

TIGER: Bridging the Multimodal Reasoning-Access Gap via Modality Counterfactuals

Gregory Kang Ruey Lau^{1,2*} Minh Huynh Nguyen^{1*} Bryan Kian Hsiang Low¹

¹ Department of Computer Science, National University of Singapore, Singapore

² CNRS@CREATE, 1 Create Way, #08-01 Create Tower, Singapore 138602

{greglau, minhnh, lowkh}@comp.nus.edu.sg

Abstract

While Multimodal Large Language Models (MLLMs) exhibit strong reasoning on text inputs, they often fail on semantically equivalent visual inputs. By rendering text problems as images, we isolate this discrepancy and identify a reasoning-access gap: models can perceive visual content but fail to engage the reasoning mechanisms effectively used for text-based tasks. To address this, we propose TIGER (Text-to-Image Gap-targeted Training for Enhanced Reasoning), which automatically converts text-only corpora into multimodal training data by mining modality counterfactuals, instances where a model succeeds on text but fails on the equivalent image. Using image-conditioned Group Relative Policy Optimization (GRPO), TIGER provides targeted supervision without manually curated multimodal datasets. Experiments show that TIGER consistently reduces the modality gap and improves visual reasoning on benchmarks such as MathVerse and EMMA, including for RLVR-trained models. Activation analyses further indicate that TIGER enables visual representations to better engage reasoning-relevant subspaces in the language backbone. These findings suggest that robust multimodal reasoning depends not only on perception, but also on reliable visual access to existing reasoning machinery.

1. Introduction

Large Language Models have demonstrated impressive complex reasoning capabilities, as exemplified by reasoning-oriented systems such as OpenAI-o1 (OpenAI, 2024) and DeepSeek-R1 (Guo et al., 2025). Naturally, there is an expectation that Multimodal Large Language Models (MLLMs) (Bai et al., 2023; Wang et al., 2024c; Chen et al., 2024b) built with LLM language backbones will inherit these "text-native" reasoning capabilities when processing multimodal input beyond text such as images. However, this expectation remains largely unmet: despite efforts in visual instruction tuning (Liu et al., 2023; Zhao et al., 2024), multimodal chain-of-thought prompting (Ge et al., 2023; Zhang et al., 2024b), and Reinforcement Learning with Verifiable Reward (RLVR) (Yu et al., 2025; Li et al., 2026; Xiao et al., 2026), MLLMs still struggle with problems that require sustained logical or mathematical reasoning over visual inputs (Hao et al., 2025; Xu et al., 2026).

A dominant explanation in the field attributes this failure to perception (Tong et al., 2024; Liu et al., 2025; Xiao et al., 2026; Wang et al., 2026). Under the common perception-then-reasoning view, if MLLMs can correctly extract relevant visual content, it should be able to reason over it as effectively as it does over text (Amizadeh et al., 2020; Zhou et al., 2025). Consequently, many recent methods focus on improving visual recognition, grounding, or perception-aware reward design (Yu et al., 2025; Xiao et al., 2026; Wang et al., 2026). While perception is a necessary capability for MLLMs, we challenge the assumption that it is sufficient to unlock reasoning capabilities. We argue that multimodal reasoning failures are not just failures of "sight" but failures of *reasoning access*: a breakdown in the model's ability to route correctly perceived visual evidence into its existing capable reasoning machinery.

To isolate this phenomenon, we study MLLMs in a controlled setting using *modality counterfactuals*. To avoid the confounding factors present in existing multimodal benchmarks where perception, grounding, abstraction, and reasoning

*Equal contribution

Accepted to FoGen 2026: Foundations of Deep Generative Models: Understanding Memorization, Generalization, and Reasoning, an ICML 2026 workshop (non-archival).

evaluations are typically entangled, we render text-only reasoning problems as images. This transformation preserves the semantic content of the problem while changing only the input modality, enabling a direct comparison between text-conditioned and image-conditioned reasoning. Our diagnosis across multiple model families (e.g., Qwen2.5-VL (Bai et al., 2025b), Qwen3-VL (Bai et al., 2025a), and InternVL3.5 (Wang et al., 2025a)) reveals a substantial and consistent modality-dependent reasoning gap: models perform markedly better when the same problem is presented in text compared to image. Crucially, this gap is not due to perception errors, as models achieve high OCR accuracy on the images and fails to perform well despite extracting text correctly. Feeding the OCR-extracted text to the models largely recovers the model’s original performance, suggesting that the required reasoning capability exists within the language backbone but is not accessed under visual conditioning.

Motivated by this diagnosis, we propose TIGER (Text-to-Image Gap-targeted Training for Enhanced Reasoning), a framework for closing the reasoning-access gap without requiring expensive manually curated multimodal datasets. Starting from text-only reasoning datasets with verifiable answers, TIGER converts the modality gap itself into a supervision signal. Specifically, it renders image versions of text problems, queries the MLLM with both modalities, and mines text-success/image-failure instances, where the model can reason correctly from text but fails on the semantically equivalent image. TIGER then trains the model on these instances using image-conditioned Group Relative Policy Optimization (GRPO) (Shao et al., 2024), enabling more effective exploration of image-conditioned reasoning pathways than fixed-trajectory methods such as Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) (Rafailov et al., 2023). Our key contributions are as follows:

- We provide a controlled study of the text–image reasoning-access gap in MLLMs to disentangle reasoning failures from perceptual errors. By evaluating semantically identical problems in text and image-rendered form, we show that MLLMs often fail to reason over visual inputs for problems they can solve in text, even when the text in the image is correctly recognized (Sec. 3).
- We introduce our framework TIGER that (1) repurposes text-native reasoning training data into multimodal training examples through modality counterfactuals (Sec. 4.1), and (2) trains MLLMs using this data via image-conditioned GRPO to achieve enhanced reasoning performance (Sec. 4).
- We empirically demonstrate that TIGER improves generalization across diverse multimodal reasoning benchmarks, including domains beyond the training data (Sec. 5). We further show that TIGER is complementary to existing RLVR-based visual reasoning methods (Yang et al., 2025b; Xiao et al., 2026; Li et al., 2026), enabling additional gains when combined with them (Sec. 5.2).
- We analyze how TIGER drives performance gains by increasing reasoning-subspace activation (RSA), which are internal MLLM latent components linked to effective text-side reasoning. Using activation patching, we further demonstrate how fine-tuning aligns visual inputs with the model’s existing reasoning mechanisms (Sec. 5.3).

Overall, these results suggest that reliable multimodal intelligence requires not only stronger perception, but also explicit training to align visual inputs with models’ internal reasoning mechanisms.

2. Related works

Visual Reasoning Benchmarks. While reasoning evaluation traditionally focused on text-only tasks (Clark et al., 2018; Hendrycks et al., 2021a; Cobbe et al., 2021; Hendrycks et al., 2021b; Suzgun et al., 2023), recent benchmarks increasingly ground reasoning in diverse visual inputs. These range from static charts (Masry et al., 2022; 2025) and mathematical diagrams (Lu et al., 2024; Zhang et al., 2024a; Wang et al., 2024b; Qiao et al., 2024) to dynamic video and 3D spatial environments (Li et al., 2024; Fu et al., 2025; Mangalam et al., 2023; Wu et al., 2024; Yang et al., 2025a), alongside expert-level multidisciplinary suites (Yue et al., 2025; Hao et al., 2025; Yao et al., 2025). Although essential for assessing end-to-end multimodal competence, these datasets often entangle perception, domain knowledge, and reasoning, obscuring modality-specific failures. Our work complements this landscape via a controlled text-to-image protocol that preserves task semantics by construction and avoids confounding variables. This design minimizes confounding factors and explicitly isolates the modality gap, allowing us to directly measure whether MLLMs can deploy their inherent reasoning capabilities when equivalent problems are presented visually.

Post-training for Multimodal Reasoning. Post-training has become a central paradigm for improving reasoning in LLMs (Guo et al., 2025; Kumar et al., 2025), and is increasingly being adapted to multimodal models through SFT, Direct Preference Optimization (DPO) (Rafailov et al., 2023), and Reinforcement Learning objectives such as Group

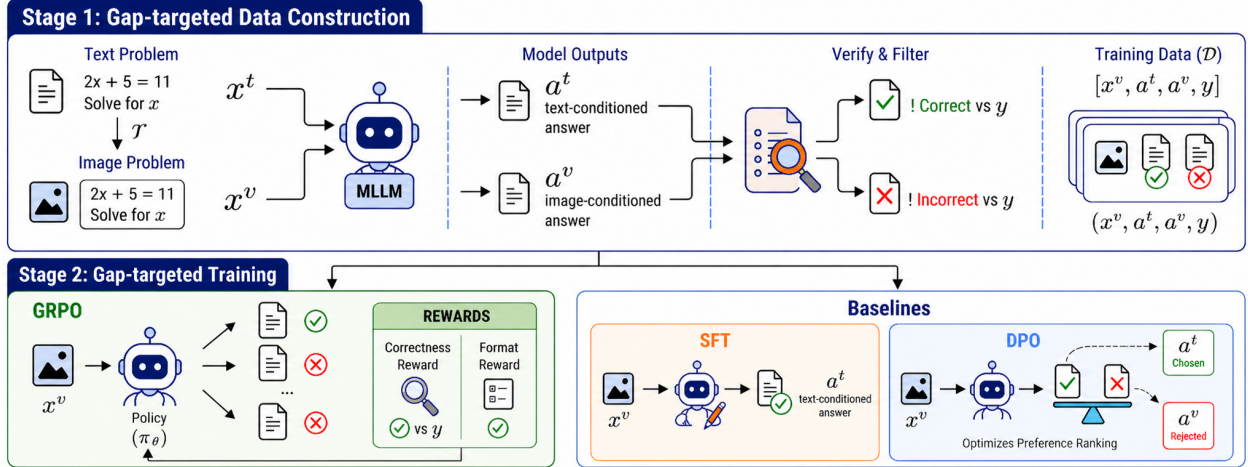


Figure 1. Overview of the proposed pipeline. In Stage 1, we construct training samples by querying a MLLM with text and image versions of the same problem, then verifying and filtering the resulting outputs to retain pairs with correct text-conditioned answers and incorrect image-conditioned answers. In Stage 2, the resulting data is used for gap-targeted training using GRPO to improve image-conditioned reasoning and comparing with alternatives like SFT and DPO.

Relative Policy Optimization (GRPO) (Shao et al., 2024). Recent work follows three main directions. SFT-based methods transfer text-derived chain-of-thought and instruction-following behaviors to vision-language models (Chen et al., 2024a; Wei et al., 2026; Zhang et al., 2025b). DPO-based methods construct multimodal preference pairs to improve visual faithfulness or encourage stronger grounding in image and video inputs (Wang et al., 2025b; Yu et al., 2024; Li et al., 2023; Zhang et al., 2025a; Wang et al., 2024a; Pi et al., 2024). RLVR-style methods instead optimize models with verifiable rewards or process-level supervision over visual perception, grounding, or intermediate reasoning (Yang et al., 2025b; Yu et al., 2025; Xiao et al., 2026; Li et al., 2026). Our work is complementary to these methods but addresses a distinct diagnostic question: whether a model preserves its reasoning behavior under semantically equivalent modality shifts. By focusing on problems solved in text but not in rendered visual form, our evaluation isolates a modality-dependent reasoning gap and directly compares how SFT, DPO, and GRPO bridge visual inputs to downstream reasoning.

3. The Modality Reasoning-Access Gap

To investigate the modality-dependent reasoning-access gap discussed in Sec. 1, we construct a controlled diagnostic setting using modality counterfactuals. This allows us to compare the model’s reasoning performance across semantically identical inputs presented through different modalities.

Setup. Let $\mathcal{D} = \{(q_i^t, y_i)\}_{i=1}^N$ be a set of text-based reasoning questions, where q_i^t is the text input and y_i is the corresponding verifiable answer. We define a text-to-image transformation $\mathcal{T}(q_i^t) \rightarrow q_i^v$ that renders the text problem q_i^t into a semantically equivalent image q_i^v , preserving semantics while changing only the input modality. In practice, this could be done by straightforward processing e.g., rendering the text in black font on a white background. For a given MLLM \mathcal{M} , we can evaluate its performance across three distinct input configurations, each combined with reasoning instruction r :

- Text input (x_i^t): The original text question $x_i^t = \text{concat}(r, q_i^t)$.
- Visual input (x_i^v): The rendered image question $x_i^v = \text{concat}(r, q_i^v)$.
- Extracted text input (\tilde{x}_i^t): A perception control where the model first extracts text \tilde{q}_i^t from the image q_i^v given an OCR task instruction, and then reasons over this extracted text $\tilde{x}_i^t = \text{concat}(r, \tilde{q}_i^t)$.

We evaluate the overall accuracy of responses generated by \mathcal{M} against the ground-truth answer y_i for each input type, denoted as acc^t , acc^v , and acc^{ext} , respectively. To further assess transcription fidelity, we compute OCR accuracy acc^{OCR} by comparing the extracted text \tilde{q}_i^t with the original text question q_i^t .

Perception-control Gap. The extracted-text condition serves as a control for perception. If the model can faithfully

read the rendered problem and reason over the recovered text, then performance on extracted text should match performance on the original text: $\text{acc}^t \approx \text{acc}^{ext}$. We define the perception-control gap as $\Delta_{\text{perc}} = \text{acc}^t - \text{acc}^{ext}$. This quantity measures the downstream reasoning loss induced by replacing the original text with text extracted from the image. A large Δ_{perc} may arise from OCR failures, formatting changes, or other perception-related degradations that affect subsequent reasoning. Conversely, high acc^{OCR} and small Δ_{perc} indicate that the model can recover a reasoning-sufficient textual representation of the visual input.

Reasoning-access Gap. We define the reasoning-access gap as the residual loss incurred when the model reasons directly from the image rather than from the corresponding text problem:

$$\text{acc}^t - \text{acc}^v = (\text{acc}^t - \text{acc}^{ext}) + (\text{acc}^{ext} - \text{acc}^v) = \Delta_{\text{perc}} + (\text{acc}^{ext} - \text{acc}^v) \quad (1)$$

Therefore, a reasoning-access gap is characterized by $\text{acc}^{ext} \approx \text{acc}^t > \text{acc}^v$, especially when acc^{OCR} is high. This pattern indicates that the model possesses the relevant reasoning ability, as evidenced by high text-input accuracy, and can recover the problem content from the image, as evidenced by the extracted-text control. Nevertheless, the same reasoning ability is not effectively engaged when the semantically equivalent input is processed directly under the visual inputs. In MLLMs, different modalities traverse distinct processing pathways: text inputs are handled directly by the language backbone, whereas image inputs are first processed by a vision encoder and projection module. While MLLM training might have achieved *semantic alignment* between vision and text embeddings, it might not have aligned the vision embeddings such that they also activate reasoning capabilities that would have been engaged for semantically equivalent text embeddings.

