SWITCH-REASONER: LEARN WHEN TO THINK IN MULTITASK MIXTURES VIA REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal Large Language Models (MLLMs) often follow a fixed *Think-then-Answer* paradigm, which is inefficient in heterogeneous multitask settings because simple inputs may not require explicit reasoning while difficult ones can benefit substantially from it. Learning when to think is also unstable during post-training, where imbalanced rollouts can drive the model toward always-thinking or always-direct behavior. We propose `Switch-Reasoner`, a GRPO-based framework that learns to adaptively select reasoning modes for MLLMs. It treats thinking as a virtual tool invocation and allows the model to either answer directly or invoke explicit reasoning before answering. To stabilize this decision, we introduce a dual-level regulation mechanism that balances the overall use of *Thinking Mode* and *Direct Mode* while providing sample-level supervision based on the relative benefit of the two choices. Experiments on 11 multimodal tasks show that Switch-Reasoner reduces unnecessary reasoning while maintaining strong performance, achieving a better accuracy-efficiency trade-off.

1 INTRODUCTION

Multimodal Large Language Models (MLLMs) (Liu et al., 2024; Li et al., 2024a) have achieved remarkable progress in visual understanding and reasoning, supporting a broad range of tasks such as mathematical problem solving, scene understanding, chart reasoning, and visual question answering (Chen et al., 2024; Li et al., 2024b; Wang et al., 2024; Ye et al., 2025). As these models become increasingly general-purpose, reinforcement learning (RL)-based post-training (Liu et al., 2025a; Zhou et al., 2025) has emerged as an effective approach for improving their reasoning capabilities beyond supervised fine-tuning (Schulman et al., 2017; Rafailov et al., 2023). In particular, Group Relative Policy Optimization (GRPO) has become widely adopted for reasoning-oriented training (Guo et al., 2025; Shao et al., 2024) because of its simplicity and strong empirical performance without requiring a separately trained value model.

However, improving reasoning accuracy alone is insufficient for practical MLLM deployment. Existing RL-based reasoning pipelines typically encourage a fixed *Think-then-Answer* behavior (Yao et al., 2025), where the model generates an explicit chain of thought before producing its final answer. Although such deliberation is useful for difficult problems (Feng et al., 2025; Huang et al., 2025a; Rong et al., 2026), it incurs substantial token, latency, and deployment costs. More importantly, this strategy assumes that all inputs require the same amount of reasoning, which is fundamentally mismatched with heterogeneous multimodal tasks (Wu et al., 2026b). As illustrated in Figure 1, some tasks benefit substantially from explicit reasoning, whereas others can often be solved reliably through direct responses. A desirable MLLM should therefore selectively switch between a *Thinking Mode* and a *Direct Mode*, invoking reasoning only when necessary.

Recent work has pursued reasoning efficiency from two main directions. Early-exit (Yang et al., 2026a) and reasoning-pruning (Hou et al., 2025) methods reduce redundant computation by shortening chain-of-thought generation after reasoning has already begun (Nagle et al., 2026). While effective at reducing output length, they do not address the preceding decision of whether explicit reasoning is needed at all. A more closely related line studies adaptive reasoning, allowing models to adaptively select reasoning behaviors across different inputs (Fang et al., 2025; Lou et al., 2025; WANG et al., 2025; Xu et al., 2026; Huang et al., 2026). To learn these behaviors, prior work employs constrained optimization and importance sampling (Zhang et al., 2025a), cold-start supervision (WANG et al.,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

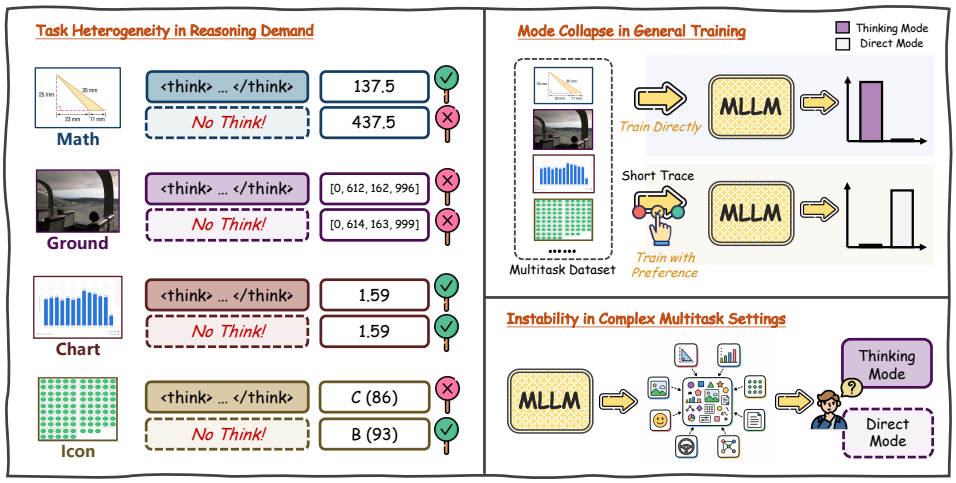


Figure 1: **Motivation for selective reasoning in heterogeneous multimodal tasks.** *Left:* Tasks differ in their need for explicit reasoning. *Top right:* Accuracy-only training can collapse to one mode. *Bottom right:* Heterogeneous mixtures make stable *Thinking/Direct Mode* selection difficult.

2025; Chen et al., 2026), staged training and reward shaping (Tu et al., 2025; Zhu et al., 2025; Yang et al., 2025a), deterministic grouping (Yang et al., 2026b), and difficulty-aware curricula or rewards (Xiao & Gan, 2025; Chen et al., 2026). Although effective in specific settings, these approaches often rely on carefully designed schedules or multi-stage pipelines, which can increase training complexity and cost. In addition, some studies primarily compare the relative merits of long versus short reasoning traces, providing useful empirical evidence but without explicitly learning a policy for deciding when reasoning is necessary (Xiao & Gan, 2025; Hou et al., 2025).

This challenge becomes particularly pronounced under GRPO training in heterogeneous multimodal mixtures. Transient reward advantages and imbalanced rollouts can drive the policy toward a dominant behavior (Yue et al., 2026): frequent rewards for deliberation encourage collapse into the *Thinking Mode*, whereas overly strong penalties induce collapse into the *Direct Mode*. Such mode collapse is especially problematic when the optimal reasoning strategy varies across tasks and individual instances (Yang et al., 2025b). Therefore, the key problem we address is how to stably learn **instance-adaptive policy for selection between the *Thinking Mode* and the *Direct Mode*** under GRPO training, while preventing collapse toward either excessive or insufficient deliberation.

To address this problem, we propose `Switch-Reasoner`, a GRPO-based framework that learns to switch between the *Thinking Mode* and the *Direct Mode* in heterogeneous multimodal mixtures. We first introduce a **Thinking-as-Tool formulation**, which represents explicit reasoning as a virtual tool invocation. Under a switchable prompting protocol, the model can either directly produce the final answer or invoke the thinking tool before answering, turning the whether-to-think decision into an explicit mode-selection action. To stabilize this mode-selection behavior during RL training, we further introduce a **dual-level mode-selection mechanism**. At the global level, it regulates the relative usage of the two modes to avoid collapse toward a dominant behavior; at the sample level, it provides mode-selection supervision based on the relative benefit of thinking and direct answering for each input. Together, these components encourage balanced exploration during training and instance-adaptive allocation of reasoning computation.

The main contributions of this work are summarized as follows:

- We introduce a Thinking-as-Tool formulation with a switchable prompting protocol, turning the whether-to-think decision into an explicit and learnable routing action before generating long reasoning traces.
- We develop a dual-level mode regulation mechanism that combines global balance control with sample-level counterfactual route supervision, preventing mode collapse while encouraging instance-adaptive reasoning allocation.
- Extensive experiments on heterogeneous multimodal benchmarks show that `Switch-Reasoner` achieves a better accuracy–efficiency trade-off.

2 Related Work

2.1 Reinforcement Learning for MLLMs

Reinforcement learning has become an effective approach for improving the reasoning capabilities of multimodal large language models (MLLMs) (Liu et al., 2025a; Zhou et al., 2025). Among existing methods, Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is widely adopted because it estimates relative advantages from multiple sampled outputs without requiring a separate value model (Fang et al., 2026), offering a simple and effective alternative to PPO (Schulman et al., 2017). DeepSeek-R1 (Guo et al., 2025) further demonstrated the effectiveness of GRPO for reasoning-oriented post-training in large language models. Recent work has extended GRPO to multimodal settings, including visual reasoning, video understanding, and omni-modal perception. Text-Debiased Hint-GRPO (Huang et al., 2025a) reduces linguistic bias in multimodal reasoning, R1-VL (Zhang et al., 2025b) introduces step-wise optimization for vision-language learning, and R1-Omni (Zhao et al., 2025) applies reinforcement learning to omni-modal emotion recognition. Video-R1 (Feng et al., 2025), VideoChat-R1 (Li et al., 2025), and Visual-RFT (Liu et al., 2025b) further demonstrate the scalability of RL-based post-training across video and vision-centric tasks. Different from these works, which primarily improve task performance through general-purpose RL objectives, our work focuses on learning when explicit reasoning is necessary in heterogeneous multimodal mixtures and stabilizing this behavior under GRPO training.

