL	61.56	74.9	67.58	Qwen2VL	88.83	92.9	90.82
SFT	67.4	70.73	69.03	SFT	27.78	47.78	35.13	SFT	93.87	93.01	93.44
Thinking-RFT	81.17	79.15	80.15	Thinking-RFT	88.12	90.01	89.05	Thinking-RFT	97.74	95.2	96.45
No-Thinking	83.42	81.88	82.64	No-Thinking	88.97	90.66	89.81	No-Thinking	97.93	95.63	96.77
(d) DescribableTextures.				(e) EuroSAT.				(f) Food101.			
	Base	New	H		Base	New	H		Base	New	H
Qwen2VL	60.99	61.34	61.17	Qwen2VL	54.52	63.54	58.69	Qwen2VL	74.07	80.42	77.12
SFT	71.98	71.41	71.7	SFT	91.55	77.87	84.16	SFT	74.27	77.82	76.0
Thinking-RFT	77.42	67.82	72.3	Thinking-RFT	58.09	69.33	63.22	Thinking-RFT	87.29	87.56	87.42
No-Thinking	77.42	70.37	73.72	No-Thinking	66.43	74.13	70.07	No-Thinking	88.77	88.93	88.85
(g) OxfordFlowers.				(h) OxfordPets.				(i) StanfordCars.			
	Base	New	H		Base	New	H		Base	New	H
Qwen2VL	61.1	60.99	61.05	Qwen2VL	75.59	91.79	82.9	Qwen2VL	43.81	33.15	37.74
SFT	97.77	94.95	96.34	SFT	84.06	86.28	85.15	SFT	74.54	69.68	72.03
Thinking-RFT	87.78	74.97	80.87	Thinking-RFT	83.28	94.49	88.53	Thinking-RFT	82.08	75.74	78.78
No-Thinking	88.71	76.73	82.29	No-Thinking	86.64	95.5	90.85	No-Thinking	91.13	87.04	89.04
(j) SUN397.				(k) UCF101.				(l) FGVC Aircraft.			
	Base	New	H		Base	New	H		Base	New	H
Qwen2VL	56.97	65.8	61.07	Qwen2VL	69.6	64.62	67.02	Qwen2VL	36.07	39.47	37.7
SFT	27.39	37.8	31.77	SFT	59.95	63.93	61.87	SFT	38.23	57.53	45.94
Thinking-RFT	81.03	82.52	81.77	Thinking-RFT	79.47	74.95	77.14	Thinking-RFT	70.53	58.07	63.69
No-Thinking	83.18	84.14	83.66	No-Thinking	80.47	79.18	79.82	No-Thinking	68.01	58.31	62.79

D Implementation Details

D.1 Dataset Construction for Classification

Due to computational resource and model input limit, we select subset of class labels as the choice list. For few-shot classification, we sample 40% of labels including ground truth labels to form the classification list in question. For base-to-new classification, we sample 80% of labels including ground truth labels to form the classification list in question. If the length of the final choice list is less than 30, we include all class labels in the choice list, and if the length of the final choice list is larger than 100, we include 100 class labels in the choice list.

D.2 Implementation Details for Classification

We implement our code in Pytorch [52]. We utilize Qwen2-VL-2B-Instruct [72] as the base model, and fine-tune all parameters during training, following [82, 3]. All training is conducted in 8 A100 GPUs. The batch size is set to 1 per GPU and we use 2-step gradient accumulation during training. All images are resized to 328×328 resolution with no data augmentation. We first extract answers from answer tags (<answer> ...</answer>) and then verify if class names are in answers. If the answer tag does not exist in model responses we directly verify if class names are in model responses, following [81]. Both the maximum prompt length and maximum response length are set to 1024 for all datasets except StanfordCars dataset. The maximum prompt length and maximum response length are set to 1024 for StanfordCars dataset are set to 2048 and 1024, respectively. The number of rollout is set to 4 for both 2B and 7B models. β is set to 0.04, learning rate is set to 1×10^{-6} .

D.3 Implementation Details for More Diverse Tasks

Visual Perception. We follow [82] to fine-tune models on SAT dataset [57] 2 epochs and then test on CVBench dataset [69]. We also include classification results for comparison. The maximum length of prompt and response is both set to 1024. The number of rollout is set to 4 for both 2B and 7B models. The number of rollout is set to 4 for both 2B and 7B models. β is set to 0.04, learning rate is set to 1×10^{-6} . The batch size is set to 1 per GPU and we use 2-step gradient accumulation during training.

Multi-Modal Math Reasoning. We utilize the Math-40K [61] as the find-tuning data and fine-tune models 1 epoch. Then we test the fine-tuned models on both MathVista [43] and MathVision [71]. The maximum length of input prompt and response are set to 4096 and 512 respectively. The number of rollout is set to 8 for 2B model, and 4 for 7B model. β is set to 0.04, learning rate is set to 1×10^{-6} . The batch size is set to 1 per GPU and we use 2-step gradient accumulation during training.

Visual Puzzle Reasoning. We follow the code of [7] to generate a training dataset with 6.5k data and fine-tune models 2 epochs. We then test fine-tuned models on PuzzleVQA [7] as in-domain testing, and on AlgoPuzzleVQA [17] as out-of-domain (OOD) testing. β is set to 0.04, learning rate is set to 1×10^{-6} . The batch size is set to 1 per GPU and we use 2-step gradient accumulation during training. The maximum length of prompt and response is both set to 1024. The number of rollout is set to 8 for 2B model, and 4 for 7B model.

E Prompt

E.1 Few-shot Prompt

We provide the prompt for thinking answer extract here. We use classification as the example, and for other datasets, the only difference is the few-shot examples.

Prompt: *I will give you a question, answer and the model response. The model response is in the form <thinking>...</thinking><answer>...</answer>. The thinking process is in <thinking>...</thinking> and the answer is in <answer>...</answer>.*

I want to to extract the answer in thinking tags and see if it is the same as the answer in answer tag. For multichoice problem, extract the choice (e.g., A,B,C) that corresponding to the answer meaning of thinking process in thinking tag. If there is no answer in thinking tag, output None. Please output three items answer in thinking tag: answer in answer tag:

consistency with answer tag:

Below is some examples:

Question: What type of texture is in the photo?

Please choose one from list [perforated, marbled, zigzagged, waffled, meshed, flecked, sprinkled, matted, woven, grooved, gauzy, bumpy, polka-dotted, potholed, veined, frilly, grid, bubbly, cobwebbed, scaly, pleated, crosshatched, wrinkled, freckled, porous, stained, swirly, chequered, spiralled, fibrous].

answer: veined

model response: <think> The texture in the photo appears to be a series of parallel lines or grooves running along the surface of the leaf. This suggests a grooved texture. </think> <answer> grooved </answer>

answer in thinking tag: grooved

answer in answer tag: grooved

consistency with answer tag: Yes

Question: What type of texture is in the photo?

Please choose one from list [fibrous, striped, chequered, swirly, studded, lacelike, knitted, freckled, crosshatched, wrinkled, cobwebbed, grooved, waffled, grid, scaly, bubbly, sprinkled, paisley, spiralled, interlaced, porous, veined, blotchy, pitted, zigzagged, crystalline, flecked, potholed, cracked, honeycombed].

answer: potholed

model response:<think> The photo shows a road with visible damage, including cracks and potholes. The texture appears to be characterized by these irregularities and depressions in the surface. </think> <answer> potholed </answer>

answer in thinking tag: None

answer in answer tag: potholed

Table 10: Comparison of Adaptive-Thinking model response type among 2B and 7B models on MathVista, MathVision, CVBench, PuzzleVQA (Puzzle), and AlgoPuzzleVQA (AlgoPuzzle).

Model	Response Form	MathVista	MathVision	CVBench	Puzzle	AlgoPuzzle
2B	w/ Thinking	0%	0%	0%	0%	0%
	w/o Thinking	100%	100%	100%	100%	100%
7B	w/ Thinking	100%	100%	99.6%	0%	0%
	w/o Thinking	0%	0%	0.4%	100%	100%

consistency with answer tag: None

Question: What type of texture is in the photo?