Empirical Observations. We evaluate the reasoning performance gaps for the various modality input types across multiple model families. Using MATH-500 (Lightman et al., 2024) as our text-based dataset \mathcal{D} , and rendering the text questions as black font on white background as our transformation \mathcal{T} to obtain our image-based dataset (MATH-500-I), we compute acc^t , acc^v , and acc^{ext} , respectively. All experiments are repeated three times to account for stochasticity, and we report mean accuracy along with standard deviations. Full experimental details (e.g., prompts, hyperparameters) are in App. A. Tab. 2 reveals a consistent *modality reasoning-*

Table 1. Contingency table between OCR correctness and answer correctness for Qwen3-VL-8B on MATH-500-I. The final column demonstrates that among all incorrect answers, the vast majority (92.36%) occur despite correct visual extraction.

	Ans. Correct	Ans. Wrong	$p(\text{Ans. Wrong} \text{OCR})$
OCR Correct	351	133	92.36%
OCR Wrong	5	11	7.64%

Table 2. Performance on semantically equivalent text and image versions of MATH-500. Scores are colored relative to each model’s text baseline (purple): compared to the text-level performance, red indicates degradation, while green indicates an increase. Despite identical semantics, all models consistently perform worse on image inputs, revealing a robust modality reasoning-access gap. This gap persists across both rendering schemes (MATH-500-I and MATH-500-SI), indicating that it is not due to a specific transformation. OCR results remain high, suggesting that the bottleneck lies in reasoning from image-conditioned representations rather than reading ability.

Dataset	Input	InternVL3.5-1B	Qwen2.5-VL-3B	Qwen2.5-VL-7B	Qwen3-VL-8B
MATH-500	Text	45.07 ± 0.12	50.87 ± 0.12	63.80 ± 1.06	75.53 ± 0.42
MATH-500-I	OCR	90.00 ± 0.35	92.93 ± 0.31	97.47 ± 0.23	96.73 ± 0.31
	Extracted text	44.73 ± 0.64	52.93 ± 0.50	64.80 ± 0.87	76.40 ± 0.53
	Image	37.27 ± 0.31	39.93 ± 0.64	58.27 ± 0.31	70.87 ± 0.64
MATH-500-SI	OCR	84.83 ± 0.15	87.33 ± 0.31	96.27 ± 0.31	92.47 ± 0.31
	Extracted text	43.40 ± 0.53	47.93 ± 0.81	63.67 ± 0.31	75.53 ± 0.31
	Square image	32.47 ± 1.33	35.07 ± 1.10	55.93 ± 0.31	68.60 ± 0.20

access gap: across all models, accuracy drops when MATH-500 problems are presented as images rather than as text, despite being semantically identical. On MATH-500-I, image inputs reduce accuracy by 7.2 points on average, with the largest drop for Qwen2.5-VL-3B. To explicitly assess whether perception problems may have caused the modality performance gap, we also perform contingency analysis in Tab. 1 showing that most incorrect image-based predictions

occur even when the problem text is extracted correctly, accounting for 92.26% among incorrect answers (see Sec. B.2 for consistent results for other models).

To assess the robustness of this observed reasoning-access gap, we also perform ablations across various experiment settings. For example, we apply a different image rendering transformation that enforces a square aspect ratio (MATH-500-SI) and find the gap to be even larger, indicating that the effect is not specific to a particular rendering style. Further ablations, such as those on different temperature settings, prompting and transformations, and checks ruling out potential confounders such as generation lengths, are in App. B, indicating the persistence of this gap.

Table 3. Performance of base and post-trained MLLMs on synthesized, mathematical, and general benchmarks. On the synthesized benchmarks, subscripts indicate the difference between each model’s image performance and the text performance of the corresponding base model, directly quantifying the remaining text–image gap after post-training. For real-world datasets, we report differences relative to the base model. **Blue (red)** numbers in parentheses represent performance **gains (drops)** relative to the baseline. The best performance is in **boldface**, and the second best is underlined.

Model	Synthesized		Math			General	Avg.
	MATH-500-I	BBH-I	MathVista	MathVerse	MathVision	EMMA	
InternVL3.5-1B	37.3 _(-7.8)	31.6 _(-6.6)	48.6	<u>25.0</u>	15.8	8.1	27.7
w/ SFT	<u>43.9</u> _(-1.2)	32.3 _(+2.8)	<u>50.4</u> _(+1.8)	23.1 _(-1.9)	16.1 _(+0.3)	7.4 _(-0.7)	28.9 _(+1.2)
w/ DPO	13.9 _(-31.2)	16.9 _(-21.3)	46.9 _(-1.7)	24.4 _(-0.6)	16.2 _(+0.4)	7.8 _(-0.3)	21.0 _(-6.7)
w/ GRPO _{text}	40.2 _(-4.9)	<u>33.5</u> _(-4.7)	50.3 _(+1.7)	24.8 _(-0.2)	<u>17.3</u> _(+1.5)	<u>8.5</u> _(+0.4)	<u>29.1</u> _(+1.4)
w/ GRPO	49.9 _(+4.8)	37.8 _(-0.4)	51.2 _(+2.6)	27.9 _(+2.9)	21.2 _(+5.4)	8.9 _(+0.8)	32.8 _(+5.1)
Qwen2.5-VL-3B	39.9 _(-11.0)	37.7 _(-9.6)	61.8	33.9	21.2	20.5	35.8
w/ SFT	52.8 _(+1.9)	46.5 _(-0.8)	63.0 _(+1.2)	<u>37.1</u> _(+3.2)	24.6 _(+3.4)	20.8 _(+0.3)	<u>40.8</u> _(+5.0)
w/ DPO	<u>54.0</u> _(+3.1)	<u>47.7</u> _(+0.4)	63.1 _(+1.3)	34.8 _(+0.9)	22.4 _(+1.2)	<u>21.3</u> _(+0.8)	40.6 _(+4.8)
w/ GRPO _{text}	45.9 _(-5.0)	43.8 _(-3.5)	<u>63.5</u> _(+1.7)	37.0 _(+3.1)	<u>25.2</u> _(+4.0)	<u>21.3</u> _(+0.8)	39.5 _(+3.7)
w/ GRPO	58.8 _(+7.9)	49.5 _(+2.2)	64.1 _(+2.3)	38.8 _(+4.9)	25.8 _(+4.6)	21.8 _(+1.3)	43.1 _(+7.3)
Qwen2.5-VL-7B	58.3 _(-5.5)	47.1 _(-10.4)	68.9	43.9	25.1	22.2	44.3
w/ SFT	59.6 _(-4.2)	46.6 _(-10.9)	69.1 _(+0.2)	45.7 _(+1.8)	26.8 _(+1.7)	22.9 _(+0.7)	45.1 _(+0.8)
w/ DPO	<u>62.0</u> _(-1.8)	48.6 _(-8.9)	<u>70.4</u> _(+1.5)	44.6 _(+0.7)	26.7 _(+1.6)	23.0 _(+0.8)	45.9 _(+1.6)
w/ GRPO _{text}	58.5 _(-5.3)	<u>56.2</u> _(-1.3)	68.5 _(-0.4)	<u>46.3</u> _(+2.4)	<u>27.7</u> _(+2.6)	<u>24.5</u> _(+2.3)	<u>47.0</u> _(+2.7)
w/ GRPO	64.8 _(+1.0)	56.8 _(-0.7)	70.6 _(+1.7)	47.5 _(+3.6)	28.1 _(+3.0)	25.7 _(+3.5)	48.9 _(+4.6)
Qwen3-VL-8B	70.9 _(-4.6)	<u>77.1</u> _(-1.4)	75.7	54.5	51.2	30.6	60.0
w/ SFT	77.1 _(+1.6)	75.8 _(-2.7)	75.9 _(+0.2)	<u>60.8</u> _(+6.3)	48.5 _(-2.7)	24.1 _(-6.5)	60.4 _(+0.4)
w/ DPO	76.9 _(+1.4)	75.4 _(-3.1)	76.6 _(+0.9)	58.0 _(+3.5)	50.2 _(-1.0)	27.8 _(-2.8)	60.8 _(+0.8)
w/ GRPO _{text}	<u>79.3</u> _(+3.8)	76.6 _(-1.9)	<u>77.1</u> _(+1.4)	57.5 _(+3.0)	<u>51.8</u> _(+0.6)	<u>34.8</u> _(+4.2)	<u>62.9</u> _(+2.9)
w/ GRPO	82.4 _(+6.9)	79.1 _(+0.6)	78.7 _(+3.0)	65.8 _(+11.3)	55.3 _(+4.1)	36.3 _(+5.7)	66.3 _(+6.3)

4. Text-to-Image Gap-targeted Training for Enhanced Reasoning

The analysis in Section 3 suggests that even when the visual content can be accurately transcribed, image-conditioned inputs may fail to engage the reasoning behavior that the same model already exhibits under text conditioning. To address this, we introduce a **TIGER: gap-targeted training**, including gap-targeted data construction (Sec. 4.1) and gap-targeted training (Sec. 4.2), shown in Figure 1, that directly optimizes image-conditioned reasoning on examples where this failure occurs.

4.1. Gap-targeted Data Construction

We construct training data from text-only problems with verifiable answers, avoiding the need for manually curated image–text reasoning pairs while still enabling semantically equivalent visual inputs through rendering. We use OpenR1-Math-220 (Hugging Face, 2025), a collection of text math problems, and retain only examples whose generated

solutions are verified as correct by Math-Verify,¹ yielding roughly 45K solvable problems. For each problem i , we construct a text-conditioned input x_i^t and a corresponding image-conditioned input x_i^v . We query the base model with both inputs using greedy decoding, obtaining text- and image-conditioned responses a_i^t and a_i^v , respectively. Each response is evaluated against the ground-truth answer y_i . Let $c_i^t, c_i^v \in \{0, 1\}$ denote the correctness of the text- and image-conditioned responses. This partitions the data into four subsets:

$$\mathcal{D}_{ab} = \{(x_i^v, a_i^t, a_i^v, y_i) : c_i^t = a, c_i^v = b\}, \quad a, b \in \{0, 1\}.$$

We focus on \mathcal{D}_{10} , consisting of instances where the model succeeds on the text input but fails on the corresponding image input. These cases directly isolate the modality gap: the model has sufficient knowledge and reasoning ability, yet cannot reliably apply them from visual inputs. In contrast, \mathcal{D}_{11} offers little learning signal since both modalities are already solved, while \mathcal{D}_{00} reflects failures unrelated to modality differences. The set \mathcal{D}_{01} (image success but text failure) is rare and does not align with the dominant degradation pattern observed in Section 3. We therefore retain only \mathcal{D}_{10} and denote it by \mathcal{D} . For each example in \mathcal{D} , we define $d_i = (x_i^v, a_i^+, a_i^-, y_i)$, where $a_i^+ := a_i^t$ is the correct response obtained from the text input, and $a_i^- := a_i^v$ is the incorrect response obtained from the image input. Since x_i^t and x_i^v are semantically equivalent, a_i^+ serves as valid supervision for the visual task, while a_i^- captures the model’s failure under image conditioning.

4.2. Gap-targeted Training

We use Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as the primary post-training objective. Unlike imitation learning, GRPO does not require the model to match a fixed reasoning trajectory. Instead, it samples multiple responses for each prompt and reinforces responses with higher rewards. This is well suited to our setting: our goal is not to reproduce the text-conditioned solution verbatim, but to improve the model’s ability to reason correctly from visual inputs. Let π_θ denote the MLLM policy parameterized by θ . For each image prompt $x_i^v \in \mathcal{D}$, we sample a group of K responses: $\{o_i^{(k)}\}_{k=1}^K \sim \pi_\theta(\cdot | x_i^v)$. Motivated by the analysis in Section 3, we do not use an explicit perception reward. Since the rendered images have high OCR recoverability and feeding the extracted text back into the model largely restores text-level performance, we focus the reward on the downstream reasoning outcome. The reward contains two terms: (1) **Answer Correctness**, $r_a(o_i^{(k)}, y_i)$, directly measuring whether the sampled response reaches the correct final answer y_i , and (2) **Format Correctness**, $r_f(o_i^{(k)})$, encouraging generated responses to follow the required answer format. The final reward is

$$r(o_i^{(k)}, y_i) = \alpha \cdot r_a(o_i^{(k)}, y_i) + \beta \cdot r_f(o_i^{(k)}), \quad (2)$$

where α and β control the relative weights of answer correctness and formatting. Then GRPO normalizes rewards within each sampled group and updates the policy toward higher-reward responses. Because all rollouts are conditioned on x_i^v , the optimization directly targets the image-conditioned reasoning process. Further details on training losses can be found in Sec. D.2.

4.3. Comparison Objectives

To assess whether simpler post-training objectives are sufficient for reducing the modality gap, we also evaluate SFT and DPO on the same gap-targeted dataset \mathcal{D} . These objectives use the same data budget as GRPO but differ in how they use the mined responses.

Supervised Fine-Tuning (SFT). For SFT, we pair each image input with the correct text-conditioned response: $\mathcal{D}_{\text{SFT}} = \{(x_i^v, a_i^+)\}_{i=1}^{|\mathcal{D}|}$. This objective trains the model to produce, from the image input, a response that is correct under text conditioning.

Direct Preference Optimization (DPO). For DPO (Rafailov et al., 2023), each example in \mathcal{D} naturally defines a preference pair: $\mathcal{D}_{\text{DPO}} = \{(x_i^v, a_i^+, a_i^-)\}_{i=1}^{|\mathcal{D}|}$. Here, a_i^+ is the preferred response because it is correct for the underlying problem, while a_i^- is rejected because it is the model’s incorrect image-conditioned output. This objective directly encourages the model to prefer correct reasoning over its previous visual failure.

¹<https://github.com/huggingface/Math-Verify>

5. Experiments

5.1. Experimental Setup

Implementation Details. We apply the procedure in Sec. 4.1 to construct a model-specific training set \mathcal{D} for each backbone, including InternVL3.5-1B, Qwen2.5-VL-3B, Qwen2.5-VL-7B, and Qwen3-VL-8B. For a fair comparison across post-training methods, we fine-tune all models end-to-end with a fixed batch size of 128. For Qwen2.5-VL and Qwen3-VL models, we use the standard fine-tuning repository of Lee (2024) for SFT and DPO, and EasyR1 (Yaowei Zheng, 2025) for GRPO. For InternVL3.5-1B, we adopt the official InternVL implementation for SFT and DPO², and adapt EasyR1 (Yaowei Zheng, 2025) for efficient GRPO training. More details can be found at App. D.