2.2 Efficient Reasoning in MLLMs

Recent work on efficient reasoning in LLMs reduces unnecessary computation either by shortening reasoning traces after generation or by adaptively determining whether and how deeply to deliberate (Huang et al., 2025b). Early-exit methods terminate reasoning trajectories once sufficient evidence is available (Yang et al., 2026a; Nagle et al., 2026), while pruning methods compress redundant chain-of-thought content (Hou et al., 2025). Other studies enable adaptive reasoning through explicit think-or-direct decisions (Fang et al., 2025; Zhang et al., 2025a; Lou et al., 2025), staged optimization (Tu et al., 2025; Zhu et al., 2025), or step-wise budget allocation (Wang et al., 2026b). Building on these advances, recent work extends adaptive reasoning to MLLMs, where visual, audio, and heterogeneous multimodal inputs introduce additional variation in reasoning demand. Selective reasoning has been explored in vision-language models (WANG et al., 2025; Yang et al., 2026b), difficulty-aware multimodal training (Xiao & Gan, 2025; Chen et al., 2026), and omni-modal reasoning settings (Yang et al., 2025a; Wu et al., 2026a). Other methods consider multi-path visual reasoning (Huang et al., 2026; Li et al., 2026), adaptive inference computation (Xu et al., 2025), and efficient reasoning for video or tool-augmented visual tasks (Liu et al., 2026; Wang et al., 2026a). In contrast, our work focuses on stably learning instance-adaptive selection between *Thinking Mode* and *Direct Mode* in heterogeneous multimodal mixtures under GRPO.

3 Switch-Reasoner

3.1 Preliminaries

Group Relative Policy Optimization (GRPO) is a variant of Proximal Policy Optimization (PPO) that can improve mathematical, visual, and other multimodal reasoning abilities. It maintains a current policy π_θ and an old policy π_{old} from the previous iteration. Given a prompt $q \sim \rho_Q$, GRPO samples a group of outputs $o_1, o_2, \ldots, o_G$ from π_{old} and optimizes π_θ by maximizing:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\rho_Q}\mathbb{E}_{o\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G} f_\epsilon\left(\frac{\pi_\theta(o_i|q)}{\pi_{\text{old}}(o_i|q)}, \hat{A}_i\right)\right] - \beta\,\mathbb{D}_{KL}[\pi_\theta\|\pi_{\text{ref}}]\,, \tag{1}$$

where β is the KL regularization hyperparameter, and $f_\epsilon(x, y) = \min(xy, \text{clip}(x, 1-\epsilon, 1+\epsilon)y)$. $\hat{A}_i$ denotes the advantage of the i-th output, computed from relative rewards within the sampled group.

For each question q, a group of outputs $\{o_1, o_2, \ldots, o_G\}$ is sampled from π_{old}. A reward function $\mathcal{R}$ scores these outputs, yielding $\mathbf{r} = \{r_1, r_2, \ldots, r_G\}$, where $r_i = \mathcal{R}(q, o_i)$. The mean reward is $\mu = \frac{1}{G}\sum_{i=1}^{G} r_i$, and the standard deviation is $\sigma = \sqrt{\frac{1}{G}\sum_{i=1}^{G}(r_i - \mu)^2}$. The normalized advantage

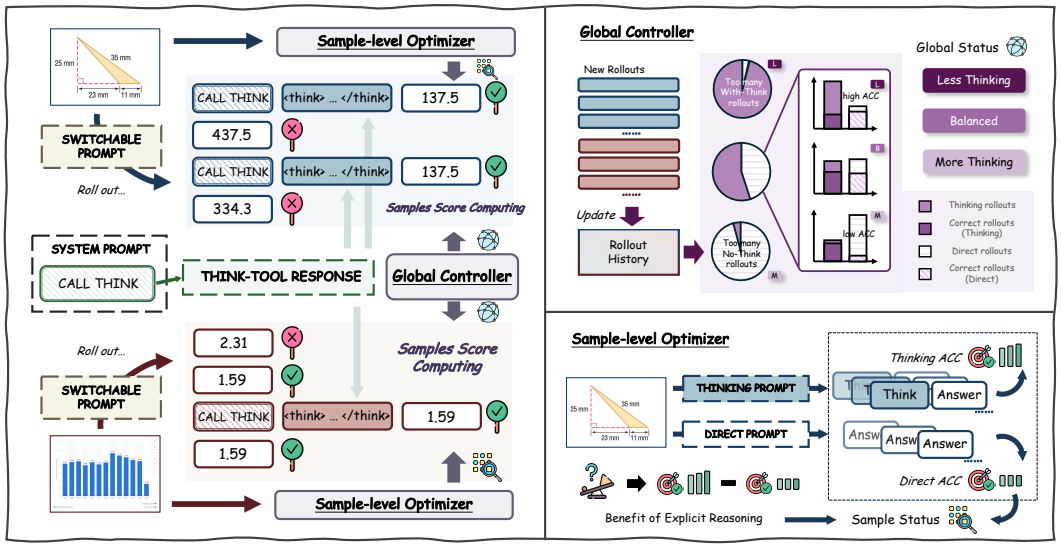


Figure 2: **Overall framework of `Switch-Reasoner`.** The model first selects between *Direct Mode* and *Thinking Mode* through a switchable prompt. Tool invocation triggers explicit reasoning, while direct answers bypass it. During training, a global controller regulates mode balance, and a sample-level optimizer provides counterfactual route supervision.

of the i-th rollout is $\hat{A}_i = \frac{r_i - \mu}{\sigma}$. This normalization gives advantages zero mean and unit variance within each group, stabilizing gradients and promoting consistent optimization.

3.2 Thinking-as-Tool Switchable Prompt

To enable selective reasoning in MLLMs, we reformulate *Thinking Mode* as an explicit tool invocation rather than an implicit always-on generation pattern. The key idea is to expose whether explicit reasoning is performed as an observable and optimizable action, allowing the model to determine when additional reasoning is necessary.

We introduce a virtual reasoning tool T_{think} with a switchable prompting protocol. Given a multimodal input $q = (x_v, x_t)$, the model first generates an initial response under a constrained format with two possible behaviors: directly producing the final answer, referred to as *Direct Mode*, or emitting a tool-call token to invoke T_{think}, referred to as *Thinking Mode*. This design turns the reasoning-mode selection decision into a discrete and observable action.

When the thinking tool is invoked, a follow-up think-tool response switches the model into *Thinking Mode* and requires intermediate reasoning before the final answer. Otherwise, the model outputs the answer directly. During rollout, the model first follows the switchable protocol; a tool call continues the trajectory with the follow-up think-tool response, while a direct answer is treated as a Direct-answering-mode trajectory. Generation can therefore be viewed as an implicit two-stage process: mode selection followed by either direct answering or explicit reasoning.

Formally, let $m \in 0, 1$ denote the mode decision, where $m = 1$ invokes the thinking tool and $m = 0$ denotes *Direct Mode*. The policy is factorized as

$$\pi_\theta(o, m|q) = \pi_\theta(m|q)\pi_\theta(o|q, m), \tag{2}$$

where o denotes the final output sequence. Although written in this form, the mode decision is realized implicitly through the output format and is influenced by the input question, visual information, and decoding context.

Compared with conventional *Think-then-Answer* prompting, this formulation allows simple samples to bypass unnecessary reasoning while providing a discrete routing interface that can be optimized directly under reinforcement learning. All prompt details are provided in Appendix B.

3.3 GLOBAL MODE BALANCE CONTROL

Although Thinking-as-Tool enables explicit switching between *Thinking Mode* and *Direct Mode*, a switchable interface alone does not guarantee stable mode preference during reinforcement learning. Short-term reward advantages and sampling fluctuations may cause the policy to overuse one mode. Explicit reasoning can temporarily obtain higher rewards through longer trajectories, whereas *Direct Mode* may dominate when it is more efficient or stable at certain stages.

To address this issue, we introduce **Global Mode Balance Control**, an online delayed controller that provides a batch-level signal for regulating the relative use of the two modes. At each training iteration, the controller applies the signal estimated from the previous rollout batch to the current reward computation, and then updates its state using the statistics of the current batch. Specifically, after reward evaluation, we collect format-valid trajectories and compute the number and accuracy of *Thinking Mode* and *Direct Mode* rollouts: n_{think}, n_{direct}, a_{think}, and a_{direct}. Their usage ratios are

$$p_{\text{direct}} = \frac{n_{\text{direct}}}{n_{\text{think}} + n_{\text{direct}}}, \qquad p_{\text{think}} = \frac{n_{\text{think}}}{n_{\text{think}} + n_{\text{direct}}}. \tag{3}$$

Only format-valid trajectories are used because malformed outputs cannot reliably reflect the actual reasoning-mode decision.