Please choose one from list [dotted, knitted, woven, honeycombed, bubbly, lacelike, swirly, interlaced, spiralled, marbled, zigzagged, pleated, stratified, gauzy, chequered, fibrous, paisley, lined, striped, studded, grid, crosshatched, crystalline, blotchy, stained, wrinkled, frilly, scaly, waffled, perforated].

answer: spiral

model response: <think> The photo shows a pattern with concentric circles that twist and turn in a spiral-like manner. This suggests that the texture is related to the spiral pattern.</think> <answer>swirly </answer>

answer in thinking tag: spiral

answer in answer tag: swirly

consistency with answer tag: No

(two more examples)

E.2 Thinking-After-Answer

The prompt of Think-After-Answer is designed as: {Question} Please first output the answer in <answer> </answer> tags and then output a brief reasoning process in <reason> </reason> tags.

E.3 Adaptive-Thinking

We provide more discussions about Adaptive-Thinking method in this subsection.

Instruction Prompt. {Question}. Please first identify whether this problem requires intermediate thinking or calculation. If the problem requires thinking or calculation, output the thinking and calculation process inside <think> </think> tags and the final answer inside <answer> </answer> tags. If no thinking or calculation is required, directly output the final answer inside <answer> </answer> tags. Your output should follow one of two cases: (1) '<answer> ... </answer>', (2) '<think> ... </think> <answer> ... </answer>'.

Format Reward. Both the thinking format (<think> ... </think> <answer> ... </answer>) and the direct-answer format (<answer> ... </answer>) will be assigned a reward of 1.

Accuracy Reward. The accuracy reward is not changed as Thinking-RFT.

F More Experimental Results on Classification

F.1 Base-to-New Results

In this subsection, we introduce the results on the base-to-new generalization setting. Following [84], We split each dataset into two disjoint groups: the base class dataset and the new class dataset. This experimental setup is designed to assess the model’s ability to acquire downstream knowledge while simultaneously demonstrating generalization to novel information. All the methods are fine-tuned on the base class dataset and evaluated on both the base and new class test dataset. We conduct 4-shot experiments. The results are shown in Table 9.

As demonstrated in Table 9, our proposed Thinking-RFT framework notably surpasses SFT in performance. In detail, Thinking-RFT exhibits an approximately 14% higher accuracy for base classes and a 9% increase for new class accuracy, leading to an aggregate improvement of 11% in the harmonic mean accuracy. These results demonstrate the effectiveness of rule-base reinforcement

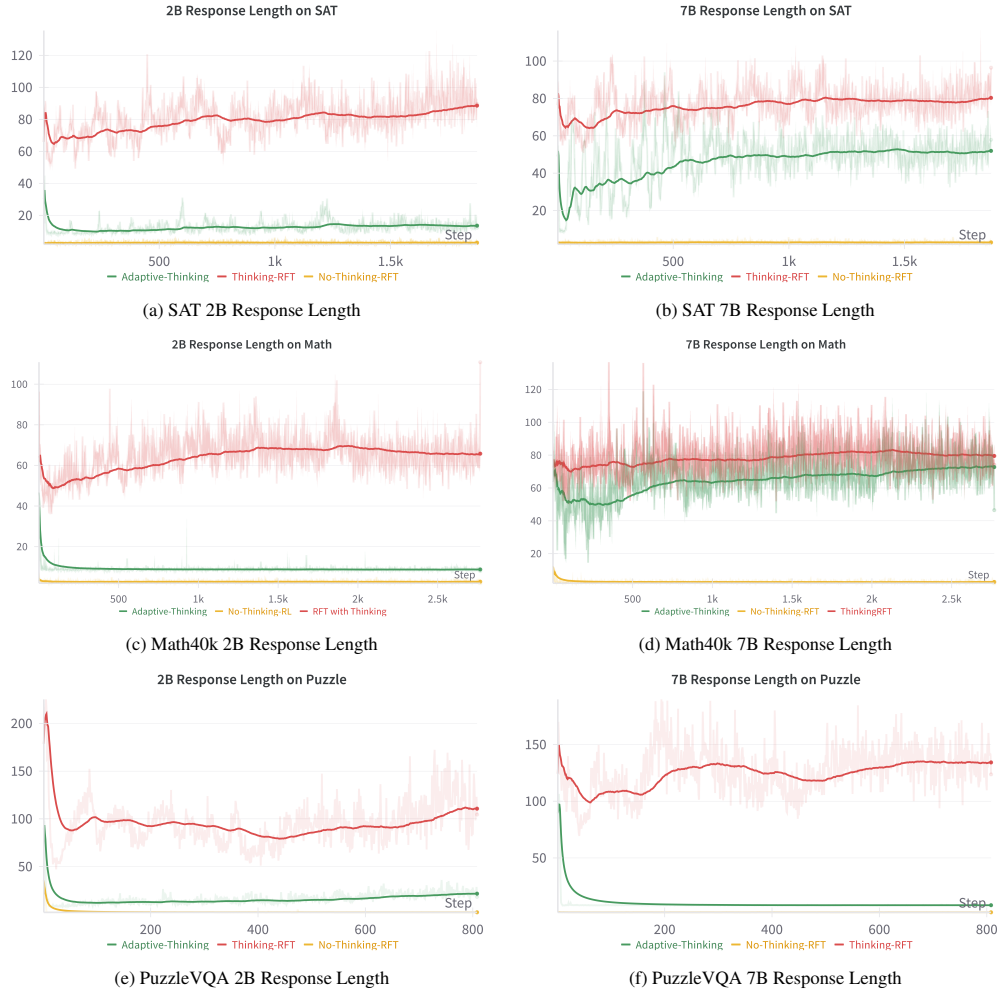


Figure 6: Comparison of response length trend on SAT, Math40k, and PuzzleVQA datasets over steps of Thinking-RFT, No-Thinking-RFT, and Adaptive-Thinking across 2B and 7B models.

fine-tuning in image classification. However, we also find that SFT can surpass Thinking-RFT in certain datasets, such as OxfordFlowers and EuroSAT, indicating that SFT may hold advantages in specific scenarios. We also observed that SFT significantly underperforms on the ImageNet and SUN397 datasets. The likely reason is that the prompts in these two datasets are very lengthy, preventing SFT from effectively memorizing classification knowledge, thereby resulting in poor performance.

The proposed No-Thinking-RFT demonstrates superior performance, in both base class and new class average accuracy, resulting in a 2.5% enhancement in average harmonic mean accuracy, compared with Thinking-RFT. These findings suggest that omitting the thinking process during fine-tuning allows rule-based RL to achieve improved classification performance and enhanced generalization capabilities than with thinking process.

F.2 Open-set Classification Comparison

We present the results of open-set classification using Thinking-RFT and No-Thinking-RFT in this subsection. Unlike closed-form classification, open-set classification is not a well-defined problem, which is much harder or even unrealistic for some datasets, since synonyms, plural forms, and name partially missing will be judged as incorrect. For example, in the StanfordCars dataset [31], the model can hardly output the correct year of the car in images. Therefore, we selected five datasets with class names that are relatively straightforward for the model to output and compare the few-shot

Table 11: Comparison on open-set Few-shot learning results.

	ImageNet	Caltech101	Food101	Flowers102	OxfordPets	Average
Qwen2VL	46.57	62.96	57.79	48.44	47.40	52.63
Thinking-RFT	54.84	79.07	73.51	67.64	89.94	73.0
No-Thinking-RFT	56.45	86.29	71.99	71.21	86.07	74.40

Table 12: Comparison of Thinking-RFT and No-Thinking-RFT with 2B and 7B models on fewshot classification. DTD: DescribableTextures.

Model	Method	DTD	EuroSAT	OxfordFlowers	StanfordCars	Average
2B	Thinking-RFT	69.92	49.46	86.56	80.24	71.55
	No-Thinking-RFT	73.52	58.02	91.6	92.5	78.91
7B	Thinking-RFT	77.90	53.17	93.91	84.19	77.29
	Think-After-Answer	76.29	62.95	94.84	89.32	80.85
	Adaptive-Thinking	79.60	56.62	96.86	89.92	80.75
	No-Thinking-RFT	80.56	58.91	94.24	94.02	81.93

learning performance between Thinking-RFT and No-Thinking-RFT. The results are shown in Table 11. No-Thinking-RFT outperforms Thinking-RFT on three datasets among five datasets, ultimately achieving a 1.4% improvement in average accuracy over Thinking-RFT.