Evaluation Benchmarks. Because all fine-tuning settings are based on OpenR1-Math-220 (Hugging Face, 2025), successfully reducing the modality gap should produce improvements that generalize beyond the training set, including both newly synthesized benchmarks and real multimodal reasoning tasks. To evaluate transfer on synthesized data, in addition to MATH-500-I in Sec. 3, we consider a more challenging benchmark derived from BBH (Suzgun et al., 2023), which covers a diverse set of reasoning skills, including temporal, spatial understanding, and linguistic reasoning. We apply the same transformation \mathcal{T} to construct its image-based counterpart, denoted BBH-I. For real-data evaluation, we consider standard multimodal mathematical reasoning benchmarks, including MathVista (Lu et al., 2024), MathVerse (Zhang et al., 2024a), and MathVision (Wang et al., 2024b). In addition, since the training data is exclusively mathematical, we evaluate on EMMA (Hao et al., 2025), a broader benchmark spanning domains such as chemistry, coding, and physics, to test whether the resulting gains transfer beyond mathematics. For synthesized benchmarks, we run each experiment three times to account for possible variance and report average accuracy; evaluation details are provided in App. A. For real-world evaluation, we adopt VLMEvalKit (Duan et al., 2024), a widely used toolkit for evaluating MLLMs on diverse benchmarks, to ensure a consistent evaluation protocol.

Table 4. Comparison with representative RLVR methods based on Qwen2.5-VL-7B. Prior approaches add components such as cross-modal formalization, multi-stage self-rewarding, or perception-aware rewards. In contrast, our method directly applies GRPO to gap-targeted, image-conditioned data. For synthesized datasets, subscripts denote performance differences relative to the corresponding text-input base model; for other datasets, differences are reported relative to Qwen2.5-VL-7B.

Model	MATH-500-I	BBH-I	MathVista	MathVerse	MathVision	EMMA	Avg.
Qwen2.5-VL-7B	58.3 _(-5.5)	47.1 _(-10.4)	68.9	43.9	25.1	22.2	44.3
w/ our training	64.8 _(+1.0)	56.8 _(-0.7)	70.6 _(+1.7)	47.4 _(+3.5)	28.1 _(+3.0)	25.7 _(+3.5)	48.9 _(+4.6)
R1-OneVision	50.5 _(-4.8)	47.7 _(-4.6)	62.9 _(-6.0)	45.1 _(+1.2)	23.3 _(-1.8)	22.1 _(-0.1)	41.9 _(-2.4)
w/ our training	65.1 _(+9.8)	51.7 _(-0.6)	67.4 _(-2.0)	48.6 _(+3.8)	28.3 _(+3.2)	24.3 _(+1.5)	47.6 _(+3.3)
Vision-SR1	34.1 _(-3.9)	51.9 _(-0.9)	69.4 _(+0.5)	44.1 _(+0.2)	26.5 _(+1.4)	24.1 _(+1.9)	41.7 _(-2.6)
w/ our training	52.9 _(+14.9)	55.4 _(+2.6)	69.9 _(+1.0)	44.5 _(+0.6)	26.7 _(+1.6)	25.1 _(+2.9)	45.8 _(+1.5)
Perception-R1	60.3 _(-4.6)	57.6 _(-2.2)	70.4 _(+1.5)	45.5 _(+1.2)	28.7 _(+3.6)	25.3 _(+3.1)	48.0 _(+3.7)
w/ our training	65.6 _(+0.7)	59.2 _(-0.6)	70.8 _(+1.9)	47.9 _(+3.6)	29.4 _(+4.3)	25.2 _(+3.0)	49.7 _(+5.4)

5.2. Main Results

Comparison with the base model. We first evaluate whether post-training can reduce the modality-dependent reasoning gap identified in Sec. 3. As shown in Tab. 3, this gap is not limited to MATH-500-I but also clearly appears on BBH-I, indicating its persistence across benchmarks. Among all post-training objectives, GRPO yields the strongest and most consistent improvements. It achieves the largest gains on the synthesized benchmarks and substantially narrows, and in some cases eliminates, the gap to the text-only performance of the base model. These gains also transfer to held-out multimodal benchmarks, where GRPO consistently performs best, with notable improvements on MathVerse and EMMA. For instance, Qwen2.5-VL-7B improves from 43.9 to 47.5 on MathVerse and from 22.2 to 25.7 on EMMA, while Qwen3-VL-8B improves from 54.5 to 65.8 on MathVerse, corresponding to gains of up to $\sim 15\%$ over the base model. In contrast, although SFT and DPO can improve performance on synthesized settings, they do not consistently generalize to real-world multimodal benchmarks. For example, on Qwen3-VL-8B, both methods improve MathVerse

²<https://github.com/OpenGVLab/InternVL>

but degrade performance on MathVision and EMMA. Overall, these results suggest that GRPO is the most effective objective for reducing the in-domain modality gap while maintaining strong generalization to realistic multimodal reasoning tasks.

Complementarity to RLVR approaches. We further compare with representative RLVR methods built on the same Qwen2.5-VL-7B, using the official models released by the respective authors for evaluation. Existing RLVR pipelines often introduce complex components, such as reasoning trace formalization (Yang et al., 2025b), multi-stage self-rewarding (Li et al., 2026), or perception-specific rewards (Xiao et al., 2026). In contrast, our approach directly applies GRPO to gap-targeted image-conditioned data. As Tab. 4 illustrates, prior RLVR methods (Xiao et al., 2026; Li et al., 2026; Yang et al., 2025b) still suffer from substantial modality-dependent degradation, particularly on synthesized image-conditioned benchmarks. Integrating our training paradigm consistently improves all evaluated RLVR checkpoints (+3.4 on average), demonstrating its broad complementarity. Notably, our standalone model already surpasses all prior RLVR baselines on average, while combining our method with Perception-R1 (Xiao et al., 2026) yields the highest overall performance. These results indicate that explicitly optimizing image-conditioned reasoning provides a simple yet effective strategy to narrow the modality gap and strengthen more elaborate RLVR frameworks.

5.3. Analysis

We next analyze why TIGER improves image-conditioned reasoning. Our analysis is organized around three questions: (i) whether optimization must be performed under visual conditioning, (ii) whether TIGER increases the activation of text-derived reasoning-relevant directions, and (iii) whether the gap-targeted construction of \mathcal{D} is necessary.

Is image-conditioning necessary for TIGER’s GRPO? In TIGER, GRPO might contribute to better reasoning performance due to two effects: improved overall reasoning capabilities, and the closing of the text-image reasoning-access gap. To analyze this, we evaluate $\text{GRPO}_{\text{text}}$, a variant trained on the identical problems but conditioned on text rather than rendered images. This controls for additional exposure to the same reasoning task training and RL objective, isolating the conditional modality during optimization. Tab. 3 shows that GRPO consistently outperforms $\text{GRPO}_{\text{text}}$ (+3.1 points on average). On synthesized benchmarks (MATH-500-I, BBH-I), GRPO substantially reduces or eliminates the residual gap to the base model’s text performance. Crucially, these gains successfully transfer to real multimodal benchmarks (MathVista, MathVerse, MathVision, EMMA). These results suggest how gains depend on optimizing policy conditioned on visual input that adapt the visual-to-reasoning pathway, and supporting TIGER’s premise that the model needs to learn how visual input can engage existing reasoning capabilities. We further analyze OCR versus answer correctness after training. For our GRPO-finetuned Qwen3-VL-8B on MATH-500-I (Tab. 5), training significantly decreases incorrect answers given correct problem extraction, and performance gains are derived not from increased OCR/perception but better reasoning over visual content. Qualitative case studies on real-world benchmarks are in App. E.

Does TIGER increase reasoning-subspace activation?

Next, we test whether TIGER changes internal representations consistent with improved reasoning access. Our hypothesis is that text-side reasoning depends on a subspace of latent directions in the language backbone, and that image-conditioned reasoning failures can occur when visual inputs under-activate these directions even when their content is correctly perceived. To estimate this subspace, we follow prior work on reasoning directions (Chandra et al., 2026) and construct contrastive prompt pairs for text inputs. For each text problem q_i^t , we form a reasoning-oriented prompt $s_i^c = [r^c; q_i^t]$ using a Chain-of-Thought (CoT) instruction r^c and a non-CoT prompt $s_i^n = [r^n; q_i^t]$ with a non-CoT instruction r^n . From a set of m such pairs, we extract the hidden states of the final sequence token at each layer l , denoted

Table 5. Contingency table between OCR correctness and answer correctness for our GRPO-finetuned Qwen3-VL-8B on MATH-500-I. Subscripts report differences relative to the base model.

	Ans. Correct	Ans. Wrong
OCR Correct	407(+56)	77(-56)
OCR Wrong	7(+2)	9(-2)

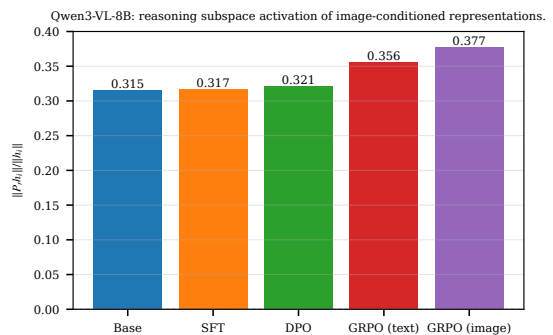


Figure 2. Reasoning subspace activation of image-conditioned representations.

$\mathbf{h}_i^{c,(l)}, \mathbf{h}_i^{n,(l)} \in \mathbb{R}^d$. Then we concatenate these states across all L layers to form representations $\mathbf{h}_i^c, \mathbf{h}_i^n \in \mathbb{R}^{L \times d}$. We define the reasoning subspace \mathcal{R} as the span of the top- k principal components of the contrastive differences $\{\mathbf{h}_i^c - \mathbf{h}_i^n\}_{i=1}^m$. Note that this procedure does not identify all reasoning mechanisms causally, but rather provides an empirical subspace associated with text-side reasoning.

For an image input $u_i = [r^c; q_i^v]$, let \mathbf{h}_i^u be the concatenated representation, and $\mathbf{P}_{\mathcal{R}}$ be the orthogonal projection onto \mathcal{R} . We define Reasoning-Subspace Activation as $\text{RSA}(\mathbf{h}_i^u) := \|\mathbf{P}_{\mathcal{R}}\mathbf{h}_i^u\|_2 / \|\mathbf{h}_i^u\|_2$, which measures the fraction of an image-conditioned representation that lies in the text-derived latent reasoning subspace. As a validity check, we assess empirically whether RSA is predictive of visual input correctness at the question level by computing point-biserial correlation between RSA and answer correctness (Tab. 18, Sec. F.1) and find consistent positive correlation ($r \approx 0.37 - 0.43$). Fig. 2 shows that SFT and DPO only provide marginal improvements in RSA over the base model. $\text{GRPO}_{\text{text}}$ increases RSA more than SFT and DPO, but GRPO increases RSA most substantially. This mirrors the performance trends in Tab. 3 (i.e., Base, SFT, DPO, $\text{GRPO}_{\text{text}}$, GRPO). The result supports the interpretation that TIGER increases visual activation of latent directions associated with successful text-side reasoning.

Visual representation contributions. To further examine how post-training improves visual reasoning access, we conduct activation patching (Zhang & Nanda, 2024) by replacing visual representations from post-trained models into the fixed base model before decoding. If patched activations improve accuracy, this suggests that our trained visual representations carry useful information to engage downstream reasoning. As shown in Tab. 6 (see Sec. F.1 for other models), patching with image-conditioned GRPO leads to the largest improvement over other variants, suggesting that it more effectively aligns visual representations with the model’s existing reasoning mechanisms. This provides intervention-based evidence that TIGER changes the intermediate visual representations useful for reasoning, complementing the RSA analysis.

Table 6. Activation patching results across methods.

MATH-500-I	
Qwen3-VL-8B	71.6
w/ SFT	73.8 _(+2.2)
w/ DPO	75.2 _(+3.6)
w/ GRPO	76.2 _(+4.6)

The necessity of constructing \mathcal{D} for GRPO. While Tab. 3 demonstrates that fine-tuning with GRPO on gap-targeted training data \mathcal{D} yields consistent improvements, it raises the question of whether comparable performance could be achieved by simply sampling an equivalently sized subset from OpenR1-Math-220 (Hugging Face, 2025). To investigate this, we randomly sample three distinct, $|\mathcal{D}|$ -sized subsets that are strictly disjoint from \mathcal{D} and report the averaged results to account for variance. As detailed in Tab. 7, although GRPO fine-tuning generally enhances the base model, training on these random subsets yields substantially lower performance (-2.1) than training on \mathcal{D} . This highlights the critical importance of our targeted data construction, which specifically isolates instances where the model possesses the requisite reasoning capabilities but struggles to apply them to visual inputs.

Table 7. Ablation on the importance of constructing the gap-targeted training set \mathcal{D} for GRPO fine-tuning. Randomly sampled non-overlapping subsets provide some gains, but consistently underperform training on \mathcal{D} .

Model	MATH-500-I	BBH-I	MathVista	MathVerse	MathVision	EMMA	Avg.
InternVL3.5-1B	37.3 _(-7.8)	31.6 _(-6.6)	48.6	25.0	15.8	8.1	27.7
w/ non-overlapping	44.0 _(-1.1)	37.1 _(-1.1)	49.1 _(+0.5)	27.0 _(+2.0)	18.4 _(+2.6)	8.5 _(+0.4)	30.7 _(+3.0)
w/ \mathcal{D}	49.9 _(+4.8)	37.8 _(-0.4)	51.2 _(+2.6)	27.9 _(+2.9)	21.2 _(+5.4)	8.9 _(+0.8)	32.8 _(+5.1)

6. Conclusion

We identify a modality-dependent reasoning-access gap in MLLMs, where models fail to route correctly perceived visual evidence into its existing reasoning capabilities. We characterize this gap using analysis based on modality counterfactuals, and propose TIGER, a simple framework that automatically mines *text-success*, *image-failure* examples from text-only corpora without requiring manually curated image-text pairs, before training via image-conditioned GRPO. Despite not using multimodal training data, TIGER achieves consistent reasoning performance gains across multimodal benchmarks. We find that TIGER is also complementary to RLVR methods and can yield additional gains. These results suggest that improving multimodal reasoning requires not only better visual perception, but also explicit visual access to models’ internal reasoning capabilities.