Let

$$\Delta a = a_{\text{direct}} - a_{\text{think}} \tag{4}$$

denote the accuracy advantage of *Direct Mode*. We generate a global control signal B according to performance difference and usage ratio:

$$B = \begin{cases} +B_g, & \Delta a > \delta \text{ and } p_{\text{direct}} < \tau_{\text{low}}, \\ -B_g, & -\Delta a > \delta \text{ and } p_{\text{direct}} > \tau_{\text{high}}, \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

When $B > 0$, *Direct Mode* performs better but is underused, so the training reward relatively favors direct answers by penalizing Thinking-mode trajectories. When $B < 0$, *Thinking Mode* performs better while *Direct Mode* is overused, so the training reward relatively favors reasoning by penalizing Direct-answering-mode trajectories.

To preserve exploration when one mode becomes excessively dominant, we introduce an extreme imbalance safeguard:

$$B = \begin{cases} +B_f, & p_{\text{think}} > \pi_{\text{high}}, \\ -B_f, & p_{\text{think}} < \pi_{\text{low}}. \end{cases} \tag{6}$$

A count-ratio criterion can additionally identify nearly single-mode batches and restore exploration when one route almost disappears. The updated control signal is stored and used in the next training iteration. This one-batch delayed design avoids circular dependence between the rewards of the current batch and the statistics computed from those same rewards, while still providing a coarse-grained routing preference based on empirical performance and mode usage.

3.4 SAMPLE-LEVEL FINE-GRAINED OPTIMIZATION

Global mode control provides only a batch-level preference, whereas the need for explicit reasoning is inherently sample-dependent. Even within the same batch, some inputs can be solved reliably with concise direct responses while others require multi-step visual or textual reasoning. We therefore introduce **Sample-level Fine-grained Optimization** to provide counterfactual route supervision for each training sample.

For each input q_i, we evaluate the current policy under two fixed choices: *Direct Mode* and *Thinking Mode*. Specifically, we construct a direct prompt that forces an answer without invoking the thinking tool and a thinking prompt that requires explicit reasoning before the final answer:

$$\{o_{i,k}^{\text{direct}}\}_{k=1}^{K} \sim \pi_\theta(\cdot|q_i, m=0), \qquad \{o_{i,k}^{\text{think}}\}_{k=1}^{K} \sim \pi_\theta(\cdot|q_i, m=1). \tag{7}$$

Let $R_{\text{acc}}(q, o)$ denote the accuracy component of the task reward. We estimate the empirical success rate of each route as

$$s_i^{\text{direct}} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left[R_{\text{acc}}(q_i, o_{i,k}^{\text{direct}}) > 0\right], \tag{8}$$

$$s_i^{\text{think}} = \frac{1}{K} \sum_{k=1}^{K} \mathbb{1}\left[R_{\text{acc}}(q_i, o_{i,k}^{\text{think}}) > 0\right]. \tag{9}$$

Their difference reflects the sample-specific benefit of explicit reasoning:

$$\Delta_i = s_i^{\text{think}} - s_i^{\text{direct}}. \tag{10}$$

Based on this comparison, each sample is assigned to one of three categories:

$$c_i = \begin{cases} \texttt{must-think}, & \Delta_i \geq \delta_{\text{think}}, \\ \texttt{safe-direct}, & s_i^{\text{direct}} \geq \tau_{\text{direct}} \text{ and } \Delta_i \leq -\delta_{\text{safe}}, \\ \texttt{uncertain}, & \text{otherwise.} \end{cases} \tag{11}$$

`must-think` indicates that *Thinking Mode* provides a clear improvement. In contrast, `safe-direct` is assigned only when *Direct Mode* is already reliable and *Thinking Mode* does not provide a sufficient benefit; in our default setting, this requires the empirical thinking gain to fall below a negative safety margin. For uncertain samples, we impose no strong preference to avoid introducing noisy route supervision.

Together, the global controller captures the overall tendency of the two modes, while sample-level comparison identifies the inputs that benefit from reasoning and those that can be answered directly.

3.5 Reward Computation

We design the reward to jointly encourage answer correctness, format compliance, balanced mode usage, and sample-level route selection. For each response o_i, the base reward is

$$r_i^{\text{base}} = (1-\alpha) r_i^{\text{acc}} + \alpha r_i^{\text{fmt}}, \tag{12}$$

where r_i^{acc} measures task correctness and r_i^{fmt} measures compliance with the switchable protocol.

We apply the global control signal from the previous rollout batch as a multiplicative penalty on the temporarily disfavored mode:

$$r_i^{\text{global}} = r_i^{\text{base}} \cdot \eta^{\mathbb{1}[(B>\epsilon \wedge m_i=1) \vee (B<-\epsilon \wedge m_i=0)]}, \tag{13}$$

where $m_i = 1$ denotes *Thinking Mode*, $m_i = 0$ denotes *Direct Mode*, and $\eta \in (0,1)$ is the penalty factor. The controller state B is updated after scoring the current batch and used in the next iteration.

For sample-level route supervision, we define a preferred-route sign

$$\gamma_i = \begin{cases} +1, & c_i = \texttt{must-think}, \\ -1, & c_i = \texttt{safe-direct}, \\ 0, & c_i = \texttt{uncertain}, \end{cases} \tag{14}$$

and a selected-route sign

$$\mu_i = \begin{cases} +1, & m_i = 1, \\ -1, & m_i = 0. \end{cases} \tag{15}$$

The route reward is then

$$r_i^{\text{route}} = w_{c_i} \gamma_i \mu_i, \tag{16}$$

where $w_{c_i} = w_{\text{think}}$ for `must-think`, $w_{c_i} = w_{\text{direct}}$ for `safe-direct`, and $w_{c_i} = 0$ for `uncertain`. Thus, the route reward is positive when the selected route matches the preferred route, negative when it mismatches, and zero for uncertain samples.

The final reward is

$$r_i^* = r_i^{\text{global}} + r_i^{\text{route}}. \tag{17}$$

This reward is used for group-wise advantage estimation in GRPO.

The relevant notation table and algorithms are provided in Appendix A and Appendix C.

Table 1: **Comparison on multitask multimodal benchmarks.** The reported results are averaged over the last three validation steps except for Vanilla and ± denotes the standard deviation. * indicates the variant without sample-level fine-grained correction. See Section 4.2 for details.

Method	Mathematics	Chart/Doc	Grounding	General	Overall	
					Score	Think Rate
Qwen3-VL-4B						
Vanilla	26.20	91.87	57.60	40.43	48.76	0.00%
GRPO-Thinking	71.90	91.52	61.40	64.84	70.66 ± 0.54	100.00%
GRPO-Direct	43.40	91.47	61.07	64.08	65.02 ± 0.23	0.00%
`Switch-Reasoner*`	66.83	92.06	62.13	66.47	70.79 ± 0.68	85.73%
`Switch-Reasoner`	67.23	91.66	60.27	68.26	**71.60** ± 0.30	51.53%
Qwen3-VL-8B						
Vanilla	29.90	92.22	58.60	42.16	50.53	0.00%
GRPO-Thinking	73.57	93.01	60.13	66.19	71.86 ± 0.34	100.00%
GRPO-Direct	48.23	92.30	58.87	64.03	65.83 ± 0.07	0.00%
`Switch-Reasoner*`	68.43	92.67	58.20	68.60	72.00 ± 0.24	30.25%
`Switch-Reasoner`	68.80	92.62	58.93	68.78	**72.22** ± 0.16	37.73%

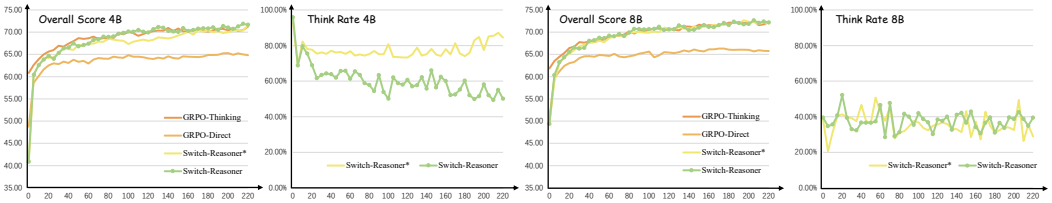


Figure 3: Training dynamics of validation performance and thinking rate. Refer to Section 4.3.

4 Experiment

4.1 Experimental Setup

Dataset and Evaluation Metrics. We conduct experiments on a heterogeneous multitask benchmark with 11 representative multimodal tasks, covering mathematical and visual reasoning, chart/document understanding, localization, and general multimodal understanding. We follow the standard metric of each task, and more details are provided in the Appendix D.