F.3 Results of Classification with 7B Models.

We further conduct experiments of classification on DescribableTextures, EuroSAT, OxfordFlowers, StanfordCars datasets using Qwen-VL-2-7B models on and report the results on Table 12. As shown in Table 12, compared with 2B models, the performance gap between Thinking-RFT and No-Thinking-RFT narrows but remains. These results suggest that image classification does not require explicit thinking, and that excluding the explicit thinking process can enhance both performance and computational efficiency.

F.4 More Discussion About Free-Lunch Phenomenon

In this subsection, we give more discussion about the free-lunch phenomenon. Despite improvement in most cases for cross dataset improvement in Figure 4, it is also noted that such improvements can be negative in certain instances. For example, fine-tuning on the EuroSAT dataset could result in diminished performance on the OxfordPets dataset. This outcome is understandable, given that the knowledge required for the EuroSAT dataset is significantly divergent from that of the OxfordPets dataset, and the classification knowledge from the EuroSAT dataset may even be detrimental to the classification of the OxfordPets dataset.

We further illustrate the comparative improvement of No-Thinking-RFT over Thinking-RFT by testing on 11 datasets, using a model that was fine-tuned on one specific dataset. We also showcase the comparative improvement achieved by using a standard training prompt over a No-Thinking Prompt (which directs Thinking-RFT to immediately produce the answer during inference). The results are shown in Figure 7. We can find that No-Thinking-RFT has a better cross-dataset generalization ability than Thinking-RFT, except OxfordPets and FGVC Aircraft datasets. This suggests that utilizing an equality reward without engaging in any thinking process can enhance the model’s cross-dataset generalization ability, with the exceptions being the OxfordPets and FGVC Aircraft datasets. For these datasets, the thinking process during fine-tuning appears to be important for cross-dataset performance. These findings are consistent to the results of using different inference prompt strategies of Thinking-RFT, as shown in Figure 7 (b). We find that using normal prompt for in-domain fewshot test performs better than No-Thinking prompt among all datasets. This is reasonable given that the normal prompt is the one used for fine-tuning. However, it is surprising to discover that using the normal prompt for cross-dataset testing results in lower performance in many cases compared to the No-Thinking prompt, except OxfordPets and FGVC Aircraft datasets. These findings suggest

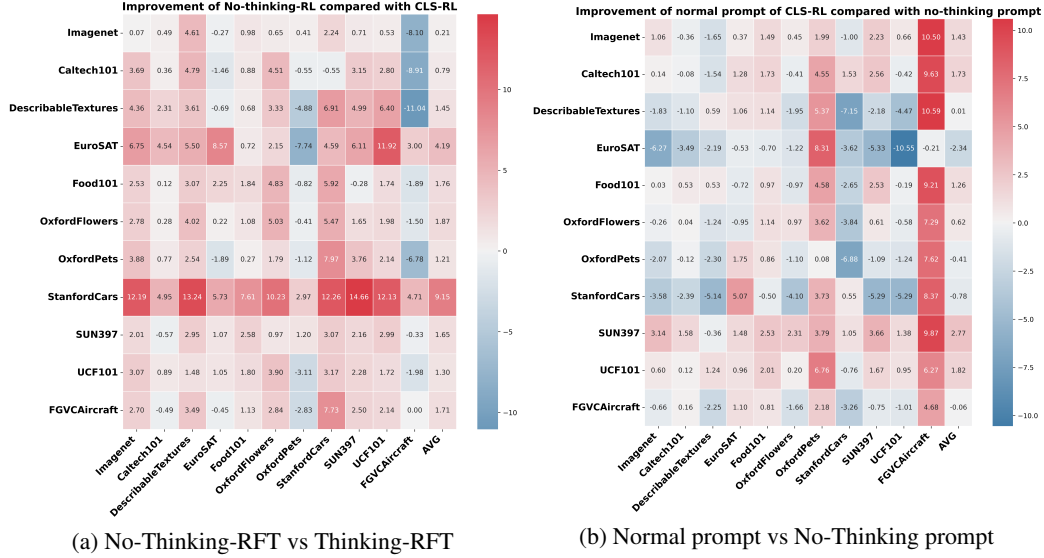


Figure 7: Visualization of improvement of No-Thinking-RFT on different datasets compared with Thinking-RFT (Left) and improvement of Thinking-RFT compared with Thinking-RFT with No-Thinking prompt (Right). The accuracy improvement is marked as red, and decrease is marked as blue

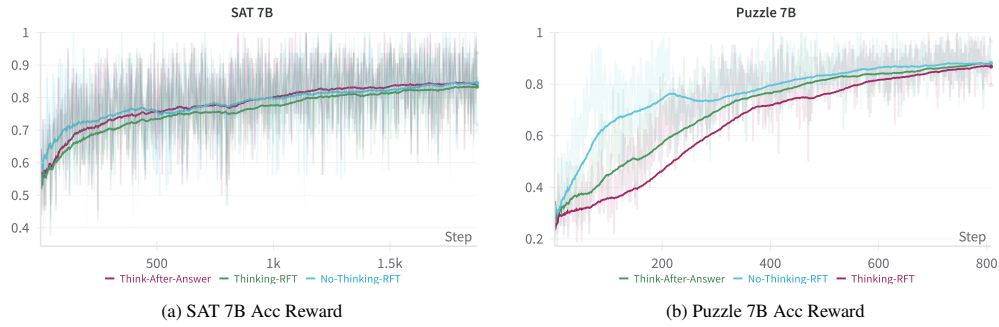


Figure 8: Comparison of accuracy reward convergence speed on Puzzle dataset over steps of Thinking-RFT, Think-After-Answer, and No-Thinking-RFT across 2B and 7B models.

that while Thinking-RFT fine-tuning may enable the model to learn good cross-dataset classification generalization ability, over-thinking during inference can potentially diminish this ability.

G More Experimental Results on More Diverse Tasks and Model Types

G.1 Experiments on More different Models

We report more experimental results on more diverse model types. We conduct experiments on InternVL2.5-1B and InternVL2.5-4B [5], and Qwen2-5-VL-Instruct-3B and Qwen2-5-VL-Instruct-7B [1]. The experimental results are shown in Table 14 and Table 13. The results are almost similar to which in the main text, i.e., visual and puzzle tasks do not need thinking during RFT.

G.2 Referring Expression Comprehension Results

We follow VLM-R1 [60] to conduct experiments of the referring expression comprehension task. The experiment setting and implementation is the same as VLM-R1 [60]. We follow [60] to use the training splits of Refcoco+/g [79, 45] as the training data, and the val split of Refcoco+/g for in-domain evaluation and test split of LISA-Grounding [33] for out-of-domain (OOD) evaluation.

Table 13: Results of Thinking-RFT and No-Thinking-RFT on CVBench across different models.

Model	Setting	Overall	Count	Relation	Depth	Distance
InternVL-2.5-1B	Thinking-RFT	68.57	62.82	80.46	75.33	56.50
	No-Thinking-RFT	70.55	65.48	77.08	78.67	62.00
InternVL-2.5-4B	Thinking-RFT	79.34	71.32	91.08	87.17	69.33
	No-Thinking-RFT	79.76	70.94	89.85	86.50	73.67
Qwen-VL-2.5-3B	Thinking-RFT	76.76	70.43	86.46	79.67	71.67
	No-Thinking-RFT	79.83	70.18	89.54	83.67	78.17

Table 14: Performance comparison of PuzzleVQA and AlgoPuzzleVQA on Qwen2-5-VL-Instruct-7B models under three settings.

Model	Method	PuzzleVQA	AlgoPuzzleVQ
Qwen-VL-2.5-7B	Thinking-RFT	77.10	27.72
	No-Thinking-RFT	86.45	26.11

We follow VLM-R1 [60] to report the evaluation results of first 600 steps. The results are shown in Table 15. We observe that No-Thinking-RFT generally achieves better performance than VLM-R1 on the RefCOCO/+g in-domain tests but performs worse on the LISA-Grounding out-of-domain (OOD) test. These results suggest that for referring grounding tasks, reinforcement fine-tuning (RFT) without explicit thinking can improve in-domain performance while potentially degrading OOD generalization. However, we also note that as fine-tuning progresses, the OOD performance of No-Thinking-RFT improves and consistently surpasses that of SFT.