References

- Amizadeh, S., Palangi, H., Polozov, A., Huang, Y., and Koishida, K. Neuro-symbolic visual reasoning: Disentangling "Visual" from "Reasoning". In III, H. D. and Singh, A. (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 279–290. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/amizadeh20a.html>.
- Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., and Zhou, J. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond, 2023. URL <https://arxiv.org/abs/2308.12966>.
- Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., and Zhu, K. Qwen3-vl technical report, 2025a. URL <https://arxiv.org/abs/2511.21631>.
- Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., and Lin, J. Qwen2.5-vl technical report, 2025b. URL <https://arxiv.org/abs/2502.13923>.
- Chandra, A., Agrawal, A., Hosseini, A., Fischmeister, S., Agarwal, R., Goyal, N., and Courville, A. Shape of thought: When distribution matters more than correctness in reasoning tasks, 2026. URL <https://arxiv.org/abs/2512.22255>.
- Chen, Y., Sikka, K., Cogswell, M., Ji, H., and Divakaran, A. Measuring and improving chain-of-thought reasoning in vision-language models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 192–210, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.11. URL <https://aclanthology.org/2024.naacl-long.11/>.
- Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., Li, B., Luo, P., Lu, T., Qiao, Y., and Dai, J. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks, 2024b. URL <https://arxiv.org/abs/2312.14238>.
- Clark, P., Cowhey, I., Etzioni, O., Khot, T., Sabharwal, A., Schoenick, C., and Tafjord, O. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018. URL <https://arxiv.org/abs/1803.05457>.
- Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., Dong, X., Zang, Y., Zhang, P., Wang, J., et al. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 11198–11201, 2024.
- Fu, C., Dai, Y., Luo, Y., Li, L., Ren, S., Zhang, R., Wang, Z., Zhou, C., Shen, Y., Zhang, M., Chen, P., Li, Y., Lin, S., Zhao, S., Li, K., Xu, T., Zheng, X., Chen, E., Shan, C., He, R., and Sun, X. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24108–24118, 2025. doi: 10.1109/CVPR52734.2025.02245.
- Ge, J., Luo, H., Qian, S., Gan, Y., Fu, J., and Zhang, S. Chain of thought prompt tuning in vision language models, 2023. URL <https://arxiv.org/abs/2304.07919>.
- Guo, D., Yang, D., Zhang, H., Song, J., Wang, P., Zhu, Q., Xu, R., Zhang, R., Ma, S., Bi, X., et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

- Hao, Y., Gu, J., Wang, H. W., Li, L., Yang, Z., Wang, L., and Cheng, Y. Can MLLMs reason in multimodality? EMMA: An enhanced multimodal reasoning benchmark. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=v26vwjx0Ez>.
- Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding, 2021a. URL <https://arxiv.org/abs/2009.03300>.
- Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021b. URL <https://openreview.net/forum?id=7Bywt2mQsCe>.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025. URL <https://github.com/huggingface/open-r1>.
- Kumar, K., Ashraf, T., Thawakar, O., Anwer, R. M., Cholakkal, H., Shah, M., Yang, M.-H., Torr, P. H. S., Khan, F. S., and Khan, S. Llm post-training: A deep dive into reasoning large language models, 2025. URL <https://arxiv.org/abs/2502.21321>.
- Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP '23*, pp. 611–626, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400702297. doi: 10.1145/3600006.3613165. URL <https://doi.org/10.1145/3600006.3613165>.
- Lee, Y. Qwen2-vl-finetune, 2024. URL <https://github.com/2U1/Qwen2-VL-Finetune>.
- Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Liu, Y., Wang, Z., Xu, J., Chen, G., Lou, P., Wang, L., and Qiao, Y. Mvbench: A comprehensive multi-modal video understanding benchmark. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2024. doi: 10.1109/CVPR52733.2024.02095.
- Li, L., Xie, Z., Li, M., Chen, S., Wang, P., Chen, L., Yang, Y., Wang, B., and Kong, L. Silkie: Preference distillation for large visual language models, 2023. URL <https://arxiv.org/abs/2312.10665>.
- Li, Z., Yu, W., Huang, C., Liang, Z., Liu, R., Liu, F., Chen, J., Yu, D., Boyd-Graber, J. L., Mi, H., and Yu, D. Vision-SR1: Self-rewarding vision-language model via reasoning decomposition and multi-reward policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=C1M4ETatgM>.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=v8L0pN6EOi>.
- Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=w0H2xGH1kw>.
- Liu, J., Zeng, W., Zhang, X., Wang, Y., Shan, Z., and He, J. On the perception bottleneck of vlms for chart understanding, 2025. URL <https://arxiv.org/abs/2503.18435>.
- Loshchilov, I. and Hutter, F. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL <https://openreview.net/forum?id=Bkg6RiCqY7>.
- Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.-W., Galley, M., and Gao, J. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.
- Luo, T., Cao, A., Lee, G., Johnson, J., and Lee, H. Probing visual language priors in VLMs. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=bhTBirS0qi>.

- Mangalam, K., Akshulakov, R., and Malik, J. Egoschema: A diagnostic benchmark for very long-form video language understanding. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL <https://openreview.net/forum?id=JVlWseddak>.
- Masry, A., Long, D. X., Tan, J. Q., Joty, S., and Hoque, E. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.177. URL <https://aclanthology.org/2022.findings-acl.177/>.
- Masry, A., Islam, M. S., Ahmed, M., Bajaj, A., Kabir, F., Kartha, A., Laskar, M. T. R., Rahman, M., Rahman, S., Shahmohammadi, M., Thakkar, M., Parvez, M. R., Hoque, E., and Joty, S. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19123–19151, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.978. URL <https://aclanthology.org/2025.findings-acl.978/>.
- OpenAI. Introducing OpenAI o1. <https://openai.com/o1/>, September 2024. Accessed: 2026-04-11.
- Pi, R., Han, T., Xiong, W., Zhang, J., Liu, R., Pan, R., and Zhang, T. Strengthening multimodal large language model with bootstrapped preference optimization. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXXIII*, pp. 382–398, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73413-7. doi: 10.1007/978-3-031-73414-4_22. URL https://doi.org/10.1007/978-3-031-73414-4_22.
- Qiao, R., Tan, Q., Dong, G., Wu, M., Sun, C., Song, X., GongQue, Z., Lei, S., Wei, Z., Zhang, M., Qiao, R., Zhang, Y., Zong, X., Xu, Y., Diao, M., Bao, Z., Li, C., and Zhang, H. We-math: Does your large multimodal model achieve human-like mathematical reasoning?, 2024. URL <https://arxiv.org/abs/2407.01284>.
- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL <https://openreview.net/forum?id=HPuSIXJaa9>.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Suzgun, M., Scales, N., Schärli, N., Gehrmann, S., Tay, Y., Chung, H. W., Chowdhery, A., Le, Q., Chi, E., Zhou, D., and Wei, J. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Rogers, A., Boyd-Graber, J., and Okazaki, N. (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL <https://aclanthology.org/2023.findings-acl.824/>.
- Tong, S., Liu, Z., Zhai, Y., Ma, Y., LeCun, Y., and Xie, S. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9568–9578, June 2024.
- Wang, F., Zhou, W., Huang, J. Y., Xu, N., Zhang, S., Poon, H., and Chen, M. mDPO: Conditional preference optimization for multimodal large language models. In Al-Onaizan, Y., Bansal, M., and Chen, Y.-N. (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8078–8088, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.460. URL <https://aclanthology.org/2024.emnlp-main.460/>.
- Wang, K., Pan, J., Shi, W., Lu, Z., Ren, H., Zhou, A., Zhan, M., and Li, H. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL <https://openreview.net/forum?id=QWTCcxMpPA>.

- Wang, P., Bai, S., Tan, S., Wang, S., Fan, Z., Bai, J., Chen, K., Liu, X., Wang, J., Ge, W., Fan, Y., Dang, K., Du, M., Ren, X., Men, R., Liu, D., Zhou, C., Zhou, J., and Lin, J. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution, 2024c. URL <https://arxiv.org/abs/2409.12191>.
- Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., Wang, Z., Chen, Z., Zhang, H., Yang, G., Wang, H., Wei, Q., Yin, J., Li, W., Cui, E., Chen, G., Ding, Z., Tian, C., Wu, Z., Xie, J., Li, Z., Yang, B., Duan, Y., Wang, X., Hou, Z., Hao, H., Zhang, T., Li, S., Zhao, X., Duan, H., Deng, N., Fu, B., He, Y., Wang, Y., He, C., Shi, B., He, J., Xiong, Y., Lv, H., Wu, L., Shao, W., Zhang, K., Deng, H., Qi, B., Ge, J., Guo, Q., Zhang, W., Zhang, S., Cao, M., Lin, J., Tang, K., Gao, J., Huang, H., Gu, Y., Lyu, C., Tang, H., Wang, R., Lv, H., Ouyang, W., Wang, L., Dou, M., Zhu, X., Lu, T., Lin, D., Dai, J., Su, W., Zhou, B., Chen, K., Qiao, Y., Wang, W., and Luo, G. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency, 2025a. URL <https://arxiv.org/abs/2508.18265>.
- Wang, Y., Gao, D., Long, R., Yi, L., Jin, L., Yang, L., and Cai, X. ASPO: Adaptive sentence-level preference optimization for fine-grained multimodal reasoning. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5149–5160, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.267. URL <https://aclanthology.org/2025.findings-acl.267/>.
- Wang, Z., Guo, X., Stoica, S., Xu, H., WANG, H., Ha, H., Chen, X., Chen, Y., Yan, M., Huang, F., and Ji, H. Perception-aware policy optimization for multimodal reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=izbBqTL8vb>.
- Wei, Y., Zhao, L., Sun, J., Lin, K., jisheng yin, Hu, J., Zhang, Y., Yu, E., Lv, H., Weng, Z., Wang, J., Han, Q., Ge, Z., Zhang, X., Jiang, D., and Patel, V. M. Open vision reasoner: Transferring linguistic cognitive behavior for visual reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL <https://openreview.net/forum?id=oEgybA04dY>.
- Wu, H., Li, D., Chen, B., and Li, J. Longvideobench: A benchmark for long-context interleaved video-language understanding. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL <https://openreview.net/forum?id=3G1ZDX0I4f>.
- Xiao, T., Xu, X., Huang, Z., Gao, H., Liu, Q., Liu, Q., and Chen, E. Perception-rl: Advancing multimodal reasoning capabilities of MLLMs via visual perception reward. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=KttCXdjj4w>.
- Xu, W., Wang, J., Wang, W., Chen, Z., Zhou, W., Yang, A., Lu, L., Li, H., Wang, X., Zhu, X., Wang, W., Dai, J., and Zhu, J. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=mXuzDDVXxi>.
- Yang, J., Yang, S., Gupta, A. W., Han, R., Fei-Fei, L., and Xie, S. Thinking in space: How multimodal large language models see, remember, and recall spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10632–10643, June 2025a.
- Yang, Y., He, X., Pan, H., Jiang, X., Deng, Y., Yang, X., Lu, H., Yin, D., Rao, F., Zhu, M., Zhang, B., and Chen, W. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2376–2385, October 2025b.
- Yao, H., Huang, J., Qiu, Y., Chen, M. K., Liu, W., Zhang, W., Zeng, W., Zhang, X., Zhang, J., Song, Y., Wu, W., and Tao, D. Mmreason: An open-ended multi-modal multi-step reasoning benchmark for mllms toward agi. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 273–283, October 2025.
- Yaowei Zheng, Junting Lu, S. W. Z. F. D. K. Y. X. R. Z. Easyr1: An efficient, scalable, multi-modality rl training framework. <https://github.com/hiyouga/EasyR1>, 2025.
- Yu, E., Lin, K., Zhao, L., jisheng yin, Wei, Y., Peng, Y., Wei, H., Sun, J., Han, C., Ge, Z., Zhang, X., Jiang, D., Wang, J., and Tao, W. Perception-rl: Pioneering perception policy with reinforcement learning. In *The Thirty-ninth Annual*

- Conference on Neural Information Processing Systems*, 2025. URL <https://openreview.net/forum?id=BeXcXrXetA>.
- Yu, T., Yao, Y., Zhang, H., He, T., Han, Y., Cui, G., Hu, J., Liu, Z., Zheng, H.-T., Sun, M., and Chua, T.-S. Rlhf-v: Towards trustworthy mlms via behavior alignment from fine-grained correctional human feedback. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13807–13816, June 2024.
- Yue, X., Zheng, T., Ni, Y., Wang, Y., Zhang, K., Tong, S., Sun, Y., Yu, B., Zhang, G., Sun, H., Su, Y., Chen, W., and Neubig, G. MMMU-pro: A more robust multi-discipline multimodal understanding benchmark. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.736. URL <https://aclanthology.org/2025.acl-long.736/>.
- Zhang, F. and Nanda, N. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL <https://arxiv.org/abs/2309.16042>.
- Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.-W., Qiao, Y., Gao, P., and Li, H. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VIII*, pp. 169–186, Berlin, Heidelberg, 2024a. Springer-Verlag. ISBN 978-3-031-73241-6. doi: 10.1007/978-3-031-73242-3_10. URL https://doi.org/10.1007/978-3-031-73242-3_10.
- Zhang, R., Gui, L., Sun, Z., Feng, Y., Xu, K., Zhang, Y., Fu, D., Li, C., Hauptmann, A. G., Bisk, Y., and Yang, Y. Direct preference optimization of video large multimodal models from language model reward. In Chiruzzo, L., Ritter, A., and Wang, L. (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 694–717, Albuquerque, New Mexico, April 2025a. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.30. URL <https://aclanthology.org/2025.naacl-long.30/>.
- Zhang, R., Zhang, B., Li, Y., Zhang, H., Sun, Z., Gan, Z., Yang, Y., Pang, R., and Yang, Y. Improve vision language model chain-of-thought reasoning. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1631–1662, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.82. URL <https://aclanthology.org/2025.acl-long.82/>.
- Zhang, Z., Zhang, A., Li, M., hai zhao, Karypis, G., and Smola, A. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024b. ISSN 2835-8856. URL <https://openreview.net/forum?id=y1pPWFVfvR>.
- Zhao, X., Li, X., Duan, H., Huang, H., Li, Y., Chen, K., and Yang, H. Mg-llava: Towards multi-granularity visual instruction tuning, 2024. URL <https://arxiv.org/abs/2406.17770>.
- Zhou, J., Wang, S., Dong, J., Liu, K., Li, L., Gao, J., Jiang, J., Kong, L., and Wu, C. ProReason: Multi-modal proactive reasoning with decoupled eyesight and wisdom. In Christodoulopoulos, C., Chakraborty, T., Rose, C., and Peng, V. (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 31662–31691, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1614. URL <https://aclanthology.org/2025.emnlp-main.1614/>.