Architecture and Counterparts. We use Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct as vanilla base models (Bai et al., 2025) for RL training. We compare our method with Vanilla model, all-thinking and all-direct variants, as well as a variant without sample-level fine-grained correction, to evaluate the effects of switchable reasoning and instance-level route supervision.

Implementation Details. All models are trained with GRPO using EasyR1.[1] In the switchable reasoning setting, each prompt allows the model either to directly produce the final answer or to invoke the thinking tool before responding. We set the maximum prompt length to 8,192 tokens and train all models for 20 epochs using 8 NVIDIA H20 GPUs, each equipped with 96 GB of memory. Additional hyperparameter details and parameter ablation results are provided in Appendix E.

4.2 Comparison Experiment

Table 1 compares different training strategies across four categories of multimodal tasks. The reported results are averaged over the last three validation steps. GRPO-Thinking achieves strong performance but requires a 100% thinking rate, while GRPO-Direct eliminates explicit thinking entirely, leading to clear performance degradation.

Our method achieves a better trade-off between performance and thinking cost, with particularly clear gains on Qwen3-VL-4B. Switch-Reasoner reaches the best overall score among all 4B methods,

[1] `https://github.com/hiyouga/easyr1`

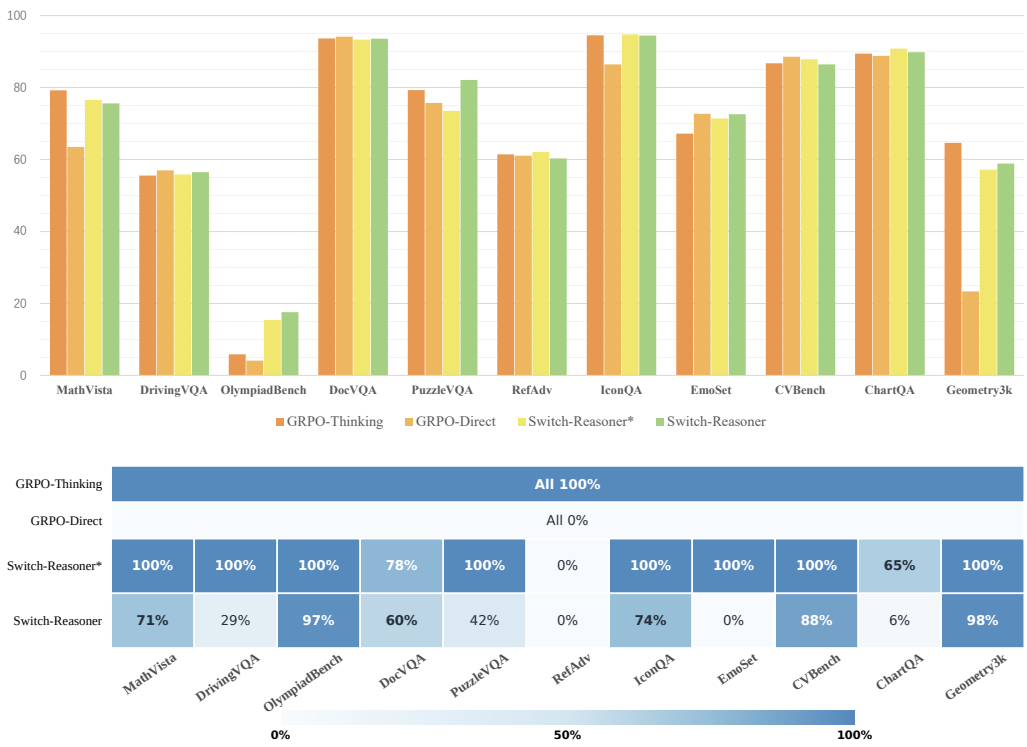


Figure 4: Dataset-level score and thinking rate of the 4B model. See Section 4.3 for details.

improving over both GRPO-Thinking and GRPO-Direct while reducing the thinking rate to 51.53%. On Qwen3-VL-8B, Switch-Reasoner also achieves the best overall score, reaching 72.22 with only a 37.73% thinking rate. These results indicate that the proposed switching strategy can effectively reduce unnecessary reasoning while preserving, and in some cases improving, multimodal performance. The improvement is especially evident on the General task, which contains a more diverse mixture of datasets and therefore better reflects the benefit of adaptive reasoning-mode selection.

The comparison between Switch-Reasoner* and Switch-Reasoner further demonstrates the benefit of sample-level fine-grained correction. Compared with the variant without this correction, the full method achieves a higher overall score with a comparable or lower thinking rate. This suggests that fine-grained route supervision helps the model make more reliable sample-level switching decisions.

4.3 Visualization Analysis

We conduct visualization experiments to further analyze the behavior of Switch-Reasoner during training and evaluation. Specifically, we visualize the validation performance curves, dataset-level score, and dataset-level thinking rates to understand how the model learns to allocate reasoning computation across different tasks.

Visualization of Training Dynamics. To analyze the training behavior of different reasoning strategies, we visualize the validation score and thinking rate across training steps. As shown in Figure 3, Switch-Reasoner gradually learns to invoke thinking selectively. It achieves higher validation scores than GRPO-Direct while using substantially fewer thinking trajectories than GRPO-Thinking. This indicates that our method does not simply suppress reasoning uniformly, but learns an adaptive reasoning policy that balances task performance and thinking cost.

Visualization of Dataset-level Performance. We further visualize dataset-level score and thinking rates to examine how different methods behave across heterogeneous tasks. As shown in Figure 4, GRPO-Thinking consistently relies on explicit reasoning for all datasets, whereas GRPO-Direct always produces direct answers without thinking. In contrast, Switch-Reasoner assigns adaptive thinking rates to different datasets while maintaining competitive or stronger accuracy. Reasoning-intensive datasets tend to receive higher thinking rates, while some recognition- or understanding-

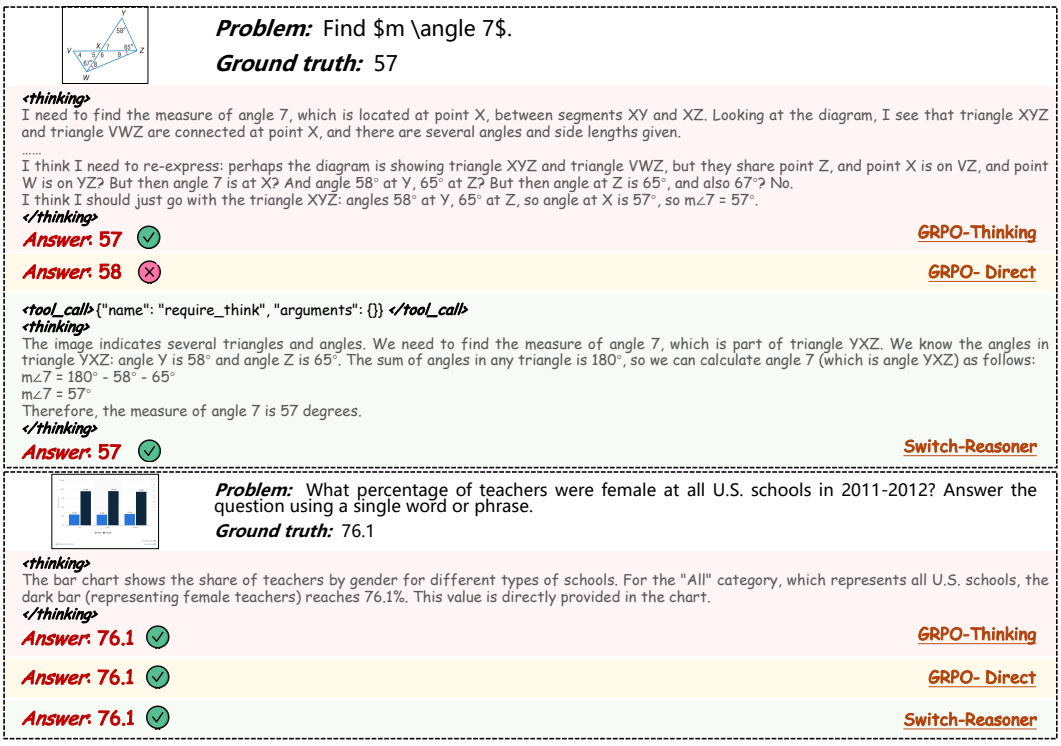


Figure 5: Case studies on different datasets with the 4B model. Refer to Section 4.4.

oriented datasets can be handled effectively with direct answering. These results show that Switch-Reasoner learns to adapt its reasoning mode according to dataset-level demands, reducing unnecessary thinking while preserving overall performance.