G.3 Experiments on Few-Shot Detection

We follow Viusal-RFT [41] to conduct experiments on few-shot object detection with MLLMs. We follow [41] to select eight categories from the COCO dataset with 4 images per category, to construct training dataset. We then follow [41] to train the Qwen2-VL-2B-instruct-2B and Qwen2-VL-2B-instruct-7B models for 100 steps (50 epochs). The results are shown in Table 16. No-Thinking-RFT consistently outperforms Thinking-RFT among both 2B and 7B models.

G.4 Experiments of Thinking-RFT inference with no-thinking tags.

We further conduct experiments to investigate where the performance gain of No-Thinking-RFT over Thinking-RFT on perception and puzzle tasks originates. We explore the No-Thinking inference mode of Thinking-RFT by appending an empty thinking tag (*<thinking>* *</thinking>*) during inference. The results are shown in Table 17. We find that while appending empty thinking tags during inference could improve performance on CVBench and puzzle tasks, its performance is still far behind No-Thinking-RFT. Moreover, appending empty thinking tags during inference will decrease performance on math tasks. These results suggest that the performance gain of No-Thinking-RFT over Thinking-RFT on perception and puzzle tasks primarily stems from two factors: improved learning during fine-tuning and the avoidance of overthinking by bypassing inference.

H More Discussion

H.1 Limited Gains in Puzzle Tasks

In this subsection, we discuss the explanation accounts for limited puzzle gains. As a pragmatic proxy, we evaluated the o4-mini model, whose API supports optional in-context visual reasoning (e.g., cropping). On PuzzleVQA, performance rises from 84.30% (no visual reasoning) to 86.48% (with visual reasoning), a gain of +2.18% accuracy. This suggests that visual-context learning can improve puzzle performance.

Table 15: Performance comparison of Refcoco+/g and LISA ground datasets.

Training method	Evaluation Dataset	100	200	300	400	500	600
SFT	Refcoco _{val}	88.7	88.85	88.7	88.25	88.85	88.7
VLM-R1	Refcoco _{val}	88.7	88.7	89.4	89.25	90	90.55
No-Thinking-RFT	Refcoco _{val}	90.29	90.85	90.52	90.88	90.85	90.83
SFT	Refcoco+ _{val}	82.55	82.15	81.85	81.9	82.3	82.25
VLM-R1	Refcoco+ _{val}	82.6	81.9	82.8	83.35	83.6	84.3
No-Thinking-RFT	Refcoco+ _{val}	83.71	84.39	84.76	85.02	85.28	85.24
SFT	Refcocog _{val}	85.65	85.95	85.85	85.6	85.95	85.95
VLM-R1	Refcocog _{val}	85.95	85.05	85.45	85.65	87.15	87.1
No-Thinking-RFT	Refcocog _{val}	85.70	86.15	86.70	86.60	86.97	86.91
SFT	LISA-Grounding	55.91	56.51	55.66	55.18	55.66	54.82
VLM-R1	LISA-Grounding	61.82	61.27	61.64	62.6	61.88	63.14
No-Thinking-RFT	LISA-Grounding	57.90	57.24	58.99	59.65	59.95	61.76

Table 16: Performance comparison of 4-shot results on COCO dataset of 8 categories.

Models	mAP	bus	train	fire hydrant	stop sign	cat	dog	bed	toilet
<i>Qwen2-VL-2B</i>									
Zero-Shot	19.6	19.0	15.8	25.8	18.4	29.9	23.2	14.6	9.8
Thinking-RFT	40.6	30.0	40.6	45.7	35.0	60.9	44.9	24.6	43.1
No-Thinking-RFT	43.72	32.29	44.64	47.38	43.43	61.59	52.94	24.81	42.71
<i>Qwen2-VL-7B</i>									
Zero-shot	43.0	35.0	43.3	37.1	36.7	57.3	50.3	37.4	47.1
Thinking-RFT	54.3	44.3	59.8	52.0	46.0	72.7	62.8	41.9	55.0
No-Thinking-RFT	54.99	49.10	58.72	53.37	45.52	72.35	63.64	46.41	50.79

H.2 Discussion between offline DPO, No-Thinking-RFT, and Thinking-RFT

In this subsection, we discuss the difference between offline DPO, No-Thinking-RFT, and Thinking-RFT.

- (i) Although No-Thinking-RFT removes the "think" phase, a list-wise DPO algorithm **should not** be expected to reproduce similar effect due to the fundamental difference between DPO and GRPO. We discuss the detailed difference below:
 - (a) **Sampling strategy.** List-wise DPO selects one positive and several negatives *offline*. In contrast, No-Thinking-RFT samples responses *online* at every training step: the mix of positives and negatives is not fixed at $1 : (N - 1)$ but varies with task difficulty and model competence (from $0 : N$ to $N : 0$). Thus while removing thinking, No-Thinking-RFT continues to explore and learn from fresh errors, whereas DPO does not.
 - (b) **Gradient update rule.** No-Thinking-RFT computes a policy-gradient using verifiable rewards ($R = 1$ for an exact match, $R = 0$ otherwise) and a group-advantage formulation, i.e. GRPO. Listwise DPO, by contrast, minimises a cross-entropy loss over a pre-computed list, functioning more like contrastive SFT. This distinction yields different learning dynamics.

Because of these differences, listwise DPO **should not** be expected to match the empirical behaviour of No-Thinking-RFT.

- (ii) We also implemented preliminary DPO and listwise DPO baselines to quantify the performance difference. We evaluated two negative-sampling strategies: one using samples generated by Qwen2-VL (DPO model) and the other using randomly generated samples (DPO random). Experiments were carried out on the FGVAircraft classification dataset and the MathVQA dataset. The results shown in Table 18 indicate that both DPO variants under both sampling strategies perform substantially worse than Thinking-RFT and

Table 17: Comparison of Adaptive-Thinking model response type among 2B and 7B models on MathVista, MathVision, CVBench, PuzzleVQA (Puzzle), and AlgoPuzzleVQA (AlgoPuzzle).

Model	Response Type	MathVista	MathVision	CVBench	Puzzle	AlgoPuzzle
2B	w/ Thinking	44.90	16.45	70.36	52.50	24.78
	w/o Thinking	41.9	15.79	71.72	62.90	27.50
	No-Thinking-RFT	48.80	13.16	76.76	70.85	29.17
7B	w/ Thinking	64.60	21.71	80.36	66.60	24.78
	w/o Thinking	61.20	22.04	81.62	76.85	27.95
	No-Thinking-RFT	59.10	18.09	80.67	80.65	29.39

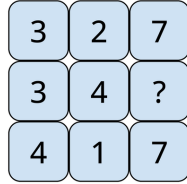
Table 18: Performance comparison of DPO, Listwise DPO, Thinking-RFT, and No-Thinking-RFT.

Model	FGVC	MathVista	MathVision
DPO random	46.23	40.15	12.3
DPO model	51.25	40.32	14.4
Listwise DPO random	47.19	40.78	12.45
Listwise DPO model	51.36	42.66	15.1
Thinking-RFT	74.41	44.90	16.45
No-Thinking-RFT	74.41	48.80	13.49

No-Thinking-RFT. This finding highlights the importance of online sampling and policy-gradient optimization in RFT. Moreover, model-generated negative samples consistently outperform random negatives in both DPO and listwise DPO settings, demonstrating that the quality of negative samples is critical to DPO training.

I Examples of the Model Response

In this subsection, we delve into the content of the thought process in Thinking-RFT among different model sizes and tasks. Specifically, we show We compare the response examples of four different thinking strategies in Figure 21~Figure 25, the examples of trivial reasoning responses of 2B models in Figure 15 ~ Figure 20, the inconsistent responses of 7B models in Figure 9~Figure 14. Typically, for 2B model responses, the content found in the "thinking" tags are somewhat trivial, such as "This is a photo of <class>" (classification) or "To find the area of the overlap between the two squares, we need to find the length of the diagonal of the smaller square." (MathVision) which offers little to no benefit towards arriving at the final answer, or they might already represent the final answers themselves without any reasoning process. For 7B models, the content within the thinking tag is significantly more meaningful and contributes more effectively to the reasoning process leading to the final answers. However, we frequently observe inconsistencies in the responses, where the content within the thinking tag diverges significantly from that in the answer tag. For example, the reasoning may support choice B, while the final answer provided is choice A. To quantify this issue, we use GPT-4o to extract the answers from thinking tag and answer tag and then calculate the proportion of such inconsistencies across each dataset. The detailed results are presented in Figure 2. Our analysis reveals that reasoning tasks are particularly prone to inconsistencies between the thinking and answer tags. Notably, as illustrated in Figure 2, the average accuracy of the answer tags in inconsistent responses is substantially higher than that of their corresponding thinking tags. This observation is expected, given that the answer tags are directly optimized via an accuracy-based reward signal, whereas the thinking tags receive no explicit supervision. Additionally, we find that the average accuracy of answer tags in inconsistent responses is lower than the overall average accuracy across all responses. This indicates that maintaining internal consistency between reasoning and final answers is conducive to improved model performance.