A. Experiment Setting Details

A.1. Decoding Hyperparameters

We use a uniform decoding configuration for all models and datasets. The temperature is fixed at 0.1, and the maximum generation length is set to 2048 tokens. Each experiment is repeated three times with different random seeds. We report the mean accuracy and corresponding standard deviation across runs.

A.2. Prompts

We use the instructions below. Then we extract final answers from model outputs by parsing expressions enclosed in `\boxed{ }`. The extracted predictions and ground-truth answers are then normalized using the official implementation of Lightman et al. (2024),³ and evaluated against the normalized ground-truth answers.

Prompt

Solve the complex logical reasoning problem through step-by-step reasoning. Think about the reasoning process first, answer the question and then put the final answer in one `"\boxed{ }"`.

Two-step Prompt

Solve the complex logical reasoning problem through step-by-step reasoning. First describe the image in detail, then solve the visual reasoning problem step by step, and finally put the final answer in one `"\boxed{ }"`.

Prompt for Joint Input

The description of the given image is {question}. Solve the complex logical reasoning problem through step-by-step reasoning. Think about the reasoning process first, answer the question and then put the final answer in one `"\boxed{ }"`.

Prompt for OCR Extraction

Please extract all text in the image and return it directly and exactly. Put the extracted text between `<extracted_text>` and `</extracted_text>` tags.

B. Additional Results on Modality Gap

B.1. Results on MATH-500-I with Varying Temperatures

To verify that the observed modality gap is not an artifact of a particular decoding temperature, we evaluate all models across a wide range of temperatures (0.01 to 1.0), where higher temperatures induce less greedy and more stochastic decoding. For each temperature, we run three independent trials and report the mean accuracy along with the standard deviation. We compute the modality gap as the difference between text and image accuracy. The corresponding uncertainty is estimated as $\sqrt{\text{std}_{\text{text}}^2 + \text{std}_{\text{image}}^2}$, where std_{text} and $\text{std}_{\text{image}}$ denote the standard deviations under text and image inputs at the same temperature, respectively. As shown in Figure 3, although performance exhibits moderate fluctuations across temperatures, a consistent gap between text and image inputs persists across all models. This indicates that the modality gap is robust and cannot be attributed to a specific temperature choice.

Table 8. Contingency tables between OCR extraction correctness and answer correctness on MATH-500-I, comparing the base model (B) and GRPO-finetuned model (G). Columns are aligned by outcome type (answer correct vs. wrong) to enable direct, class-wise comparison. Across all models, most errors occur despite correct extraction, indicating that failures are dominated by downstream reasoning rather than OCR. GRPO primarily shifts mass from incorrect to correct answers under correct extraction conditions, reflecting improved reasoning given the same perceived input.

Extract.	InternVL3.5-1B		Qwen2.5-VL-3B		Qwen2.5-VL-7B		Qwen3-VL-8B									
	Ans. Correct	Ans. Wrong	Ans. Correct	Ans. Wrong	Ans. Correct	Ans. Wrong	Ans. Correct	Ans. Wrong								
	B	G	B	G	B	G	B	G								
Correct	183	242	267	204	191	288	275	174	288	321	200	167	351	407	133	77
Wrong	7	10	43	44	9	16	25	21	3	3	9	9	5	7	11	9

³<https://github.com/openai/prm800k/>

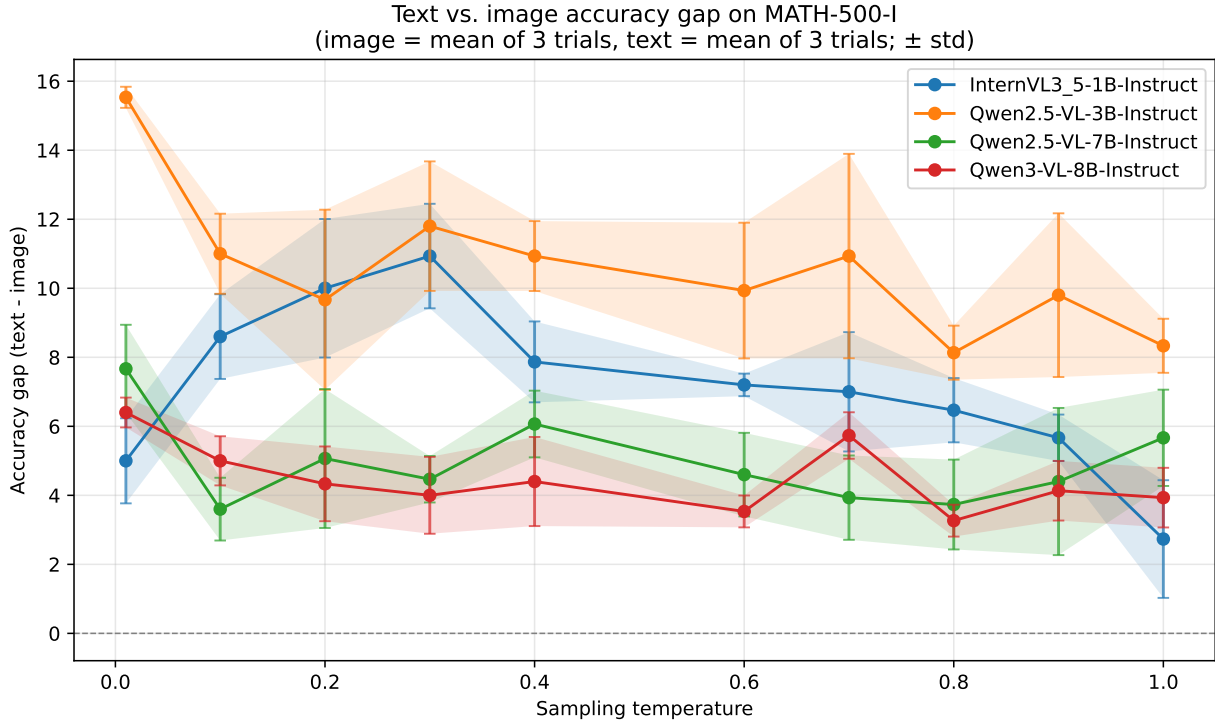


Figure 3. Accuracy gap (%) between text and image modalities on MATH-500-I across different sampling temperatures. Shaded regions denote standard deviation. Despite some variability, all models consistently exhibit a non-trivial modality gap across the full temperature range.

B.2. Contingency Tables

We report the contingency tables of all models in Table 8.

B.3. Qualitative Examples on MATH-500-I

Beyond the quantitative findings in Section 3, we present qualitative examples illustrating instances where models successfully comprehend the image content but fail in their downstream reasoning.

- Example 1.

Problem: Convert the point $(0, 3)$ in rectangular coordinates to polar coordinates. Enter your answer in the form (r, θ) , where $r > 0$ and $0 \leq \theta < 2\pi$.

Ground truth: $(3, \frac{\pi}{2})$

InternVL3.5-1B under Image Input:

To convert the point $(0, 3)$ from rectangular coordinates to polar coordinates, we need to find the radius r and the angle θ . The formulas for converting from rectangular coordinates (x, y) to polar coordinates (r, θ) are given by:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \tan^{-1} \left(\frac{y}{x} \right)$$

Given the point $(0, 3)$, we can substitute $x = 0$ and $y = 3$ into these formulas. First, we calculate the radius r :

$$r = \sqrt{0^2 + 3^2} = \sqrt{0 + 9} = \sqrt{9} = 3$$

Next, we calculate the angle θ :

$$\theta = \tan^{-1} \left(\frac{3}{0} \right)$$

However, the tangent of θ is undefined when $x = 0$ because $\tan(\theta) = \frac{y}{x}$ and $\frac{3}{0}$ is undefined. This means that the angle θ is undefined in the context of the unit circle, which corresponds to the point $(0, 3)$ lying on the positive y -axis.

Since the point $(0, 3)$ is on the positive y -axis, the angle θ is 0 radians. Therefore, the polar coordinates are:

$$(r, \theta) = (3, 0)$$

Thus, the polar coordinates of the point $(0, 3)$ are $\boxed{(3, 0)}$.

InternVL3.5-1B under Text Input:

To convert the point $(0, 3)$ from rectangular coordinates to polar coordinates, we need to find the radius r and the angle θ .

1. **Calculate the radius r :** The radius r is the distance from the origin to the point $(0, 3)$. This can be found using the Pythagorean theorem:

$$r = \sqrt{x^2 + y^2}$$

Here, $x = 0$ and $y = 3$, so:

$$r = \sqrt{0^2 + 3^2} = \sqrt{9} = 3$$

2. **Determine the angle θ :** The angle θ is the angle between the positive x -axis and the line connecting the origin to the point $(0, 3)$. Since the point $(0, 3)$ is on the positive y -axis, the angle θ is:

$$\theta = \frac{\pi}{2}$$

Therefore, the polar coordinates of the point $(0, 3)$ are $(3, \frac{\pi}{2})$.

The final answer is:

$$\boxed{\left(3, \frac{\pi}{2}\right)}$$

- Example 2.

Problem: Solve for x : $3^{2x} + 19 = 10^x$.

Ground truth: 2

Qwen2.5-VL-3B under Image Input:

To solve for x in the equation $3^{2x} + 19 = 10^x$, we can follow these steps:

1. **Rewrite the equation:** The given equation is $3^{2x} + 19 = 10^x$.

2. **Isolate the exponential terms:** We need to isolate the terms involving x . Let's rewrite the equation as:

$$3^{2x} - 10^x = -19$$

3. **Substitute $y = 10^x$:** To simplify the equation, let's introduce a new variable y such that $y = 10^x$. Then, 3^{2x} can be rewritten using the property of exponents $(a^m)^n = a^{mn}$:

$$3^{2x} = (3^2)^x = 9^x$$

So, the equation becomes:

$$9^x - y = -19$$

4. **Express 9^x in terms of y** : Since $y = 10^x$, we have $9^x = (10^x)^{\log_{10} 9} = y^{\log_{10} 9}$. Therefore, the equation is:

$$y^{\log_{10} 9} - y = -19$$

5. **Solve the equation**: This is a transcendental equation, which typically does not have a straightforward algebraic solution. However, we can use numerical methods or graphing to find the value of x .

6. **Numerical solution**: Using numerical methods or graphing tools, we find that the solution to the equation $y^{\log_{10} 9} - y = -19$ is approximately $y \approx 10$.

7. **Back-substitute to find x** : Since $y = 10^x$, we have:

$$10^x = 10$$

Taking the logarithm base 10 on both sides, we get:

$$x = 1$$

Therefore, the solution to the equation $3^{2x} + 19 = 10^x$ is $x = 1$.

1

Qwen2.5-VL-3B under Text Input:

To solve the equation $3^{2x} + 19 = 10^x$, we can start by simplifying the left side of the equation. Since $3^{2x} = (3^2)^x = 9^x$, we can rewrite the equation as:

$$9^x + 19 = 10^x$$

Now, let's consider the properties of exponents and the nature of the numbers involved. Notice that 9^x and 10^x are both exponential functions with bases that are powers of 3 and 10, respectively. However, the equation is not straightforward to solve algebraically due to the different bases and the presence of both 9^x and 10^x .

Instead, let's try to find a value of x that satisfies the equation by inspection or by using numerical methods. We can start by testing some simple values of x :

1. If $x = 1$, then $9^1 + 19 = 9 + 19 = 28$ and $10^1 = 10$. This does not satisfy the equation. 2. If $x = 2$, then $9^2 + 19 = 81 + 19 = 100$ and $10^2 = 100$. This satisfies the equation.

So, $x = 2$ is a solution to the equation. To confirm that this is the only solution, we can analyze the behavior of the functions 9^x and 10^x as x increases or decreases. Since 9^x grows slower than 10^x for $x > 0$, and since 9^x is always less than 10^x for $x > 0$, there can be no other solutions.

Therefore, the solution to the equation $3^{2x} + 19 = 10^x$ is 2.

- Example 3.

Problem: What power of 4 is equal to 8? Express your answer as a common fraction.

Ground truth: $\frac{3}{2}$

Qwen2.5-VL-7B under Image Input:

To determine what power of 4 is equal to 8, we need to express both numbers as powers of 2.

First, let's express 8 as a power of 2:

$$8 = 2^3$$

Next, let's express 4 as a power of 2:

$$4 = 2^2$$

Now, we need to find a number n such that:

$$(2^2)^n = 2^3$$

Using the property of exponents $(a^m)^n = a^{mn}$, we can rewrite the left side:

$$2^{2n} = 2^3$$

Since the bases are the same, we can set the exponents equal to each other:

$$2n = 3$$

Solving for n :

$$n = \frac{3}{2}$$

Therefore, the power of 4 that equals 8 is:

$$4^{\frac{3}{2}}$$

So, the final answer is:

$$\boxed{4^{\frac{3}{2}}}$$

Qwen2.5-VL-7B under Text Input:

To solve this problem, we need to find the power of 4 that equals 8. Let's break it down step by step:

1. **Understand the problem**: We need to find an integer n such that $4^n = 8$.
2. **Express 4 and 8 as powers of 2**: $4 = 2^2$ - $8 = 2^3$
3. **Substitute these expressions into the equation**:

$$(2^2)^n = 2^3$$

4. **Simplify the left side using the power of a power rule**:

$$2^{2n} = 2^3$$

5. **Since the bases are the same, set the exponents equal to each other**:

$$2n = 3$$

6. **Solve for n** :

$$n = \frac{3}{2}$$

Therefore, the power of 4 that equals 8 is $\frac{3}{2}$.

$$\boxed{\frac{3}{2}}$$

- Example 4.

Problem: Remy wants to divide 10 by $\frac{2}{3}$, but he cannot remember how to do that. By what number should he multiply 10 to get the answer?

Ground truth: $\frac{3}{2}$

Qwen2.5-VL-7B under Image Input:

To divide 10 by $\frac{2}{3}$, Remy needs to find a number that, when multiplied by $\frac{2}{3}$, gives 10. This can be represented mathematically as:

$$10 \div \frac{2}{3} = x$$

To solve for x , we can multiply both sides of the equation by $\frac{2}{3}$:

$$x = 10 \times \frac{3}{2}$$

Now, let's calculate this:

$$x = 10 \times \frac{3}{2} = 15$$

So, Remy should multiply 10 by 15 to get the answer.

15

Qwen2.5-VL-7B under Text Input:

To solve the problem of dividing 10 by $\frac{2}{3}$, we need to understand that dividing by a fraction is equivalent to multiplying by its reciprocal. The reciprocal of $\frac{2}{3}$ is $\frac{3}{2}$.

So, the problem can be rewritten as:

$$10 \div \frac{2}{3} = 10 \times \frac{3}{2}$$

Next, we perform the multiplication:

$$10 \times \frac{3}{2} = \frac{10 \times 3}{2} = \frac{30}{2} = 15$$

Therefore, Remy should multiply 10 by $\frac{3}{2}$ to get the answer.