4.4 Case Study

As illustrated in Figure 5, we present two representative examples to illustrate the adaptive behavior of Switch-Reasoner. For a geometry problem requiring multi-step angle calculation, direct answering fails to produce the correct result, while Switch-Reasoner invokes the thinking mode and correctly derives the answer through explicit reasoning. In contrast, for a chart-based question asking for a directly observable percentage, Switch-Reasoner answers correctly without unnecessary reasoning, achieving the same result as the always-thinking baseline with lower computation cost. These cases demonstrate that Switch-Reasoner can selectively allocate reasoning effort according to the difficulty and reasoning demand of each sample. Broader case studies are provided in Appendix F.

5 Conclusion

Existing GRPO-based reasoning approaches often follow a fixed *Think-then-Answer* paradigm, which is inefficient for heterogeneous multimodal tasks and can induce mode collapse during adaptive training. We present **`Switch-Reasoner`**, a GRPO-based framework that learns when to invoke explicit reasoning in multimodal large language models. By treating thinking as a virtual tool invocation, the model can adaptively choose between *Thinking Mode* and *Direct Mode* rather than adhering to a fixed reasoning pattern. To mitigate mode collapse in heterogeneous multitask training, we introduce a dual-level regulation mechanism that balances global mode usage while providing sample-level route supervision. Experiments across 11 multimodal tasks show that **`Switch-Reasoner`** achieves a stronger trade-off between performance and thinking cost, attaining competitive overall scores while substantially reducing unnecessary reasoning. Visualizations and case studies further demonstrate that the learned switching policy captures task- and sample-level reasoning demands. Beyond binary switching, this tool-based formulation may support richer reasoning tools and adaptive allocation of thinking resources. We hope this work inspires future research on adaptive reasoning policies for more efficient multimodal intelligence.

AI Assistance Disclosure

We used generative AI tools to improve the clarity and readability of the manuscript, assist with the visualization of experimental results. All AI-assisted content was reviewed by the authors. The authors take full responsibility for the accuracy and integrity of the final paper, including its text, figures, and scientific claims.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Shuang Chen, Hangyu Guo, Yimeng Ye, Shijue Huang, Wenbo Hu, Jiayu Chen, Manyuan Zhang, Haoxi Li, Song Guo, and Nanyun Peng. ARES: Multimodal adaptive reasoning via difficulty-aware token-level entropy shaping. In *ICLR*, pp. 147474–147512, 2026.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, pp. 24185–24198, 2024.

Yew Ken Chia, Vernon Toh, Deepanway Ghosal, Lidong Bing, and Soujanya Poria. PuzzleVQA: Diagnosing multimodal reasoning challenges of language models with abstract visual patterns. In *ACL Findings*, pp. 16259–16273, 2024.

Charles Corbière, Simon Roburin, Syrielle Montariol, Antoine Bosselut, and Alexandre Alahi. DRIVINGVQA: A dataset for interleaved visual chain-of-thought in real-world driving scenarios. In *EACL Findings*, pp. 3309–3333, 2026.

Qihua Dong, Kuo Yang, Lin Ju, Handong Zhao, Yitian Zhang, Yizhou Wang, Huimin Zeng, Jianglin Lu, and Yun Fu. Ref-Adv: Exploring MLLM visual reasoning in referring expression tasks. In *ICLR*, 2026.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: LLM learns when to think. In *NeurIPS*, volume 38, pp. 151268–151295, 2025.

Yiyang Fang, Wenke Huang, Pei Fu, Yihao Yang, Kehua Su, Zhenbo Luo, Jian Luan, and Mang Ye. EMO-R3: Reflective reinforcement learning for emotional reasoning in multimodal large language models. In *CVPR*, pp. 745–755, 2026.

Kaituo Feng, Kaixiong Gong, Bohao Li, Zonghao Guo, Yibing Wang, Tianshuo Peng, Junfei Wu, Xiaoying Zhang, Benyou Wang, and Xiangyu Yue. Video-R1: Reinforcing video reasoning in MLLMs. In *NeurIPS*, volume 38, pp. 99114–99137, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *ACL*, 2024.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. ThinkPrune: Pruning long chain-of-thought of LLMs via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Qihan Huang, Weilong Dai, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. Boosting MLLM reasoning with text-debiased Hint-GRPO. In *ICCV*, pp. 4848–4857, 2025a.

Shijue Huang, Hongru Wang, Wanjun Zhong, Zhaochen Su, Jiazhan Feng, Bowen Cao, and Yi R Fung. AdaCtrl: Towards adaptive and controllable reasoning via difficulty-aware budgeting. *arXiv preprint arXiv:2505.18822*, 2025b.

Yixu Huang, Tinghui Zhu, and Muhao Chen. Learning adaptive reasoning paths for efficient visual reasoning. *arXiv preprint arXiv:2604.14568*, 2026.

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

Xinhao Li, Ziang Yan, Desen Meng, Lu Dong, Xiangyu Zeng, Yinan He, Yali Wang, Yu Qiao, Yi Wang, and Limin Wang. VideoChat-R1: Enhancing spatio-temporal perception via reinforcement fine-tuning. *arXiv preprint arXiv:2504.06958*, 2025.

Zejun Li, Yingxiu Zhao, Jiwen Zhang, Siyuan Wang, Yang Yao, Runzhou Zhao, Jun Song, Bo Zheng, and Zhongyu Wei. Mixture-of-Visual-Thoughts: Exploring context-adaptive reasoning mode selection for general visual reasoning. In *ICLR*, pp. 81981–82016, 2026.

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. In *CVPR*, pp. 26763–26773, 2024b.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, pp. 26296–26306, 2024.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. Reinforcement learning meets large language models: A survey of advancements and applications across the LLM lifecycle. *arXiv preprint arXiv:2509.16679*, 2025a.

Shuming Liu, Mingchen Zhuge, Changsheng Zhao, Jun Chen, Lemeng Wu, Zechun Liu, Chenchen Zhu, Zhipeng Cai, Chong Zhou, Haozhe Liu, Ernie Chang, Saksham Suri, Hongyu Xu, Qi Qian, Wei Wen, Balakrishnan Varadarajan, Zhuang Liu, Hu Xu, Florian Bordes, Raghuraman Krishnamoorthi, Bernard Ghanem, Vikas Chandra, and Yunyang Xiong. VideoAuto-R1: Video auto reasoning via thinking once, answering twice. In *CVPR*, pp. 32829–32839, 2026.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-RFT: Visual reinforcement fine-tuning. In *ICCV*, pp. 2034–2044, 2025b.

Chenwei Lou, Zewei Sun, Xinnian Liang, Meng Qu, Wei Shen, Wenqi Wang, Yuntao Li, Qingping Yang, and Shuangzhi Wu. AdaCoT: Pareto-optimal adaptive chain-of-thought triggering via reinforcement learning. *arXiv preprint arXiv:2505.11896*, 2025.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. In *ACL*, pp. 6774–6786, 2021a.

Pan Lu, Liang Qiu, Jiaqi Chen, Tony Xia, Yizhou Zhao, Wei Zhang, Zhou Yu, Xiaodan Liang, and Song-Chun Zhu. IconQA: A new benchmark for abstract diagram understanding and visual language reasoning. In *NeurIPS*, 2021b.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *ACL Findings*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *WACV*, pp. 2200–2209, 2021.

Alliot Nagle, Jakhongir Saydaliev, Dhia Garbaya, Michael Gastpar, Ashok Vardhan Makkuva, and Hyeji Kim. TERMINATOR: Learning optimal exit points for early stopping in chain-of-thought reasoning. *arXiv preprint arXiv:2603.12529*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, volume 36, pp. 53728–53741, 2023.

Xuankun Rong, Wenke Huang, Tingfeng Wang, Daiguo Zhou, Bo Du, and Mang Ye. SafeGRPO: Self-rewarded multimodal safety alignment via rule-governed policy optimization. In *CVPR*, pp. 7901–7911, 2026.

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

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

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *NeurIPS*, volume 37, pp. 87310–87356, 2024.

Songjun Tu, Jiahao Lin, Qichao Zhang, Xiangyu Tian, Linjing Li, Xiangyuan Lan, and Dongbin Zhao. Learning when to think: Shaping adaptive reasoning in R1-style models via multi-stage RL. In *NeurIPS*, volume 38, pp. 15181–15207, 2025.

Chaoyang Wang, Kaituo Feng, Dongyang Chen, Zhongyu Wang, Zhixun Li, Sicheng Gao, Meng Meng, Xu Zhou, Manyuan Zhang, Yuzhang Shang, et al. AdaTooler-V: Adaptive tool-use for images and videos. In *ACL Findings*, pp. 18058–18072, 2026a.

Jiaqi WANG, Kevin Qinghong Lin, James Cheng, and Mike Zheng Shou. Think or Not? Selective Reasoning via Reinforcement Learning for Vision-Language Models. In *The Exploration in AI Today Workshop at ICML 2025*, 2025.