Question: What is the missing number of the part denoted with a question mark? Please choose your answer from: ['1', '2', '7', '5']

Ground Truth: 5.

Response from Thinking-RFT:

<think> The pattern in the rows and columns is as follows:

- First row: 3, 2, 7
- Second row: 3, 4, ?
- Third row: 4, 1, 7

The pattern in the rows and columns is that each number in the second row is the sum of the corresponding numbers in the first and third rows.

So, for the second row:

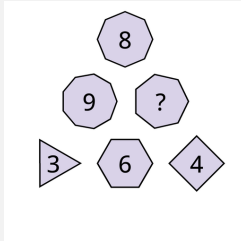
- $3 + 4 = 7$

- $4 + 1 = 5$

Therefore, the missing number in the second row is 5.

So, the answer is 5. </think>

<answer>1</answer>



Question: What is the missing number of the part denoted with a question mark? Please choose your answer from: [7, 4, 5, 3]

Ground Truth: 7.

Response from Thinking-RFT:

<think> The pattern in the shapes is that the number of sides of each shape increases by one from the bottom to the top. The shapes are a triangle (3 sides), a hexagon (6 sides), and an octagon (8 sides). Following this pattern, the next shape should have 9 sides. The number in the center of the octagon is 8, so the missing number in the center of the pentagon should be 7. Therefore, the missing number is 7. </think>

<answer>5</answer>

Response from No-Thinking-RFT: 7.

Figure 9: Examples of the inconsistency error of fine-tuned Qwen-VL-2-7B model on PuzzleVQA. The correct answer part is marked as green, and wrong answer part is marked as red.



Question: How many walls are in the image? A. 1 B. 3 C. 2 D. 0

Ground Truth: A.

Response from Thinking-RFT:

<think> The image shows a window with curtains on both sides. The curtains are hanging from a rod that is attached to the wall. There is no visible wall in the image, only the window and the curtains. Therefore, **the number of walls in the image is zero.**</think>

<answer> (A) 1 </answer>



Question: How many rugs are in the image? A.3 B. 1 C. 0 D. 2

Ground Truth: B.

Response from Thinking-RFT:

<think> The image shows a living room with various items and decorations. There is a rug on the floor in front of the coffee table, and another rug on the floor in front of the fireplace. Therefore, **there are two rugs in the image.**</think>

<answer> (B) 1 </answer>

Figure 10: Examples of the inconsistency error of fine-tuned Qwen-VL-2-7B model on CVBench. The correct answer part is marked as **green**, and wrong answer part is marked as **red**.



Question: Hint: Please answer the question requiring an integer answer and provide the final value, e.g., 1, 2, 3, at the end.

Question: what is the total volume of the measuring cup? (Unit: g)

Ground Truth: 1000.

Response from Thinking-RFT:

<think> The measuring cup has a scale marked in milliliters (mL). To convert milliliters to grams, we need to know the density of the substance being measured. For flour, the density is approximately 0.5 g/mL. The total volume of the measuring cup is 1000 mL. Therefore, **the total volume in grams is 1000 mL * 0.5 g/mL = 500 g.** </think>

<answer> 1000 </answer>



Question: Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, at the end.

Question: What time is shown? Answer by typing a time word, not a number. It is () past six.

Choices:

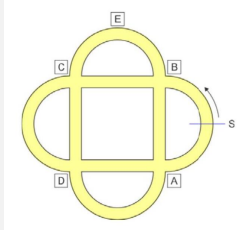
- (A) half
- (B) quarter
- (C) o'clock
- (D) quarter to
- (E) quarter past

Ground Truth: B.

Response from Thinking-RFT:

<think> The clock shows a time where the minute hand is pointing at the 3, which represents 15 minutes past the hour. The hour hand is between the 6 and the 7, which means it is 6 o'clock. Therefore, **the time is 6:15.** </think> <answer> A </answer>

Figure 11: Examples of the inconsistency error of fine-tuned Qwen-VL-2-7B model on MathVista. The correct answer part is marked as green, and wrong answer part is marked as red.



Question: Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, E, F at the end.

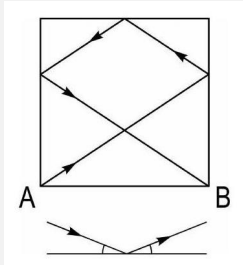
Peter rides his bike along a cycle path in a park. He starts at point S and rides in the direction of the arrow. At the first crossing he turns right, then at the next left, and then again to the right and then again to left. Which crossing does he not reach?

Choices: A. A B. B C. C D. D E. E

Ground Truth: D.

Response from Thinking-RFT:

<think> The arrow indicates the direction Peter rides his bike. He starts at point S and follows the path indicated by the arrow. He turns right at the first crossing, then left at the next, right again, and finally left. This means he will pass through the crossings labeled B, C, and D. The crossing labeled A is not mentioned in his path. Therefore, **the crossing he does not reach is A.**</think> <answer> C </answer>



Question: Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, E, F at the end.

A (very small) ball is kicked off from point A on a square billiard table with side length 2 m. After moving along the shown path and touching the sides three times as indicated, the path ends in point B . How long is the path that the ball travels from A to B ? (As indicated on the right: incident angle = emergent angle.)

Choices: A. 7 B. $2\sqrt{13}$ C. 8 D. $4\sqrt{3}$ E. $2 \cdot (\sqrt{2} + \sqrt{3})$

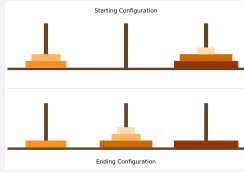
Ground Truth: B.

Response from Thinking-RFT:

<think> The path of the ball is a series of right angles, so we can break it down into segments of equal length. Each segment is the hypotenuse of a right triangle with legs of length 1 meter (since the side length of the square is 2 meters). Using the Pythagorean theorem, the length of each segment is $\sqrt{1^2 + 1^2} = \sqrt{2}$ meters. Since there are four segments, the total length of the path is $4\sqrt{2}$ meters. However, we can simplify this to $2\sqrt{2} \cdot 2 = 4\sqrt{2}$ meters. Therefore, **the correct answer is D.**</think>

<answer> B </answer>

Figure 12: Examples of the inconsistency error of fine-tuned Qwen-VL-2-7B model on MathVision. The correct answer part is marked as green, and wrong answer part is marked as red.



Question: You are playing a Tower of Hanoi game with 3 rods and 5 disks of various diameters, which can slide onto any rod. You are given the starting and ending configuration of the game as shown in the top and the bottom of the image, respectively. The game has the following rules: i) Only one disk may be moved at a time; ii) Each move consists of taking the upper disk from one of the stacks and placing it on top of another stack or on an empty rod; and iii) No disk can be placed on top of a disk that is smaller than it. What is the minimum number of moves required to go from the starting to the ending configuration? Please choose your answer from: 6, 2, 4, 5.

Ground Truth: 5.

Response from Thinking-RFT:

<think> The Tower of Hanoi problem is a classic puzzle that requires moving a stack of disks from one rod to another, following specific rules. The minimum number of moves required to solve the puzzle can be calculated using the formula $2^n - 1$, where n is the number of disks.

In this case, we have 5 disks, so the minimum number of moves required is $2^5 - 1 = 32 - 1 = 31$.

However, the options provided are 6, 2, 4, and 5. It seems there might be a misunderstanding in the question or the options. The correct answer based on the formula is 31, but it is not listed among the options.

Let's re-examine the image to see if there's any additional information or a different interpretation of the problem that might lead to a different answer.