The final answer is:

$\frac{3}{2}$

• Example 5.

Problem: Find the number of ordered pairs (a, b) of integers such that $|a + bi| \leq 5$.

Ground truth: 81

Qwen3-VL-8B under Image Input:

To find the number of ordered pairs (a, b) of integers such that $|a + bi| \leq 5$, we need to consider the magnitude (or modulus) of the complex number $a + bi$. The magnitude is given by:

$$|a + bi| = \sqrt{a^2 + b^2}$$

We want this magnitude to be less than or equal to 5:

$$\sqrt{a^2 + b^2} \leq 5$$

Squaring both sides, we get:

$$a^2 + b^2 \leq 25$$

This inequality represents all points (a, b) within or on the boundary of a circle centered at the origin with radius 5. We need to count all integer points (a, b) that lie within this circle.

Let's list out the possible values for a and b :

- For $a = 0$: b can be from -5 to 5 (11 values). - For $a = \pm 1$: b can be from -4 to 4 (9 values each). - For $a = \pm 2$: b can be from -3 to 3 (7 values each). - For $a = \pm 3$: b can be from -2 to 2 (6 values each). - For $a = \pm 4$: b can be from -1 to 1 (3 values each). - For $a = \pm 5$: b can be 0 (1 value each).

Now, let's sum up these values:

- For $a = 0$: 11 values. - For $a = \pm 1$: 9 values each, so $2 \times 9 = 18$ values. - For $a = \pm 2$: 7 values each, so $2 \times 7 = 14$ values. - For $a = \pm 3$: 6 values each, so $2 \times 6 = 12$ values. - For $a = \pm 4$: 3 values each, so $2 \times 3 = 6$ values. - For $a = \pm 5$: 1 value each, so $2 \times 1 = 2$ values.

Adding these together gives us the total number of ordered pairs:

$$11 + 18 + 14 + 12 + 6 + 2 = 63$$

Therefore, the number of ordered pairs (a, b) of integers such that $|a + bi| \leq 5$ is 63.

Qwen3-VL-8B under Text Input:

To solve the problem of finding the number of ordered pairs (a, b) of integers such that $|a + bi| \leq 5$, we need to understand the condition given in the problem. The expression $|a + bi|$ represents the modulus of the complex number $a + bi$, which is defined as $\sqrt{a^2 + b^2}$. The condition $|a + bi| \leq 5$ translates to $\sqrt{a^2 + b^2} \leq 5$. Squaring both sides of the inequality, we get $a^2 + b^2 \leq 25$.

We need to count all the integer pairs (a, b) that satisfy this inequality. Let's consider the possible values for a and find the corresponding values for b .

1. If $a = 0$, then $b^2 \leq 25$, so b can be any integer from -5 to 5 . This gives us 11 possible values for b (i.e., $-5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5$). 2. If $a = \pm 1$, then $1 + b^2 \leq 25$, so $b^2 \leq 24$, and b can be any integer from -4 to 4 . This gives us 9 possible values for b (i.e., $-4, -3, -2, -1, 0, 1, 2, 3, 4$). 3. If $a = \pm 2$, then $4 + b^2 \leq 25$, so $b^2 \leq 21$, and b can be any integer from -4 to 4 . This gives us 9 possible values for b (i.e., $-4, -3, -2, -1, 0, 1, 2, 3, 4$). 4. If $a = \pm 3$, then $9 + b^2 \leq 25$, so $b^2 \leq 16$, and b can be any integer from -4 to 4 . This gives us 9 possible values for b (i.e., $-4, -3, -2, -1, 0, 1, 2, 3, 4$). 5. If $a = \pm 4$, then $16 + b^2 \leq 25$, so $b^2 \leq 9$, and b can be any integer from -3 to 3 . This gives us 7 possible values for b (i.e., $-3, -2, -1, 0, 1, 2, 3$). 6. If $a = \pm 5$, then $25 + b^2 \leq 25$, so $b^2 \leq 0$, and b can only be 0. This gives us 1 possible value for b (i.e., 0).

Now, we sum up all the possible values of b for each corresponding value of a : - For $a = 0$, there are 11 values. - For $a = \pm 1$, there are 9 values each, so $2 \times 9 = 18$ values. - For $a = \pm 2$, there are 9 values each, so $2 \times 9 = 18$ values. - For $a = \pm 3$, there are 9 values each, so $2 \times 9 = 18$ values. - For $a = \pm 4$, there are 7 values each, so $2 \times 7 = 14$ values. - For $a = \pm 5$, there is 1 value each, so $2 \times 1 = 2$ values.

Adding these together, we get:

$$11 + 18 + 18 + 18 + 14 + 2 = 81.$$

Therefore, the number of ordered pairs (a, b) of integers such that $|a + bi| \leq 5$ is 81.

- Example 6.

Problem: What is the domain of the function $f(x) = \frac{2-x}{\log(2-\log(x-2))}$, where \log is the base 10 logarithm function? Express your answer in interval notation.

Ground truth: $(2, 12) \cup (12, 102)$

Qwen3-VL-8B under Image Input:

We are given the function:

$$f(x) = \frac{2-x}{\log(2-\log(x-2))}$$

and told that \log denotes the base 10 logarithm. We need to find the **domain** of this function — that is, all real numbers x for which the expression is defined.

Step 1: Identify restrictions from the denominator

The denominator is $\log(2 - \log(x - 2))$. For the logarithm to be defined, its argument must be **positive**. So we require:

$$2 - \log(x - 2) > 0$$

Also, the entire expression involves $\log(x - 2)$, so we must have:

$$x - 2 > 0 \Rightarrow x > 2$$

So, we have two conditions:

1. $x > 2$
2. $2 - \log(x - 2) > 0$

Step 2: Solve the inequality $2 - \log(x - 2) > 0$

$$2 - \log(x - 2) > 0 \Rightarrow \log(x - 2) < 2$$

Since \log is base 10, we can rewrite this as:

$$x - 2 < 10^2 = 100 \Rightarrow x < 102$$

Step 3: Combine the conditions

From above:

$$- x > 2 - x < 102$$

So, combining: $2 < x < 102$

Step 4: Check for any other restrictions

The numerator is $2 - x$, which is a polynomial and defined for all real x . So no issue there.

The only potential problem is the **logarithm in the denominator**, which we already handled: we require the argument to be **positive**, and we also require the **inner log** to be defined, which requires $x - 2 > 0$.

We have already included both of these in our domain: $x > 2$ and $x < 102$.

Step 5: Final domain in interval notation

So, the domain is:

$$(2, 102)$$

Double-check

Let's test a value in the interval, say $x = 10$:

$$- x - 2 = 8 > 0 \rightarrow \text{inner log is defined. } - \log(8) \approx 0.903 - 2 - \log(8) \approx 2 - 0.903 = 1.097 > 0 \rightarrow \text{outer log is defined.}$$

Now test a value just above 2, say $x = 2.1$:

$$- x - 2 = 0.1 > 0 - \log(0.1) = -1 - 2 - (-1) = 3 > 0 \rightarrow \text{fine.}$$

Test a value just below 102, say $x = 101.9$:

$$- x - 2 = 99.9 - \log(99.9) \approx 1.9996 - 2 - 1.9996 = 0.0004 > 0 \rightarrow \text{still positive.}$$

Test $x = 102$:

$$- x - 2 = 100 - \log(100) = 2 - 2 - 2 = 0 \rightarrow \log(0) \text{ is } \textbf{undefined} \rightarrow \text{so } x = 102 \text{ is } \textbf{excluded}.$$

Test $x = 2$:

$$- x - 2 = 0 \rightarrow \log(0) \text{ is undefined} \rightarrow \text{so } x = 2 \text{ is } \textbf{excluded}.$$

Perfect.

Final Answer:

$$(2, 102)$$

Qwen3-VL-8B under Text Input:

We are given the function:

$$f(x) = \frac{2 - x}{\log(2 - \log(x - 2))}$$

and we are to find its **domain**, i.e., the set of all real numbers x for which the function is defined. Since this is a rational function with a logarithm in the denominator, we must ensure:

1. The **denominator is not zero**. 2. The **argument of the outer logarithm** (i.e., the expression inside the log in the denominator) is **positive**. 3. The **argument of the inner logarithm** (i.e., $x - 2$) is **positive** (since log is only defined for positive reals).

Let's break this down step by step.

—
Step 1: Inner Logarithm Domain

The inner log is $\log(x - 2)$. Since the logarithm is only defined for positive arguments, we require:

$$x - 2 > 0 \Rightarrow x > 2$$

So, the domain so far is $x > 2$.

—
Step 2: Denominator Not Zero

The denominator is $\log(2 - \log(x - 2))$. This expression must be **non-zero** (since division by zero is undefined). So we require:

$$\log(2 - \log(x - 2)) \neq 0$$

Recall that $\log(a) = 0$ if and only if $a = 1$. So:

$$2 - \log(x - 2) \neq 1 \Rightarrow \log(x - 2) \neq 1$$

Solve $\log(x - 2) = 1$:

$$x - 2 = 10^1 = 10 \Rightarrow x = 12$$

So, $x = 12$ must be **excluded** from the domain.

—
Step 3: Argument of the Outer Log Must Be Positive

The outer log is $\log(2 - \log(x - 2))$. The argument of a logarithm must be **positive**, so:

$$2 - \log(x - 2) > 0 \Rightarrow \log(x - 2) < 2$$

Solve this inequality:

$$\log(x - 2) < 2 \Rightarrow x - 2 < 10^2 = 100 \Rightarrow x < 102$$

So, combining with Step 1, we now have:

$$2 < x < 102$$

But we also must exclude $x = 12$ (from Step 2).

—
Final Domain

So, the domain is all x such that $2 < x < 102$, **except** $x = 12$.

In interval notation, this is:

$$(2, 12) \cup (12, 102)$$

—
Final Answer:

$$(2, 12) \cup (12, 102)$$

B.4. Additional Results on Image-based MATH-500

Beyond the results presented in Sec. 3, we explore two prompting strategies to assess whether they can mitigate the modality-gap in performance: (1) providing a joint text-image input, and (2) using a prompt that instructs the model to first describe the image before solving the problem. As shown in Table Tab. 9, prompting alone is insufficient to close the gap. Joint inputs underperform compared to text-only inputs, while the two-step prompting yields inconsistent improvements and still falls short of text-only performance.

Table 9. Performance on semantically equivalent text and image versions of MATH-500. Despite identical semantics, all models consistently perform worse on image inputs, revealing a robust modality gap. This gap persists across both rendering schemes (MATH-500-I and MATH-500-SI), indicating that it is not due to a specific transformation. OCR results remain high, suggesting that the bottleneck lies in reasoning from image-conditioned representations rather than reading ability.

Dataset	Input	InternVL3.5-1B	Qwen2.5-VL-3B	Qwen2.5-VL-7B	Qwen3-VL-8B
MATH-500	Text	45.07 ± 0.12	50.87 ± 0.12	63.80 ± 1.06	75.53 ± 0.42
MATH-500-I	OCR	90.00 ± 0.35	92.93 ± 0.31	97.47 ± 0.23	96.73 ± 0.31
	Extracted text	44.73 ± 0.64	52.93 ± 0.50	64.80 ± 0.87	76.40 ± 0.53
	Image	37.27 ± 0.31	39.93 ± 0.64	58.27 ± 0.31	70.87 ± 0.64
	Joint input	43.20 ± 0.20	43.27 ± 0.31	61.73 ± 1.30	73.07 ± 0.61
	Two-step	37.53 ± 0.99	52.13 ± 0.76	57.93 ± 0.58	71.16 ± 0.46
MATH-500-SI	OCR	84.83 ± 0.15	87.33 ± 0.31	96.27 ± 0.31	92.47 ± 0.31
	Extracted text	43.40 ± 0.53	47.93 ± 0.81	63.67 ± 0.31	75.53 ± 0.31
	Square image	32.47 ± 1.33	35.07 ± 1.10	55.93 ± 0.31	68.60 ± 0.20
	Joint input	44.33 ± 0.90	37.40 ± 0.35	61.80 ± 0.40	72.53 ± 0.61
	Two-step	37.53 ± 1.00	48.40 ± 0.87	55.80 ± 1.31	68.33 ± 0.90

B.5. Results on Image-based MATH-500 under Additional Transformations

Beyond the standard black-text-on-white setting in Section 3, we evaluate two additional transformations with inverted color schemes (white text on a black background). The first directly renders the original problems with inverted colors, yielding MATH-500- \hat{I} . The second further enforces a square aspect ratio to examine the effect of spatial layout, resulting in MATH-500- \widehat{SI} .

As shown in Table 10, models consistently exhibit a substantial modality gap between text and image inputs, despite semantic equivalence. This gap persists across both transformations, indicating that it is not attributable to a specific rendering choice.

B.6. Analysis on Generation Lengths

We report the average number of generated tokens for both base and GRPO-finetuned models on MATH-500 and MATH-500-I in Table 11. Overall, several consistent trends emerge. First, for all base models, generation lengths on MATH-500 and MATH-500-I are largely comparable, suggesting that input modality (text vs. image) does not substantially affect the verbosity of reasoning traces, so the observed modality reasoning-access gap in performance is unlikely to be driven by differences in output length. Figures 4, 5, 6, and 7 further show that the length distributions remain similar across modalities. Second, all models exhibit high variance in generation length, with standard deviations often exceeding half of the mean, indicating considerable variability in reasoning depth across problems. Third, the impact of GRPO fine-tuning is model-dependent: it produces shorter, more concise outputs for InternVL3.5-1B and Qwen3-VL-8B, while increasing generation length for Qwen2.5-VL-3B and Qwen2.5-VL-7B, suggesting more elaborate reasoning. Finally, the inconsistent changes in length across models indicate that performance gains from GRPO are unlikely to stem purely from verbosity, but rather from improved reasoning quality.

Table 10. Performance on MATH-500 under inverted-color and square-layout transformations. Despite semantic equivalence, a consistent modality gap between text and image inputs persists across all settings.