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

Yuxin Wang, Jiahao Lu, Qifeng Wu, Shicheng Fang, Chuanyuan Tan, Yining Zheng, Xuanjing Huang, and Xipeng Qiu. AdaptR1: Reinforcement learning based adaptive interleaved thinking in multi-hop question answering. *arXiv preprint arXiv:2605.31062*, 2026b.

Shu Wu, Chenxing Li, Wenfu Wang, Hao Zhang, Hualei Wang, Meng Yu, and Dong Yu. Audio-Thinker: Guiding large audio language model when and how to think via reinforcement learning. In *AAAI*, volume 40, pp. 33962–33970, 2026a.

Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in LLMs. In *ICLR*, 2026b.

Wenyi Xiao and Leilei Gan. Fast-Slow Thinking GRPO for large vision-language model reasoning. In *NeurIPS*, volume 38, pp. 171601–171631, 2025.

Wenyue Xu, Xu Wan, Wei Wang, Wenqi Huang, Wotao Yin, Shengjie Zhao, and Mingyang Sun. AdapThink: Adaptive thinking preferences for reasoning language models. In *ACL Findings*, pp. 9808–9825, 2026.

Zhuoyan Xu, Khoi Duc Nguyen, Preeti Mukherjee, Saurabh Bagchi, Somali Chaterji, Yingyu Liang, and Yin Li. Learning to inference adaptively for multimodal large language models. In *ICCV*, pp. 3552–3563, 2025.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In *ICLR*, 2026a.

Dongchao Yang, Songxiang Liu, Disong Wang, Yuanyuan Wang, Guanglu Wan, and Helen Meng. Omni-AutoThink: Adaptive multimodal reasoning via reinforcement learning. *arXiv preprint arXiv:2512.03783*, 2025a.

Jingyuan Yang, Qirui Huang, Tingting Ding, Dani Lischinski, Danny Cohen-Or, and Hui Huang. EmoSet: A large-scale visual emotion dataset with rich attributes. In *ICCV*, pp. 20383–20394, 2023.

Qi Yang, Bolin Ni, Shiming Xiang, and Houwen Peng. R-4B: Incentivizing general-purpose auto-thinking in MLLMs via bi-mode annealing and reinforce learning. In *CVPR*, pp. 7891–7900, 2026b.

Senqiao Yang, Junyi Li, Xin Lai, Jinming Wu, Wei Li, Zejun MA, Bei Yu, Hengshuang Zhao, and Jiaya Jia. VisionThink: Smart and efficient vision language model via reinforcement learning. In *NeurIPS*, volume 38, pp. 95187–95227, 2025b.

Huanjin Yao, Qixiang Yin, Jingyi Zhang, Min Yang, Yibo Wang, Wenhao Wu, Fei Su, Li Shen, Minghui Qiu, Dacheng Tao, and Jiaxing Huang. R1-ShareVL: Incentivizing reasoning capabilities of multimodal large language models via Share-GRPO. In *NeurIPS*, volume 38, 2025.

Mang Ye, Xuankun Rong, Wenke Huang, Bo Du, Nenghai Yu, and Dacheng Tao. A survey of safety on large vision-language models: Attacks, defenses and evaluations. *arXiv preprint arXiv:2502.14881*, 2025.

Tan Yue, Qiong Wu, and Dongyan Zhao. MARS: Multimodal adaptive reasoning model for avoiding overthinking. In *AAAI*, volume 40, pp. 34539–34547, 2026.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. AdaptThink: Reasoning models can learn when to think. In *EMNLP*, pp. 3716–3730, 2025a.

Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-VL: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *ICCV*, pp. 1859–1869, 2025b.

Jiaxing Zhao, Xihan Wei, and Liefeng Bo. R1-Omni: Explainable omni-multimodal emotion recognition with reinforcement learning. *arXiv preprint arXiv:2503.05379*, 2025.

Guanghao Zhou, Panjia Qiu, Cen Chen, Jie Wang, Zheming Yang, Jian Xu, and Minghui Qiu. Reinforced MLLM: A survey on RL-based reasoning in multimodal large language models. *arXiv preprint arXiv:2504.21277*, 2025.

Yekun Zhu, Guang Chen, and Chengjun Mao. Think in blocks: Adaptive reasoning from direct response to deep reasoning. *arXiv preprint arXiv:2508.15507*, 2025.

APPENDIX

A NOTATION TABLE

Table 2 summarizes the main symbols used throughout the paper. We list the variables associated with the switchable prompting protocol, global mode-balance control, sample-level route supervision, and reward computation, together with the equations where they first appear.

Table 2: **Notation used in `Switch-Reasoner`.** Summary of key variables and operations. The Equation column indicates where each symbol first appears in the main text.

Symbol	Description	Equation
$q = (x_v, x_t)$	Original multimodal input with visual content x_v and text prompt x_t	–
T_{think}	Virtual thinking tool used to trigger explicit reasoning	–
G	Number of rollouts sampled per prompt in GRPO	Eq. 1
$\hat{A}_i$	Normalized group-relative advantage of the i-th rollout	Eq. 1
$m \in \{0, 1\}$	Mode decision: 1 for Thinking, 0 for Direct	Eq. 2
π_θ	Trainable policy model	Eq. 2
o	Final output sequence generated under the selected mode	Eq. 2
$n_{\text{think}}, n_{\text{direct}}$	Number of format-valid Thinking and Direct-answering rollouts	Eq. 3
$p_{\text{think}}, p_{\text{direct}}$	Usage ratios of *Thinking Mode* and *Direct Mode*	Eq. 3
$a_{\text{think}}, a_{\text{direct}}$	Accuracy of Thinking and Direct-answering rollouts	Eq. 4
Δa	Accuracy advantage of *Direct Mode* over *Thinking Mode*	Eq. 4
B	Global mode-control signal	Eq. 5
B_g, B_f	Magnitudes of global and safeguard signals	Eq. 5, Eq. 6
K	Number of counterfactual rollouts sampled for each fixed route	Eq. 7
$s_i^{\text{direct}}, s_i^{\text{think}}$	Empirical success rates of Direct and Thinking routes for sample i	Eq. 8, Eq. 9
Δ_i	Sample-specific benefit of explicit reasoning	Eq. 10
c_i	Route category: `must-think`, `safe-direct`, or `uncertain`	Eq. 11
$r_i^{\text{acc}}, r_i^{\text{fmt}}$	Accuracy reward and format-compliance reward	Eq. 12
r_i^{base}	Base reward combining correctness and format compliance	Eq. 12
r_i^{global}	Reward after applying global mode-balance adjustment	Eq. 13
r_i^{route}	Sample-level route reward	Eq. 16
r_i^*	Final reward used for GRPO advantage estimation	Eq. 17

B DETAILED PROMPT DESIGN

We provide the prompt templates used by **`Switch-Reasoner`** for switchable reasoning and counterfactual route supervision. The switchable prompt allows the model to either answer directly or invoke the thinking tool, while the direct and thinking prompts are used to construct fixed-route rollouts for sample-level comparison.

Switchable Prompt

```
Answer the user's question by default.

First, attempt to answer directly WITHOUT using any tools. If you can
    answer reliably, output ONLY the final answer inside <answer> and
    </answer>.

If you cannot answer reliably, do NOT guess and do NOT output <
    thinking>. Instead, call the function tool "require_think" by
    outputting a single tool call inside <tool_call> and </tool_call>.

Formatting rules (STRICT):
```

```
- Either output:
  <answer>...</answer>
- Or output:
  <tool_call>
  {"name":"require_think","arguments":{}}
  </tool_call>
- Do not include any other text outside the tags.
- Do not call any other tool in this stage.

Here is the question:
[Question]
```

Think-Tool Response Prompt

```
I noticed you tried to call a tool. Please do not call any tools.

Now You FIRST think about the reasoning process as an internal
    monologue and then provide the final answer. Put your concise
    reasoning inside <thinking>...</thinking>, and put the final
    answer inside <answer>...</answer>.
```

Direct Prompt

```
[Question]

Answer directly. Do NOT use any tools. Do NOT include any reasoning,
    analysis, or intermediate steps.

Output ONLY the final answer, wrapped exactly in the tags below (no
    extra text):
<answer>...</answer>
```

Thinking Prompt

```
[Question]

Solve the problem carefully using explicit reasoning.

Output ONLY the final response in exactly this format:
<thinking>...</thinking>
<answer>...</answer>
```

C ALGORITHM

Algorithm 1 summarizes the training procedure of **`Switch-Reasoner`**. The model first generates switchable rollouts to decide between *Direct Mode* and *Thinking Mode*. It then performs counterfactual direct and thinking rollouts for sample-level route supervision, computes the final reward with both global and sample-level regulation, updates the global controller using current batch statistics, and optimizes the policy with GRPO.