Upon closer inspection, the image shows a specific starting and ending configuration, but it does not provide enough information to determine the exact number of moves required to solve the puzzle. The image does not show the intermediate steps or the specific rules applied to reach the ending configuration.

Given the options provided, it seems there might be a misunderstanding or an error in the question. The correct answer based on the formula is 31, but it is not listed among the options.

Therefore, the minimum number of moves required to go from the starting to the ending configuration is 31, but this is not listed among the options.

Final answer: 31 (not listed among the options)</think>

<answer>5</answer>

March						
Mon	Tue	Wed	Thu	Fr1	Sat	Sun
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Question: The image shows the calendar of a month of a particular non-leap year. Which day of the week was on February 25 of that year?

Please choose your answer from: Friday, Sunday, Monday, Tuesday.

Ground Truth: Friday.

Response from Thinking-RFT:

<think> The calendar shows that February 25 is a Friday. Therefore, **the day of the week for February 25 of that year is Friday.** </think>

<answer> **Tuesday** </answer>

Figure 13: Examples of the inconsistency error of fine-tuned Qwen-VL-2-7B model on AlgoPuzzleVQA. The correct answer part is marked as green, and wrong answer part is marked as red.



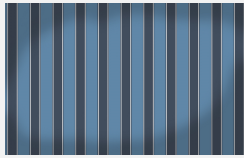
Question: What type of texture is in the photo?

Please choose one from list [matted, grid, crosshatched, frilly, flecked, zigzagged, freckled, waffled, chequered, porous, striped, fibrous, sprinkled, perforated, crystalline, cracked, banded, lacelike, veined, polka-dotted, paisley, scaly, pleated, smeared, dotted, cobwebbed, wrinkled, gauzy, interlaced, knitted].

Ground Truth: wrinkled.

Response from Thinking-RFT:

<think> The photo shows a close-up of a fabric with a smooth, continuous surface. There are no visible patterns, lines, or textures that would suggest a grid, crosshatched, frilly, flecked, zigzagged, freckled, waffled, chequered, porous, striped, fibrous, sprinkled, perforated, crystalline, cracked, banded, lacelike, veined, polka-dotted, paisley, scaly, pleated, smeared, dotted, cobwebbed, wrinkled, gauzy, interlaced, or knitted texture. The fabric appears to be soft and smooth, with a consistent, even texture. Therefore, **the most appropriate texture to describe this fabric is "pleated"**.</think> <answer> wrinkled </answer>



Question: What type of texture is in the photo?

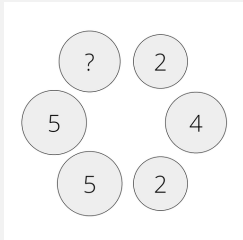
Please choose one from list [smeared, bubbly, fibrous, lacelike, frilly, potholed, spiralled, chequered, flecked, paisley, stained, meshed, wrinkled, zigzagged, polka-dotted, matted, freckled, lined, stratified, banded, cracked, sprinkled, gauzy, striped, marbled, crosshatched, swirly, pitted, grid, veined].

Ground Truth: striped.

Response from Thinking-RFT:

<think> <think> The photo shows a pattern of vertical lines. The lines are straight and evenly spaced, creating a consistent and repetitive visual effect. There are no visible curves, ripples, or other irregularities in the lines. Therefore, **the texture in the photo is best described as "lined."** </think> <answer> striped </answer>

Figure 14: Examples of the inconsistency error of fine-tuned Qwen-VL-2-7B model on classification DTD dataset. The correct answer part is marked as green, and wrong answer part is marked as red.



Question: How many cubes have been taken from the block?

Ground Truth: 4.

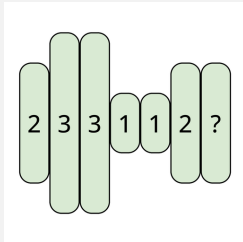
Response from Thinking-RFT:

<think>

To solve this, we need to determine the missing number in the pattern. Let's examine the numbers and their positions.

</think>

<answer>4</answer>



Question: What is the missing number of the part denoted with a question mark?

Please choose your answer from: [2, 3, 4, 1]

Ground Truth: 2.

Response from Thinking-RFT:

<think>

To solve this, we need to determine the missing number in the sequence. Let's analyze the pattern in the sequence: 2, 3, 3, 1, 1, 1, 2.

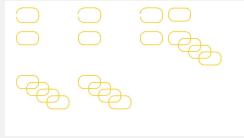
</think>

<answer>

1

</answer>

Figure 15: Examples of the trivial reasoning of fine-tuned Qwen-VL-2-2B model on PuzzleVQA.



Question: Alice has 10 segments of chains of different lengths as shown in the image. The total length of all the segments combined is 19 pieces. She has a saw machine with which a closed piece can be cut opened. She also has a welding machine with which an open piece can be closed. Each cut takes 5 minutes and each welding takes 5 minutes. Initially, she has 3 segments each with 1 open piece as shown in the image. All the other pieces are closed. She now wants to make the longest possible necklace using all the available 19 pieces. Each piece in the necklace would be connected to exactly two other pieces. This would require cutting open some pieces and then joining all the resulting segments together. What is the minimum time in which she can create the necklace?

Please choose your answer from: 40, 41, 35, 3.

Ground Truth: 35.

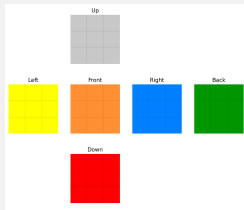
Response from Thinking-RFT:

<think>

To create the longest possible necklace, Alice needs to maximize the number of segments she can use. She has 10 segments, and she wants to use all 19 pieces. The goal is to maximize the number of segments used while ensuring that each segment is connected to exactly two other segments.

</think>

<answer>35</answer>



Question: A 3×3 Rubik's Cube has six different coloured panels: red, green, blue, yellow, orange, and grey. The initial state of the cube in terms of the different colour positions in its six faces is shown in the image. To represent the movements of the cube we use six letters: U for Up, D for Down, L for Left, R for Right, F for Front, B for Back. These letters are used in sequence where you need to perform each letter in the sequence from left to right. Each letter tells you to move that face clockwise by 90 degrees. A number 'n' immediately after a letter denotes that you need to move that face clockwise by $90 \times n$ degrees. For example, 'U R3' would mean rotating the up face 90 degrees clockwise and then rotating the right face 270 degrees clockwise. You perform the move sequence 'B B2' starting from the state shown in the image. What would be the number of small 1×1 red squares in the left face after completing the move sequence?

Please choose your answer from: 3, 4, 7, 0.

Ground Truth: 3.

Response from Thinking-RFT:

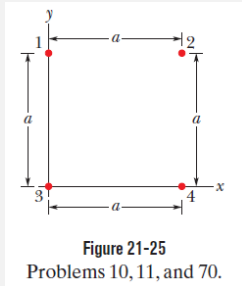
<think>

To solve this problem, we need to understand the sequence of moves and the effect each move has on the cube's state. The sequence is 'B B2', which means rotating the left face clockwise by 90 degrees twice. We need to calculate the final state of the cube after these moves.

</think>

<answer>7</answer>

Figure 16: Examples of the trivial reasoning of fine-tuned Qwen-VL-2-2B model on AlgoPuzzleVQA.



Question: Hint: Please answer the question requiring a floating-point number with two decimal places and provide the final value, e.g., 1.23, 1.34, 1.45, at the end.

Question: In Fig. 21-25, the particles have charges $q_1 = -q_2 = 100\text{nC}$ and $q_3 = -q_4 = 200\text{nC}$, and distance $a = 5.0\text{ cm}$. What is the x component of the net electrostatic force on particle 3?

Ground Truth: 0.17.

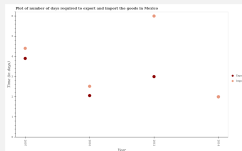
Response from Thinking-RFT:

<think>

To find the x component of the net electrostatic force on particle 3, we need to find the net force acting on particle 3 and then divide it by the charge of particle 3.

</think>

<answer>100</answer>



Question: Hint: Please answer the question requiring an integer answer and provide the final value, e.g., 1, 2, 3, at the end.

Question: What is the difference between the highest and the lowest time required to import goods in Mexico?

Ground Truth: 4.