Dataset	Input	InternVL3.5-1B	Qwen2.5-VL-3B	Qwen2.5-VL-7B	Qwen3-VL-8B
MATH-500	Text	45.07 ± 0.12	50.87 ± 0.12	63.80 ± 1.06	75.53 ± 0.42
MATH-500- \hat{I}	OCR	89.53 ± 0.83	91.20 ± 0.72	94.27 ± 0.12	95.67 ± 0.12
	Extracted text	42.53 ± 1.03	50.80 ± 1.11	63.00 ± 0.60	75.27 ± 0.23
	Image	34.53 ± 1.01	39.60 ± 1.22	57.93 ± 0.70	70.20 ± 0.35
	Joint input	44.20 ± 0.53	42.87 ± 0.42	61.80 ± 0.53	70.93 ± 0.42
	Two-step	41.80 ± 0.35	47.00 ± 1.59	58.13 ± 0.61	68.53 ± 0.61
MATH-500- \hat{SI}	OCR	82.07 ± 0.12	84.67 ± 0.23	95.60 ± 0.20	92.53 ± 0.31
	Extracted text	41.27 ± 0.50	47.93 ± 0.61	61.20 ± 0.35	74.00 ± 0.20
	Square image	37.73 ± 0.70	40.93 ± 1.36	55.60 ± 0.60	62.40 ± 0.72
	Joint input	47.27 ± 0.81	40.33 ± 1.50	61.13 ± 0.42	69.40 ± 0.92
	Two-step	39.93 ± 0.81	48.13 ± 0.50	54.33 ± 0.31	60.80 ± 0.53

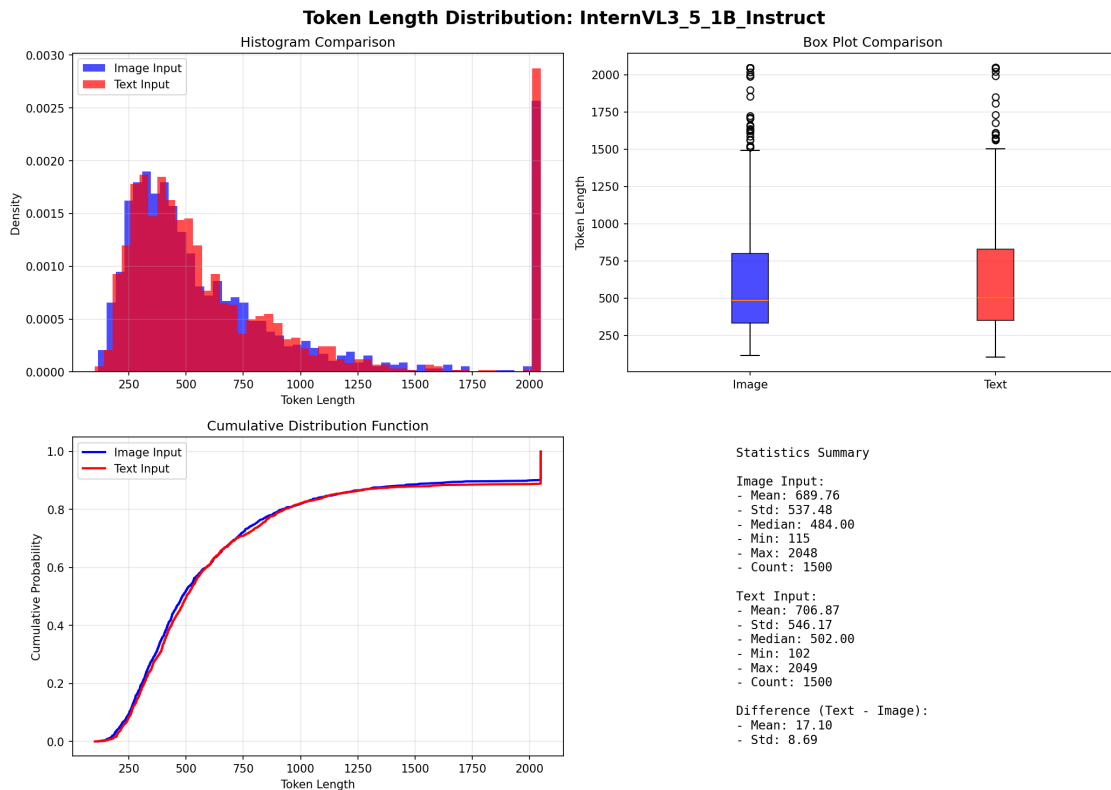


Figure 4. Token length distribution comparison between Image and Text inputs for InternVL3.5-1B. The analysis includes a histogram of probability density, box plots detailing quartile spread and outliers, a cumulative distribution function (CDF), and summary statistics.

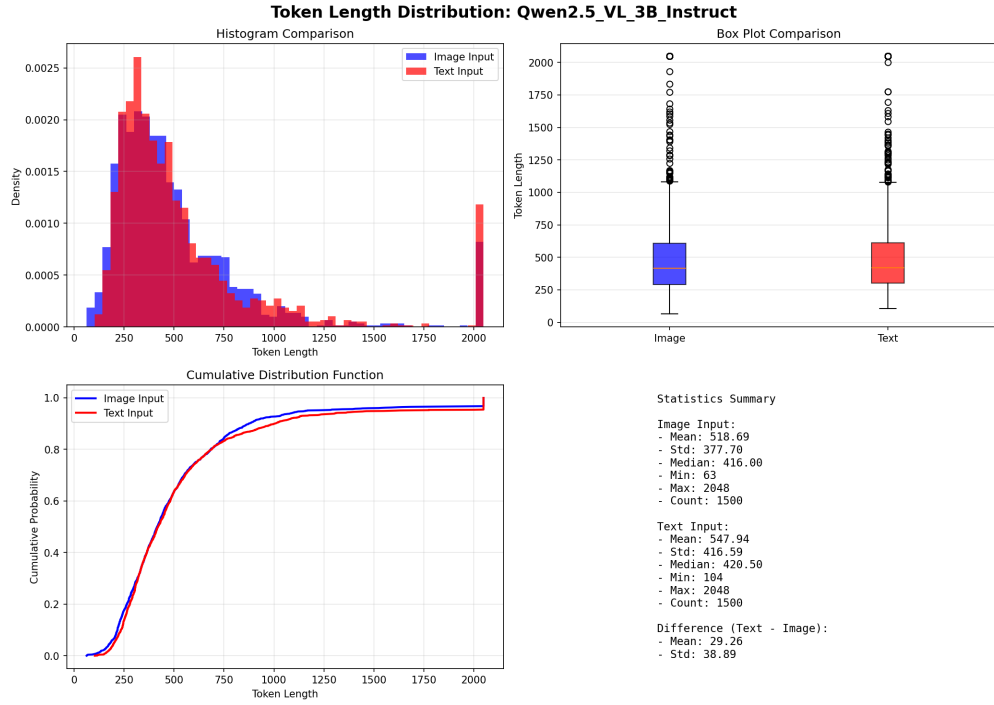


Figure 5. Token length distribution comparison between Image and Text inputs for Qwen2.5-VL-3B. The analysis includes a histogram of probability density, box plots detailing quartile spread and outliers, a cumulative distribution function (CDF), and summary statistics.

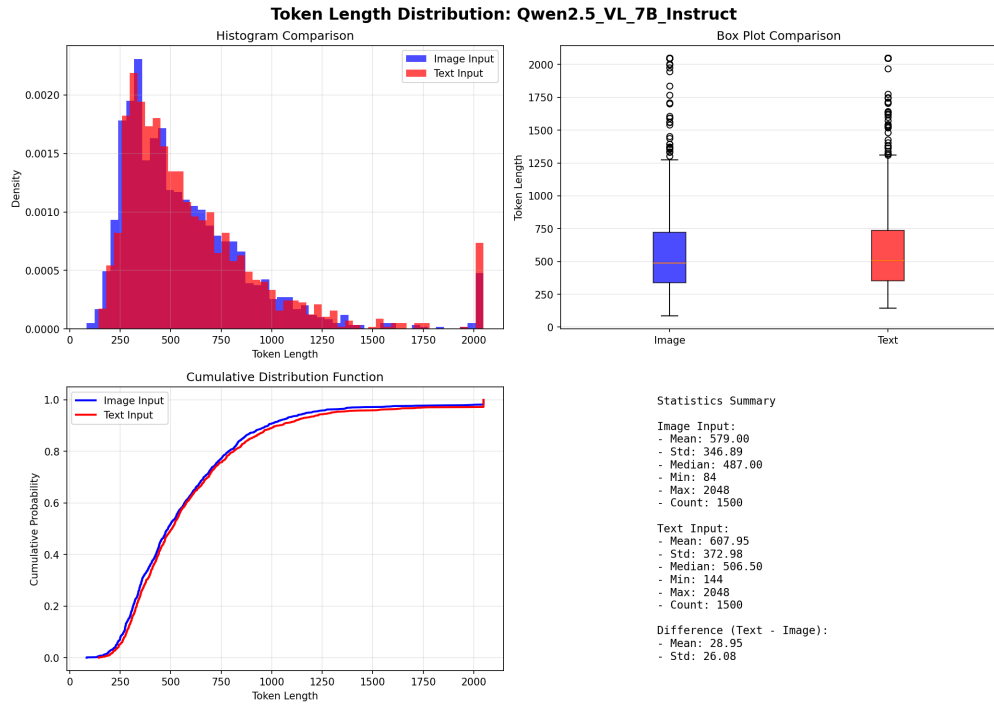


Figure 6. Token length distribution comparison between Image and Text inputs for Qwen2.5-VL-7B. The analysis includes a histogram of probability density, box plots detailing quartile spread and outliers, a cumulative distribution function (CDF), and summary statistics.

Table 11. Generation lengths (mean \pm standard deviation) of base and GRPO-finetuned models on MATH-500 and MATH-500-I, showing comparable verbosity across datasets and model-dependent effects of GRPO.

Model	Dataset	Generation Length
InternVL3.5-1B	MATH-500	706.9 \pm 546.3
InternVL3.5-1B	MATH-500-I	689.8 \pm 537.7
InternVL3.5-1B w/ our training	MATH-500-I	655.0 \pm 354.7
Qwen2.5-VL-3B	MATH-500	547.9 \pm 416.9
Qwen2.5-VL-3B	MATH-500-I	518.7 \pm 377.5
Qwen2.5-VL-3B w/ our training	MATH-500-I	577.2 \pm 333.9
Qwen2.5-VL-7B	MATH-500	608.0 \pm 373.1
Qwen2.5-VL-7B	MATH-500-I	579.0 \pm 346.5
Qwen2.5-VL-7B w/ our training	MATH-500-I	633.4 \pm 355.7
Qwen3-VL-8B	MATH-500	948.5 \pm 678.7
Qwen3-VL-8B	MATH-500-I	985.4 \pm 689.4
Qwen3-VL-8B w/ our training	MATH-500-I	888.9 \pm 632.2

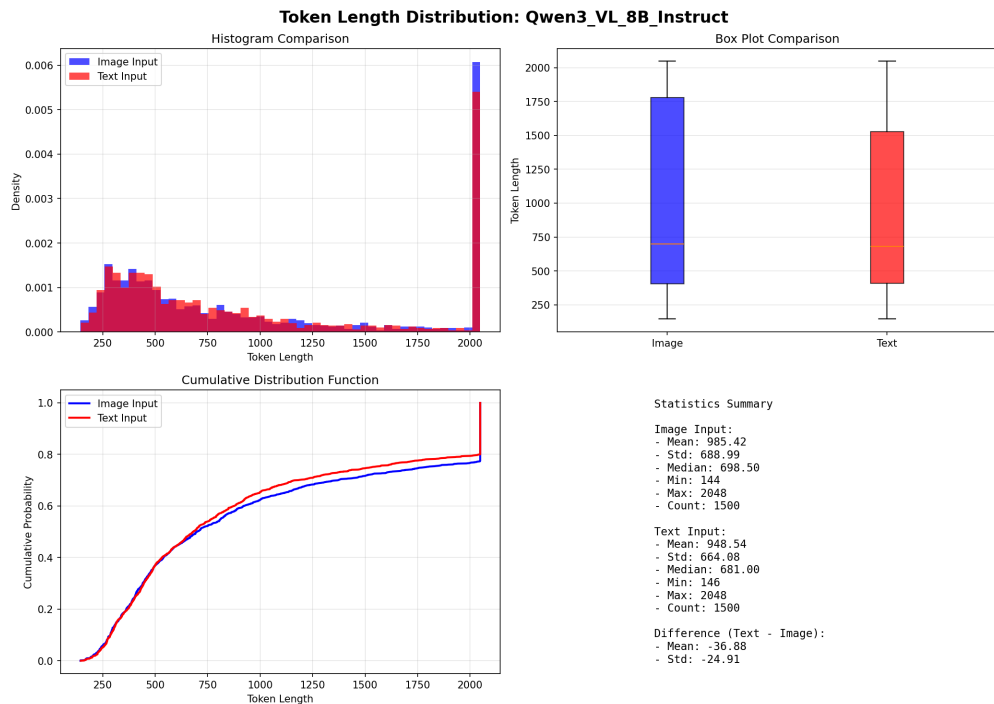


Figure 7. Token length distribution comparison between Image and Text inputs for Qwen3-VL-8B. The analysis includes a histogram of probability density, box plots detailing quartile spread and outliers, a cumulative distribution function (CDF), and summary statistics.

B.7. Results on Additional Datasets

To determine whether the observed modality gap persists across other domains, we extend our analysis beyond MATH-500 (Lightman et al., 2024) and BBH (Suzgun et al., 2023) to evaluate the GSM8K dataset (Cobbe et al., 2021). Applying the same transformation \mathcal{T} , we construct GSM8K-I, an image-based variant of the dataset. As shown in Table 12, while a performance disparity between text and image inputs remains prominent in smaller models (e.g., InternVL3.5-1B), this gap narrows considerably in more advanced models.

Table 12. Performance comparison on the standard GSM8K dataset and its image-based variant (GSM8K-I). While early models exhibit a noticeable modality gap, advanced models demonstrate a substantially reduced disparity between text and image inputs.

Dataset (Modality)	InternVL3.5-1B	Qwen2.5-VL-3B	Qwen2.5-VL-7B	Qwen3-VL-8B
GSM8K (Text)	66.14 ± 0.52	82.31 ± 0.37	86.73 ± 0.66	93.58 ± 0.39
GSM8K-I (Image)	43.57 ± 0.80	76.45 ± 0.64	85.92 ± 0.50	92.34 ± 0.76

C. Broader Impacts

This work studies and mitigates a modality-dependent reasoning gap in MLLMs, where models may reason more effectively from text than from semantically equivalent visual inputs. By improving the alignment between visual inputs and downstream reasoning, our approach may support more reliable multimodal systems in applications such as education, accessibility, and visual question answering. In particular, reducing failures caused by input modality could make MLLMs more useful for users who rely on visual representations of information. At the same time, more capable MLLMs may be used in ways not intended by the authors, including potential misuse in surveillance or the unauthorized analysis of sensitive visual data. Furthermore, because our pipeline relies on existing text math corpora, the resulting models may inherit coverage gaps and biases present in the source data. We therefore encourage careful evaluation, appropriate human oversight, and domain-specific safeguards before applying such systems in practical scenarios.