Algorithm 1 `Switch-Reasoner`

Input: Dataset $\mathcal{D} = \{q_i = (x_v, x_t)\}$, policy model π_θ, reference model π_{ref}, rollout number G, counterfactual number K, global controller thresholds $\tau_{\text{low}}, \tau_{\text{high}}, \pi_{\text{low}}, \pi_{\text{high}}$, route thresholds $\delta_{\text{think}}, \tau_{\text{direct}}, \delta_{\text{safe}}$
Output: Trained switchable reasoning policy π_θ
Initialize global control signal $B \leftarrow 0$
for each training iteration **do**
  Sample a batch of multimodal inputs $\{q_i\} \sim \mathcal{D}$
  Generate G switchable rollouts for each q_i using π_θ
  for each rollout **do**
    if the model outputs a valid `require_think` tool call **then**
      Continue generation with the follow-up tool response
      Set mode $m_i \leftarrow 1$
    else if the model outputs a valid direct answer **then**
      Set mode $m_i \leftarrow 0$
    else
      Mark the rollout as format-invalid
    end if
  end for
  Sample-level route supervision
  for each input q_i **do**
    Sample forced Direct rollouts $\{o_{i,k}^{\text{direct}}\}_{k=1}^{K} \sim \pi_\theta(\cdot|q_i, m = 0)$
    Sample forced Thinking rollouts $\{o_{i,k}^{\text{think}}\}_{k=1}^{K} \sim \pi_\theta(\cdot|q_i, m = 1)$
    Compute s_i^{direct} and s_i^{think}
    Compute $\Delta_i = s_i^{\text{think}} - s_i^{\text{direct}}$
    if $\Delta_i \geq \delta_{\text{think}}$ **then**
      $c_i \leftarrow$ `must-think`
    else if $s_i^{\text{direct}} \geq \tau_{\text{direct}}$ and $\Delta_i \leq -\delta_{\text{safe}}$ **then**
      $c_i \leftarrow$ `safe-direct`
    else
      $c_i \leftarrow$ `uncertain`
    end if
  end for
  Reward computation
  Compute r_i^{base} from r_i^{acc} and r_i^{fmt}
  Apply previous-batch global signal B to obtain r_i^{global}
  Compute route reward r_i^{route} according to c_i and m_i
  Set final reward $r_i^* = r_i^{\text{global}} + r_i^{\text{route}}$
  Global mode-balance update
  Compute $n_{\text{think}}, n_{\text{direct}}, a_{\text{think}}, a_{\text{direct}}$ from format-valid rollouts
  Compute $p_{\text{think}}, p_{\text{direct}}$ and $\Delta a = a_{\text{direct}} - a_{\text{think}}$
  Update B using the global controller and the extreme-imbalance safeguard
  Estimate group-relative advantages $\hat{A}_i$ from $\{r_i^*\}$
  Update π_θ with the GRPO objective
end for
return π_θ

D Dataset and Evaluation Details

We evaluate **`Switch-Reasoner`** on 11 multimodal benchmarks covering mathematical reasoning, chart and document understanding, visual grounding, visual recognition, and general visual reasoning. Table 3 summarizes the task type, evaluation metric, answer format, prompt instruction, and a representative example for each dataset. For the metrics, ACC denotes task-specific accuracy, including exact-match accuracy for classification and option-based tasks and rule-based answer equivalence for mathematical tasks; Relaxed ACC allows small numerical deviations following the standard ChartQA protocol; ANLS measures normalized text similarity for document VQA; and

IoU@0.5 evaluates grounding correctness by checking whether the predicted bounding box reaches an intersection-over-union score of at least 0.5.

Table 3: **Detailed dataset descriptions.** Summary of task types, metrics, answer formats, prompt instructions, and representative examples.

Dataset	Task	Metric	Answer	Prompt Instruction	Example
Geometry3k (Lu et al., 2021a)	Geometry Diagram Problem Solving	ACC	Value	Original geometry question.	 **Q:** Find y. **A:** 3
MathVista (Lu et al., 2024)	Visual Mathematical Reasoning	ACC	Mixed	Original query or option-letter prompt.	 **Q:** Find x. A. 5 B. 10 C. $10\sqrt{3}$ D. 20 **A:** C
ChartQA (Masry et al., 2022)	Visual and Logical Chart QA	Relaxed ACC	Phrase	Answer the question using a single word or phrase.	 **Q:** Sum of lung's bars? **A:** 38
DocVQA (Mathew et al., 2021)	VQA on Document Images	ANLS	Phrase	Answer the question using a single word or phrase.	 **Q:** Item no.? **A:** 79
RefAdv (Dong et al., 2026)	Referring Expression Grounding	IoU@0.5	Box	Return only a Python-style [x1, y1, x2, y2] list on a 0–1000 scale.	 **Q:** Right dog among a pair. **A:** [263, 428, 315, 498]
CVBench (Tong et al., 2024)	2D/3D Vision-centric Understanding	ACC	Option	Answer with only the option letter.	 **Q:** How many cats? A. 0 B. 2 C. 3 D. 1 **A:** D
DrivingVQA (Corbière et al., 2026)	Driving Scene VQA	ACC	Option	Answer with only the option letter(s). If there are multiple answers, separate them with commas.	 **Q:** Speed limit after signs? A. 45 B. 50 C. 60 D. 70 km/h **A:** D
EmoSet (Yang et al., 2023)	Visual Emotion Recognition	ACC	Label	Answer using one emotional category.	 **Q:** Which emotion? **A:** awe
IconQA (Lu et al., 2021b)	Abstract Diagram QA	ACC	Option	Answer with only the option letter.	 **Q:** What shape? A. cube B. sphere C. cone **A:** C
OlympiadBench (He et al., 2024)	Olympiad-level Scientific Reasoning	ACC	Mixed	Original olympiad problem.	 **Q:** Relation of C and line AB? **A:** $C \notin AB$
PuzzleVQA (Chia et al., 2024)	Visual Puzzle Reasoning	ACC	Option	Answer with only the option letter.	 **Q:** Missing shape? A. hexagon B. square C. triangle D. pentagon **A:** B

E Hyperparameter Details and Ablation Experiments

For global mode balance control, we use $\tau_{\text{low}} = 0.20$, $\tau_{\text{high}} = 0.80$, $\delta = 0.02$, $\pi_{\text{low}} = 0.02$, and $\pi_{\text{high}} = 0.98$ by default, where τ_{low} and τ_{high} define the underuse and overuse thresholds for *Direct Mode*, δ is the accuracy-margin deadband, and π_{low} and π_{high} are used by the extreme-imbalance safeguard. For sample-level route supervision, we use $K = 4$ counterfactual rollouts for each fixed route and set $w_{\text{think}} = w_{\text{direct}} = 0.03$, $\delta_{\text{think}} = 0.5$, $\tau_{\text{direct}} = 0.75$, and $\delta_{\text{safe}} = 0.25$. Here, K controls the number of forced-route rollouts, w_{think} and w_{direct} control the route-reward strengths, δ_{think} determines when a sample is classified as `must-think`, and τ_{direct} and δ_{safe} define the `safe-direct` criterion.

To examine the sensitivity of **`Switch-Reasoner`**, we conduct hyperparameter ablations on several important sample-level routing parameters, as shown in Table 4. In the table, **`Switch-Reasoner`** denotes the default setting with $K = 4$, $\delta_{\text{think}} = 0.5$, $\tau_{\text{direct}} = 0.75$, $\delta_{\text{safe}} = 0.25$, and $w_{\text{think}} = w_{\text{direct}} = 0.03$. The counterfactual sampling ablation changes K to 2 or 8; the must-think margin ablation changes δ_{think} to 0.25 or 0.75; the safe-direct ablation uses a loose setting ($\tau_{\text{direct}} = 0.60, \delta_{\text{safe}} = 0.0$) or a strict setting ($\tau_{\text{direct}} = 0.90, \delta_{\text{safe}} = 0.50$); and the reward-weight ablation favors either *Direct Mode* ($w_{\text{direct}} = 0.04, w_{\text{think}} = 0.02$) or *Thinking Mode* ($w_{\text{think}} = 0.04, w_{\text{direct}} = 0.02$). Overall, **`Switch-Reasoner`** remains relatively stable under different counterfactual sampling budgets and route-reward weight settings, suggesting that the sample-level supervision does not rely on a single fragile configuration. In contrast, the safe-direct criterion has a stronger effect on the accuracy-efficiency trade-off: a loose criterion substantially reduces the

thinking rate but also hurts overall performance, while a stricter criterion preserves stronger accuracy with a moderate thinking rate. The must-think margin also changes the learned routing behavior, where larger margins generally reduce unnecessary thinking by requiring clearer evidence before assigning samples to the `must-think` category.

Table 4: **Hyperparameter ablation on multitask multimodal benchmarks.** Results are averaged over the last three validation steps and ± denotes the standard deviation of the overall score.