Response from Thinking-RFT:

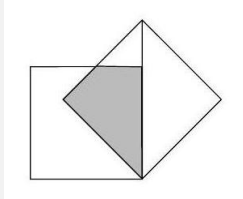
<think>

To determine the difference between the highest and the lowest time required to import goods in Mexico, we need to analyze the data provided in the plot.

</think>

<answer>4</answer>

Figure 17: Examples of the trivial reasoning of fine-tuned Qwen-VL-2-2B model on MathVista.



Question: Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, E, F at the end.

Two squares with side 1 have a common vertex, and the edge of one of them lies along the diagonal of the other. What is the area of the overlap between the squares?
 Choices: A. $\sqrt{2} - 1$

- B. $\frac{\sqrt{2}}{2}$
- C. $\frac{\sqrt{2}+1}{2}$
- D. $\sqrt{2} + 1$
- E. $\sqrt{3} - \sqrt{2}$

Ground Truth: A.

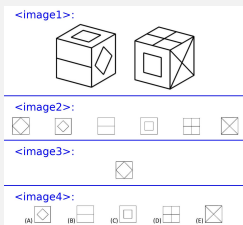
Response from Thinking-RFT:

<think>

To find the area of the overlap between the two squares, we need to find the length of the diagonal of the smaller square.

</think>

<answer>B</answer>



Question: <image1>

In the picture above we see a cube in two different positions.

The six sides of the cube look like this:

<image2>

Which side is opposite to <image3>?

<image4>

Ground Truth: C.

Response from Thinking-RFT:

<think>

To determine which side is opposite to <image3>, we need to identify the opposite side of the cube in <image4>.

</think>

<answer>D</answer>

Figure 18: Examples of the trivial reasoning of fine-tuned Qwen-VL-2-2B model on MathVision.



Question: How many chests are in the image? A. 3 B. 1 C. 2 D. 0

Ground Truth: B.

Response from Thinking-RFT:

<think>

The image shows a bed with a nightstand, a curtain, and a toy chest. There is only one chest in the image.

</think>

<answer> 1 </answer>



Question: How many walls are in the image? A. 1 B. 0 C. 3 D. 4 E. 2

Ground Truth: E.

Response from Thinking-RFT:

<think>

The image shows a classroom with a blackboard, desks, and windows. The walls are painted in a light color, and there are no visible windows on the back wall. Therefore, the correct answer is (A) 1.

</think>

<answer> 1 </answer>

Figure 19: Examples of the reasoning of fine-tuned Qwen-VL-2-2B model on CVBench. Unlike the trivial responses in reasoning tasks, the thinking process in CVBench is not trivial.

 <p>Championdealers.com</p> <p>Question: What type of object is in the photo? Please choose one from list [2012 Acura TL Sedan, 2012 Chrysler Town and Country Minivan,...]</p> <p>Response from Thinking-RFT: <think> The object in the photo is a 2012 Ford Fiesta Sedan.</think> <answer> 2012 Ford Fiesta Sedan</answer></p>	 <p>Question: What type of texture is in the photo? Please choose one from list [spiralled, knitted, sprinkled....]</p> <p>Response from Thinking-RFT: <think> The texture in the photo appears to be woven. </think> <answer>Woven</answer></p>
 <p>Question: What type of aircraft is in the photo? Please choose one from list [MD-90, Beechcraft 1900, DHC-6, MD-87, 777-300, CRJ-900...]</p> <p>Response from Thinking-RFT: <think> The aircraft in the photo is an An-12. </think> <answer>An-12</answer></p>	 <p>Question: What type of object is in the photo? Please choose one from list [okapi, bass, revolver, cup, nautilus, ...]</p> <p>Response from Thinking-RFT: <think> The object in the photo is a Joshua tree. </think> <answer>Joshua_tree</answer></p>
 <p>Question: What type of object is in the photo? Please choose one from list [grilled_cheese_sandwich, caesar_salad, lasagna, ...]</p> <p>Response from Thinking-RFT: <think> The photo shows a dish of paella, which is a traditional Spanish dish made with rice, seafood, vegetables, and spices.</think> <answer>Paella</answer></p>	 <p>Question: What type of object is in the photo? Please choose one from list [miniature_pinscher, yorkshire_terrier,...]</p> <p>Response from Thinking-RFT: <think>The dog in the photo appears to be a Leonberger, a breed known for its long, thick coat and large size. </think> <answer>Leonberger</answer></p>

Figure 20: Examples of Thinking-RFT fine-tuned model responses across six datasets. The thinking content appears not very beneficial to deriving the final answers.



Question: Hint: Please answer the question and provide the correct option letter, e.g., A, B, C, D, at the end.

Question: What time is shown? Answer by typing a time word, not a number. It is () past three.

Choices:

- (A) half
- (B) quarter
- (C) o'clock
- (D) quarter to
- (E) quarter past

Ground Truth: B.

Response from 2B Thinking-RFT: <think>

To determine the correct time shown on the clock, let's analyze the image:

- The clock is red and has the time displayed as 3:15.
 - The time is shown as "quarter to" because it is 15 minutes past the hour.
- </think>

<answer> D </answer>

Response from 2B Think-After-Answer: <answer> E </answer>

<reason> The time shown is quarter past, as indicated by the "15" in the clock. The word "quarter" is used to describe the position of the hour hand between the two halves of the clock. Therefore, the correct option is E. </reason>

Response from 2B No-Thinking-RFT: D

Response from 2B Adaptive-Thinking: <answer> E </answer>

Response from 7B Thinking-RFT: <think> The time shown on the clock is 3:15. To determine the time, we look at the minute hand, which is pointing at 3, and the hour hand, which is between 3 and 4. Since the minute hand is at 15 minutes past the hour, it is 15 minutes past 3. Therefore, the correct answer is "quarter past" because 15 minutes is a quarter of an hour. </think> <answer> B </answer>

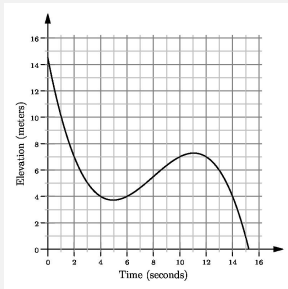
Response from 7B Think-After-Answer: <answer> B </answer>

<reason> The time shown on the clock is 3:15. Since the minute hand is on the 3, which represents 15 minutes past the hour, the correct answer is "quarter past" or "quarter to". However, since the question asks for a time word, "quarter past" is the appropriate choice. </reason>

Response from 7B No-Thinking-RFT: B

Response from 7B Adaptive-Thinking: <think> The time shown is 3:15. The minute part is 15, which is a quarter of an hour. Therefore, the time is 15 minutes past 3. </think> <answer> B </answer>

Figure 21: Comparison of the model response of different fine-tuned model on MathVista.



Question: Malaika is skiing on a mountain. The graph below shows her elevation, in meters, above the base of the mountain as she skis along a trail. In total, how many seconds does she spend at an elevation between 4 and 7 meters?

Ground Truth: 8.

Response from 2B Thinking-RFT: <think>

To find out how many seconds Malaika spends at an elevation between 4 and 7 meters, we need to find the area under the curve between these two values.

</think>

<answer>10</answer>

Response from 2B Think-After-Answer: <answer>10</answer>

<reason>She spends 10 seconds between 4 and 7 meters. The graph shows that she is at an elevation between 4 and 7 meters for 10 seconds.</reason>

Response from 2B No-Thinking-RFT: 10

Response from 2B Adaptive-Thinking: <answer> 10 </answer>

Response from 7B Thinking-RFT: <think> The graph shows the elevation of Malaika over time. To find out how many seconds she spends at an elevation between 4 and 7 meters, we need to look at the portion of the graph where the elevation is between 4 and 7 meters. This is the area under the curve between the horizontal lines at 4 meters and 7 meters. By examining the graph, we can see that Malaika spends approximately 6 seconds at an elevation between 4 and 7 meters. </think> <answer> 6 </answer>

Response from 7B Think-After-Answer: <answer>6</answer>

<reason>The graph shows that Malaika's elevation is between 4 and 7 meters for 6 seconds. This is the time when the line is above the 4-meter mark and below the 7-meter mark on the graph. </reason>

Response from 7B No-Thinking-RFT: 6

Response from 7B Adaptive-Thinking: <think> The graph shows Malaika's elevation over time. To find the time she spends at an elevation between 4 and 7 meters, we need to look at the portion of the graph where the elevation is between 4 and 7 meters. This portion of the graph is between the time of 4 seconds and 12 seconds. Therefore, Malaika spends 8 seconds at an elevation between 4 and 7 meters. </think> <answer> 8 </answer>

Figure 22: Comparison of the model response of different fine-tuned model on MathVision.