Variant	Mathematics	Chart/Doc	Grounding	General	Overall Score	Overall Think Rate
Qwen3-VL-4B						
Switch-Reasoner	67.23	91.66	60.27	68.26	71.60 ± 0.30	51.53%
Counterfactual $K = 2$	64.90	91.48	61.40	66.77	70.43 ± 0.45	72.05%
Counterfactual $K = 8$	65.17	91.90	60.13	66.71	70.41 ± 0.39	57.47%
Must-think margin 0.25	67.03	91.86	63.20	66.34	70.82 ± 0.39	80.55%
Must-think margin 0.75	66.30	92.04	63.13	67.97	71.60 ± 0.21	42.36%
Safe-direct loose	62.43	91.99	61.73	63.50	68.32 ± 0.35	13.28%
Safe-direct strict	67.40	91.69	63.80	66.08	70.77 ± 0.30	82.48%
Direct-favored weight	66.37	92.10	61.47	66.87	70.87 ± 0.29	74.33%
Thinking-favored weight	66.33	91.83	60.13	66.62	70.56 ± 0.35	79.85%
Qwen3-VL-8B						
Switch-Reasoner	68.80	92.62	58.93	68.78	72.22 ± 0.16	37.73%
Counterfactual $K = 2$	68.53	92.63	58.00	68.48	71.93 ± 0.09	37.01%
Counterfactual $K = 8$	69.80	92.67	58.60	67.98	71.95 ± 0.21	34.95%
Must-think margin 0.25	67.77	92.46	58.27	68.26	71.66 ± 0.25	37.24%
Must-think margin 0.75	67.00	92.55	58.13	67.70	71.22 ± 0.25	29.10%
Safe-direct loose	62.30	92.23	59.33	66.69	69.87 ± 0.88	12.55%
Safe-direct strict	69.80	92.77	57.53	68.73	72.28 ± 0.24	38.60%
Direct-favored weight	67.10	92.42	59.27	68.42	71.71 ± 0.35	33.32%
Thinking-favored weight	67.60	92.54	58.33	68.11	71.57 ± 0.15	33.63%

F BROADER CASE STUDY



Problem: Considering the relative positions of the bus and the book in the image provided, where is the bus located with respect to the book?
A. Left B. right Answer with only the option letter.
Ground truth: B

<thinking>
First, I need to identify the bus and the book in the image. The bus is a large, green and white vehicle in the center-right of the image, on the street. The book is not visible in the image. Therefore, there is no book to compare the bus's position to. This suggests that the question may be based on a misunderstanding or mislabeling. Since the book is not present, I cannot determine a relative position. However, if I must choose between "left" or "right" based on the bus's position in the image, the bus is on the right side of the street relative to the viewer's perspective, and the book is not there. But since the question specifically asks for the bus's position with respect to the book, and the book does not exist, the correct answer might be that the bus is not to the left or right of a book. However, since the options are only "left" or "right", and the bus is clearly on the right side of the image (relative to the book's absence), I must choose "right" as the bus is positioned to the right of the book's non-existence — which is logically inconsistent. But given the options and the context, perhaps the question intended to refer to something else, like a person holding a book? There is no book visible. I must conclude that the question is flawed. However, since I must answer, and the bus is on the right side of the image, I will go with "right" as the bus is to the right of the book's non-existence, which is the only logical choice given the options. Alternatively, perhaps there is a book on the ground? No, there isn't. I think the question is invalid, but if forced to pick, I'll say "right" because the bus is on the right side of the image, and the book is not there, so relative to the book's position (which is undefined), the bus is to the right. But this is flawed reasoning. Given the constraints, I choose B. right.
</thinking>

Answer: B ✓ — GRPO-Thinking

Answer: A ✗ — GRPO- Direct

<tool_call>{"name": "require_think", "arguments": {}}***</tool_call>***
<thinking>
The image shows a street scene with a green and white bus prominently positioned in the center-right portion of the frame. There is no visible book in the image. Therefore, the question is based on an incorrect assumption. Since there is no book in the image, I cannot determine the relative position of the bus with respect to a non-existent object. However, if the question intends to refer to a specific object that might be misidentified, I would need to clarify. But given the constraints, I must note that the book is not present.
</thinking>

Answer: B ✓ — Switch-Reasoner

Figure 6: Broader case studies (1)



Problem: Return only a Python-style list of four integers [x1, y1, x2, y2] on a 0-1000 coordinate scale. Do not include any explanation.Please provide the bounding box coordinate of the region this sentence describes: the person walking in the middle of the bridge, not facing toward the camera.

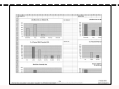
Ground truth: [369, 500, 408, 674]

<thinking>
I need to locate the person walking in the middle of the bridge, not facing toward the camera. Looking at the image, there are several people in the background. The person in the middle of the bridge is walking away from the camera, and they are wearing a light-colored top and dark pants. This person is positioned near the center of the glass-enclosed walkway. I will identify their bounding box by finding the coordinates of the top-left and bottom-right corners of their body.
</thinking>

Answer: **[363, 503, 407, 678]** ✓ — GRPO-Thinking

Answer: **[367, 502, 409, 677]** ✓ — GRPO- Direct

Answer: **[369, 504, 407, 677]** ✓ — Switch-Reasoner



Problem: Which product shows a higher percentage of 'extremely ' likely usage when compared to 'very ' likely usage; UXL or Marathon ? Answer the question using a single word or phrase.

Ground truth: MARATHON

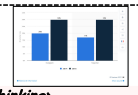
<thinking>
First, I need to compare the "Extremely" vs. "Very" likelihood usage for both UXL and Marathon.
For UXL (left side, top chart): The "Extremely" bar is at approximately 35% and the "Very" bar is at approximately 55%. So, "Very" is higher than "Extremely".
For Marathon (right side, top chart): The "Extremely" bar is at approximately 40% and the "Very" bar is at approximately 20%. So, "Extremely" is higher than "Very".
Now, comparing the two: UXL has "Very" higher than "Extremely", while Marathon has "Extremely" higher than "Very". The question asks which product shows a higher percentage of "Extremely" when compared to "Very".
Marathon shows a higher percentage of "Extremely" (40%) compared to "Very" (20%). UXL shows a lower percentage of "Extremely" (35%) compared to "Very" (55%).
Therefore, Marathon is the product where "Extremely" is higher than "Very".
Final Answer: Marathon
</thinking>

Answer: **Marathon** ✓ — GRPO-Thinking

Answer: **ULTRAMET XL** ✗ — GRPO- Direct

<tool_call>{"name": "require_think", "arguments": {}} ***</tool_call>***
<thinking>
To determine which product shows a higher percentage of 'extremely' likely usage compared to 'very' likely usage, I need to analyze the bar charts for both UXL and Marathon.
For UXL, the bar chart labeled "Likelihood to use Ultamet XL" shows:
- "Extremely" likely usage is approximately 35%.
- "Very" likely usage is approximately 55%.
For Marathon, the bar chart labeled "Likelihood to Use Marathon" shows:
- "Extremely" likely usage is approximately 45%.
- "Very" likely usage is approximately 20%.
Comparing the two, Marathon has a higher percentage of 'extremely' likely usage (45%) compared to 'very' likely usage (20%) than UXL, which has 35% for 'extremely' and 55% for 'very'.
Thus, Marathon is the product that shows a higher percentage of 'extremely' likely usage when compared to 'very' likely usage.
</thinking>

Answer: **Marathon** ✓ — Switch-Reasoner



Problem: Which social medium experienced a larger increase in penetration rate from 2014 to 2016? Answer the question using a single word or phrase.

Ground truth: Snapchat

<thinking>
To determine which social medium experienced a larger increase in penetration rate from 2014 to 2016, I need to calculate the difference for each platform.
For Instagram:
- 2014 penetration rate: 20%
- 2016 penetration rate: 30%
- Increase: 30% - 20% = 10%
For Snapchat:
- 2014 penetration rate: 17%
- 2016 penetration rate: 30%
- Increase: 30% - 17% = 13%
Comparing the two increases: Instagram increased by 10%, while Snapchat increased by 13%. Therefore, Snapchat experienced a larger increase.
Final Answer: Snapchat
</thinking>

Answer: **Snapchat** ✓ — GRPO-Thinking

Answer: **Instagram** ✗ — GRPO- Direct

<tool_call>{"name": "require_think", "arguments": {}} ***</tool_call>***
<thinking>
To determine which social medium experienced a larger increase in penetration rate from 2014 to 2016, I need to calculate the difference in penetration rates for both Instagram and Snapchat between these two years.
For Instagram:
- Penetration rate in 2014: 20%
- Penetration rate in 2016: 30%
- Increase: 30% - 20% = 10%
For Snapchat:
- Penetration rate in 2014: 17%
- Penetration rate in 2016: 30%
- Increase: 30% - 17% = 13%
Comparing the two increases, Snapchat had a larger increase (13%) than Instagram (10%).
Therefore, Snapchat experienced a larger increase in penetration rate.
</thinking>

Answer: **Snapchat** ✓ — Switch-Reasoner

Figure 7: Broader case studies (2)