Question: How many light switches are in the image?

A. 1 B. 0 C. 2 D. 3

Ground Truth: A.

Response from 2B Thinking-RFT:

<think>

To determine the number of light switches in the image, let's analyze the visible parts of the room:

1. The wall on the left side has several light switches.
2. The wall on the right side has several light switches.
3. The ceiling has several light switches.
4. The floor has no visible light switches.

Based on this analysis, there are 4 light switches in the image.

</think>

<answer> (C) 2 </answer>

Response from 2B Think-After-Answer:

<answer>1</answer>

<reason>There is only one light switch visible in the image, which is located on the wall to the right of the garage area.</reason>

Response from 2B No-Thinking-RFT: (A) 1

Response from 2B Adaptive-Thinking: <answer>1</answer>

Response from 7B Thinking-RFT:

<think>There are two light switches visible in the image, one on the wall near the ceiling and another on the wall near the floor. Therefore, the total number of light switches is 2.</think>

<answer>(A) 1</answer>

Response from 7B Think-After-Answer:

<answer> (A) 1 </answer>

<reason> There is one light switch visible on the wall in the image. </reason>

Response from 7B No-Thinking-RFT: (A) 1

Response from 7B Adaptive-Thinking:

<think> There are two light switches in the image, one on the wall near the ceiling and another on the wall near the floor. </think> <answer> A </answer>

Figure 23: Comparison of the model response of different fine-tuned model on CVBench.

Question: What is the size of the missing part denoted by a question mark?
Please choose your answer from: ['large', 'medium', 'small']

Ground Truth: large.

Response from 2B Thinking-RFT:
<think>
To determine the size of the missing part denoted by the question mark, let's analyze the sizes of the shapes in the image.
</think>

<answer>large</answer>

Response from 2B Think-After-Answer:
<answer>small</answer>
<reason>Since the shapes are arranged in a way that the larger shapes are on the left and the smaller shapes are on the right, the missing part, which is the question mark, should be of the same size as the square, which is medium in size. Therefore, the missing part is small.</reason>

Response from 2B No-Thinking-RFT: medium

Response from 2B Adaptive-Thinking: <answer> large </answer>

Response from 7B Thinking-RFT:
<think> The pattern in the image seems to be that the shapes are getting larger as we move from left to right and top to bottom. The smallest shapes are the hexagon, square, and triangle, and the largest shapes are the large triangle and square. The missing shape is in the middle row, second column, which should be the medium-sized square. Therefore, the size of the missing part denoted by a question mark is medium.</think> <answer> large</answer>

Response from 7B Think-After-Answer:
<answer> large </answer>
<reason> The question mark is placed below the large triangle, indicating that the missing part should be of the same size as the large triangle. </reason>

Response from 7B No-Thinking-RFT: large

Response from 7B Adaptive-Thinking:
<answer> large </answer>

Figure 24: Comparison of the model response of different fine-tuned model on PuzzleVQA.

	5	9	6	13
3	23	10	17	20
22	19	21	1	2
18	11	15	24	16
4	7	12	8	14

Question: The board shown in the image is a sliding puzzle of 5 * 5 tile dimensions. It has 24 numbered tiles and one unoccupied (open) position. Tiles in the same row or column of the open position can be moved by sliding them horizontally or vertically, respectively. All tiles always stay and move inside the red boundary wall, as shown in the image. A move is defined as moving the open position by one tile unit in any available direction. You start from the board position shown in the image and perform exactly 2 moves. How many unique final board positions can you reach?

Please choose your answer from: 3, 1, 4, 5.

Ground Truth: 5.

Response from 2B Thinking-RFT: <think>To determine the number of unique final board positions that can be reached by performing exactly 2 moves, we need to analyze the possible moves that can be made from the given starting position and the constraints imposed by the sliding puzzle rules. </think><answer>5</answer>

Response from 2B Think-After-Answer: <answer>5</answer>

<reason>After performing 2 moves, the board can reach 5 unique final positions: (1, 1), (1, 2), (1, 3), (1, 4), and (1, 5). These positions are reached by sliding the open position in the first row and column of the board. The other positions are not reachable because the board cannot move the open position in the first row or column of the board. Therefore, there are 5 unique final board positions.</reason>

Response from 2B No-Thinking-RFT: 5

Response from 2B Adaptive-Thinking: <answer> 5 </answer>

Response from 7B Thinking-RFT: <think> The puzzle is a 5x5 sliding puzzle with 24 numbered tiles and one empty space. We need to find the number of unique final board positions after exactly 2 moves. Let's analyze the possible moves:

1. Moving the open position to the left or right by one tile unit.
2. Moving the open position up or down by one tile unit.

Since the puzzle is 5x5, there are 4 possible moves for each direction (left, right, up, down). Therefore, there are a total of 8 possible moves. However, we need to perform exactly 2 moves. The number of ways to choose 2 moves out of 8 is given by the combination formula $C(n, k) = n! / (k!(n-k)!)$, where n is the total number of moves and k is the number of moves we want to choose. So, $C(8, 2) = 8! / (2!(8-2)!) = 28$. However, we need to consider that some of these combinations might result in the same final board position. For example, moving the open position to the left and then to the right is the same as moving it to the right and then to the left. Therefore, we need to remove these duplicates.

To do this, we can use the concept of permutations. The number of permutations of 8 items taken 2 at a time is given by $P(n, k) = n! / (n-k)!$, where n is the total number of items and k is the number of items to choose. So, $P(8, 2) = 8! / (8-2)! = 56$. However, we need to divide this by 2 to account for the duplicates mentioned earlier. Therefore, the number of unique final board positions is $28 / 2 = 14$. So, the answer is 14.</think> <answer>5</answer>

Response from 7B Think-After-Answer: <answer> 4 </answer>

<reason> The open position can move to any of the four adjacent tiles (up, down, left, right) in one move. Since there are 24 tiles and each tile can be in one of the 25 positions (including the open position), there are 24 possible starting positions for the open position. After the first move, there are 4 possible positions for the open position (since it can move to any of the four adjacent tiles). Therefore, there are $24 * 4 = 96$ possible final board positions. However, since the open position can move in any of the four directions, there are $4 * 4 = 16$ possible final board positions. </reason>

Response from 7B No-Thinking-RFT: 4

Response from 7B Adaptive-Thinking: <answer> 4 </answer>

Figure 25: Comparison of the model response of different fine-tuned model on AlgoPuzzleVQA.

NeurIPS Paper Checklist

1. Claims

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

Answer: [Yes]

Justification: All contributions are clearly stated in abstract and Sec. 1.

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: The limitations about this paper is discussed in Sec. 6.

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: This paper does not contain theoretical claims.

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: All implementation details are stated in Sec. D.

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 will release all code, data, and models.

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, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: All training and dataset details are discussed in Sec. D and Sec. 4.1.

Guidelines:

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

7. Experiment statistical significance

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

Answer: [No]

Justification: Given the high cost of fine-tuning, we do not report error bars. Please note that in Sec. 4 and Sec. 3 spent huge resources for comparison of thinking and No-Thinking study during RFT across 17 benchmarks and datasets, which makes it prohibitively to run each experiments for multiple times.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

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

Answer: [Yes]

Justification: We have include compute resouces information in Sec. D and Sec. 4.1.

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: The research in this paper is with the NeurIPS Code of Ethics.

Guidelines:

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

10. Broader impacts

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

Answer: [NA]

Justification: There is no social impact of this work.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

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

Answer: [NA]

Justification: This paper poses no such risks.

Guidelines:

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

12. Licenses for existing assets

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

Answer: [Yes]

Justification: All used datasets and other assets are cited and with proper license.

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: [NA]

Justification: No new assets are released in this paper.

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 nor research with human subjects.

Guidelines:

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

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

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

Answer: [NA]

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

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [NA]

Justification: This paper does not involve LLMs as any important, original, or non-standard components

Guidelines:

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