D. Training Setting Details

D.1. Training Prompts and Format Rewards

We use the instructions below when training Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct.

Prompt

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

As InternVL3.5-1B-Instruct and Qwen3-VL-8B-Instruct models do not output the token <think>, we use a slightly modified prompt.

Prompt

You FIRST think about the reasoning process as an internal monologue and then provide the final answer. Think about the reasoning process first and then answer the question. The final answer MUST BE put in \boxed{ }.

D.2. Training Losses

Group Relative Policy Optimization (GRPO). Following Shao et al. (2024), for each image prompt $x_i^v \in \mathcal{D}$, we sample a group of K responses

$$\{o_i^{(k)}\}_{k=1}^K \sim \pi_\theta(\cdot | x_i^v),$$

and compute the corresponding rewards $r(o_i^{(k)}, y_i)$. We then normalize rewards within each group to obtain standardized advantages:

$$A_i^{(k)} = \frac{r(o_i^{(k)}, y_i) - \text{mean}(\{r(o_i^{(j)}, y_i)\}_{j=1}^K)}{\text{std}(\{r(o_i^{(j)}, y_i)\}_{j=1}^K)} \quad (3)$$

The policy π_θ is optimized using a clipped surrogate objective:

$$\mathbb{E}_{x_i^v \in \mathcal{D}, \{o_i^{(k)}\}_{k=1}^K \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{K} \sum_{k=1}^K \frac{1}{|o_i^{(k)}|} \left\{ \begin{aligned} & \min \left[s_t(x_i^v, o_i^{(k)}) A_i^{(k)}, \right. \\ & \left. \text{clip}(s_t(x_i^v, o_i^{(k)}), 1 - \varepsilon, 1 + \varepsilon) A_i^{(k)} \right] \\ & \left. - \delta \cdot \text{KL}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right\} \right] \quad (4) \end{aligned}$$

where the importance ratio is defined as

$$s_{i,t}^{(k)} = \frac{\pi_{\theta}(o_{i,t}^{(k)} \mid x_i^v, o_{i,<t}^{(k)})}{\pi_{\theta_{\text{old}}}(o_{i,t}^{(k)} \mid x_i^v, o_{i,<t}^{(k)})}.$$

Here, ε is the clipping parameter, δ controls the KL penalty, π_{θ} is the current policy, $\pi_{\theta_{\text{old}}}$ is the behavior policy used for sampling, and π_{ref} is a fixed reference model initialized from the initial policy.

Supervised Fine-Tuning (SFT). Once having the SFT dataset $\mathcal{D}_{\text{SFT}} = \{(x_i^v, a_i^+)\}_{i=1}^{|\mathcal{D}|}$, we optimize the standard next-token prediction objective

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \log \pi_{\theta}(a_i^+ \mid x_i^v), \quad (5)$$

where θ denotes the trainable model parameters.

Direct Preference Optimization (DPO). After constructing the DPO dataset $\mathcal{D}_{\text{DPO}} = \{(x_i^v, a_i^+, a_i^-)\}_{i=1}^{|\mathcal{D}|}$, we train the model to make it prefer the correct answer a_i^+ than incorrect answer a_i^- . Formally, given a reference model π_{ref} , initialized from the base model, we optimize the standard DPO objective:

$$\mathcal{L}_{\text{DPO}}(\theta) = -\frac{1}{|\mathcal{D}|} \sum_{i=1}^{|\mathcal{D}|} \log \sigma \left(\mu \left[\log \frac{\pi_{\theta}(a_i^+ \mid x_i^v)}{\pi_{\text{ref}}(a_i^+ \mid x_i^v)} - \log \frac{\pi_{\theta}(a_i^- \mid x_i^v)}{\pi_{\text{ref}}(a_i^- \mid x_i^v)} \right] \right), \quad (6)$$

where $\sigma(\cdot)$ is the sigmoid function and μ controls the sharpness of the preference optimization.

D.3. Training Hyperparameters

For SFT, we use a learning rate of 1e-5, whereas for DPO and GRPO we set it to 1e-6. Across training objectives, we use 3 epochs for SFT, 15 epochs for GRPO, and 1 epoch for DPO in all experiments. The maximum lengths of prompts and generations are 2048. The number of rollouts is 8, and the coefficient for entropy loss is 0.001. For GRPO, we remove KL regularization ($\delta = 0$), and for DPO, we set $\mu = 0.2$. For computing the final reward, as the answer correctness r_a plays an important role in enhancing the multimodal reasoning capabilities of MLLMs while the format reward r_f only enforces structured generations, α is set to 0.9, and β is 0.1. We use AdamW (Loshchilov & Hutter, 2019) for all settings. For the remaining hyperparameters, we adopt the settings from Lee (2024) for SFT and DPO, and from EasyR1 (Yaowei Zheng, 2025) for GRPO when fine-tuning the Qwen-VL family. For InternVL3.5-1B, we use the official InternVL repository’s hyperparameters for SFT and DPO,⁴ and follow EasyR1 (Yaowei Zheng, 2025) for GRPO. Given the differences in training epochs across methods, we provide an analysis in Sec. F.1 to justify the selected epoch counts for SFT and DPO. The analysis shows that extending SFT or DPO training does not improve performance, despite incurring additional computational cost.

D.4. Training Data Statistics

After applying the procedure described in Subsection 4.1, we obtain model-specific training datasets \mathcal{D} . Table 13 summarizes the resulting dataset sizes for each model.

⁴<https://github.com/OpenGVLab/InternVL>

D.5. Resources

Most experiments are conducted using four H200 GPUs. For evaluation, each run is performed on a single H200 GPU. To enable efficient generation of the training dataset \mathcal{D} and streamline evaluation, we utilize vLLM (Kwon et al., 2023).

Table 13. Training dataset sizes for each model after applying the data construction procedure described in Subsection 4.1.

Model	Training data size
InternVL3.5-1B	1586
Qwen2.5-VL-3B	4043
Qwen2.5-VL-7B	3701
Qwen3-VL-8B	3401

E. Case Studies

E.1. Case Study 1



Figure 8. Image of case study 1.

Image: The image is shown in Figure 8.

Prompt: <image_1> Can you find the missing term?

Ground-truth: 10

Base Qwen2.5-VL-7B:

To find the missing term in the grid, let's analyze the pattern in the rows and columns.

- First Row:** - 6, 9, 15 - The difference between the numbers is: $9 - 6 = 3$ - $15 - 9 = 6$ - The differences are 3 and 6, which are multiples of 3.
- Second Row:** - 8, 12, 20 - The difference between the numbers is: $12 - 8 = 4$ - $20 - 12 = 8$ - The differences are 4 and 8, which are multiples of 4.
- Third Row:** - 4, 6, ? - The difference between the numbers is: $6 - 4 = 2$ - The difference is 2, which is a multiple of 2.

Given the pattern of multiples, the missing term should be a multiple of 2. The next multiple of 2 after 6 is 8. Therefore, the missing term is **8**.

SFT-finetuned Qwen2.5-VL-7B:

To find the missing term in the given grid, we need to identify the pattern or rule that governs the numbers in each row and column.

Let's start by examining the first row:

6, 9, 15

We can see that:

$$9 = 6 + 3$$

$$15 = 9 + 6$$

So, the difference between consecutive terms increases by 3 each time.

Now, let's look at the second row:

$$8, 12, 20$$

We can see that:

$$12 = 8 + 4$$

$$20 = 12 + 8$$

Again, the difference between consecutive terms increases by 4 each time.

Next, let's examine the third row:

$$4, 6, ?$$

Following the same pattern, the difference between the first two terms is 2:

$$6 = 4 + 2$$

If we continue this pattern, the next difference should be 3:

$$? = 6 + 3 = 9$$

Therefore, the missing term in the third row is:

9

DPO-finetuned Qwen2.5-VL-7B:

To find the missing term in the grid, let's analyze the pattern in the rows and columns.

First, let's look at the first row: - The numbers are 6, 9, and 15. - The difference between 9 and 6 is 3. - The difference between 15 and 9 is 6. - The pattern seems to be increasing by 3 each time.

Next, let's look at the second row: - The numbers are 8, 12, and 20. - The difference between 12 and 8 is 4. - The difference between 20 and 12 is 8. - The pattern seems to be increasing by 4 each time.

Now, let's look at the third row: - The numbers are 4, 6, and the missing term. - The difference between 6 and 4 is 2. - The pattern seems to be increasing by 2 each time.

Following the pattern of increasing by 2 each time, the next number after 6 should be: $- 6 + 2 = 8$.

Therefore, the missing term is 8.

GRPO_{text}-finetuned Qwen2.5-VL-7B:

To find the missing term in the grid, let's analyze the pattern in the rows and columns.

First, let's look at the first row: - The numbers are 6, 9, and 15. - The difference between 9 and 6 is 3. - The difference between 15 and 9 is 6. - It seems like the differences are increasing by 3 each time: 3, 6.

Now, let's look at the second row: - The numbers are 8, 12, and 20. - The difference between 12 and 8 is 4. - The difference between 20 and 12 is 8. - Again, the differences are increasing by 4 each time: 4, 8.

Now, let's look at the third row: - The numbers are 4, 6, and the missing term. - The difference between 6 and 4 is 2. - We need to find the difference that would continue the pattern. The differences in the first two rows were increasing by 3 and 4. Following this pattern, the next difference should be 5. - So, the difference between the missing term and 6 should be 5. - Therefore, the missing term is $6 + 5 = 11$.

So, the missing term is 11.

GRPO-finetuned Qwen2.5-VL-7B:

To find the missing term in the grid, let's analyze the pattern in each row and column.

First, let's look at the rows:

1. Row 1: 6, 9, 15 2. Row 2: 8, 12, 20 3. Row 3: 4, 6, ?

For Row 1, the pattern seems to be: - 9 is $6 + 3$ - 15 is $9 + 6$

For Row 2, the pattern seems to be: - 12 is $8 + 4$ - 20 is $12 + 8$

For Row 3, we need to find the pattern: - 6 is $4 + 2$ - The next number should be $6 + 4 = 10$

So, the missing term in the grid is 10.

Analysis: All five models exhibit flawless perception, with accurate OCR and correct mapping of grid values to spatial positions. Failures in the Base, SFT, DPO, and GRPO_{text} models arise from a logical abstraction bottleneck: despite perfect visual parsing, they fail to form a coherent representation of the governing matrix rules. Specifically, they cannot bridge the modality gap between local computations (row-wise differences) and global constraints (proportional scaling or $c_1 + c_2 = c_3$), instead relying on spurious heuristics that violate global structure. In contrast, the image-conditioned GRPO model successfully integrates perception with grounded arithmetic reasoning, correctly inferring the intra-row dependency and applying it to recover the target value.

E.2. Case Study 2

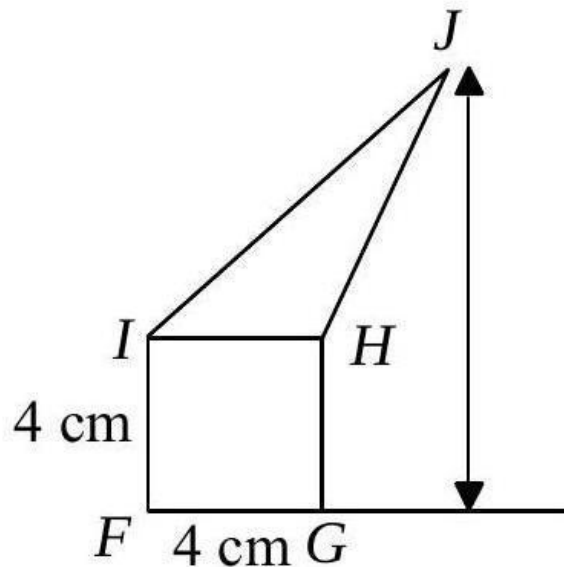


Figure 9. Image of case study 2.

Image: The image is shown in Figure 9.

Prompt: Hint: Please answer the question and provide the final answer at the end. Question: The triangle HJI has the same area as the square $FGHI$, whose sides are of length 4 cm. What is the perpendicular distance, in cm, of the point J from the line extended through F and G ?

Ground-truth: 12

Base Qwen3-VL-8B:

We are given a square $FGHI$ with side length 4 cm, and a triangle HJI that has the same area as the square. We are to find the perpendicular distance from point J to the line extended through F and G (which is the base line of the square).

—
Step 1: Area of the square

The area of square $FGHI$ is:

$$\text{Area}_{\text{square}} = \text{side}^2 = 4^2 = 16 \text{ cm}^2$$

Since triangle HIJ has the same area, we have:

$$\text{Area}_{\triangle HIJ} = 16 \text{ cm}^2$$

—
****Step 2: Use the formula for the area of a triangle****

The area of a triangle is:

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height}$$

In triangle HIJ , if we take HI as the base, then the height is the perpendicular distance from point J to the line containing HI . But note: the problem asks for the perpendicular distance from J to the line extended through F and G . Since FG is the base of the square and HI is directly above it (as it's a square), the line FG and HI are parallel (both horizontal if we assume standard orientation). So the perpendicular distance from J to the line FG (extended) is the same as the perpendicular distance from J to the line HI — because HI and FG are parallel. Therefore, we can use HI as the base of the triangle, and the height will be the perpendicular distance from J to the line HI (or equivalently, to the line FG).

Length of base $HI = 4$ cm (since it's a side of the square).

So,

$$16 = \frac{1}{2} \times 4 \times h$$

Solve for h :

$$16 = 2h \Rightarrow h = 8$$

—
****Final Answer:****

8

SFT-finetuned Qwen3-VL-8B:

We are given a square $FGHI$ with side length 4 cm, and a triangle HIJ that has the ****same area**** as the square. We are to find the ****perpendicular distance**** from point J to the line extended through F and G — which is the same as the base line of the square (since F and G are adjacent corners on the bottom side).

—
 Step 1: Area of the square

The area of square $FGHI$ is:

$$\text{Area}_{\text{square}} = \text{side}^2 = 4^2 = 16 \text{ cm}^2$$

Since triangle HIJ has the same area, we have:

$$\text{Area}_{\triangle HIJ} = 16 \text{ cm}^2$$

—
 Step 2: Use the formula for the area of a triangle

The area of a triangle is:

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height}$$

In triangle HIJ , we can take HI as the base. Since HI is a side of the square, its length is 4 cm. Let h be the perpendicular distance from point J to the line HI (which is the same as the line through F and G , since FG and HI are opposite sides of the square and parallel). So, the height of the triangle is h . Then:

$$\frac{1}{2} \times 4 \times h = 16$$

Simplify:

$$2h = 16 \Rightarrow h = 8$$

Final Answer:

8

The perpendicular distance from point J to the line extended through F and G is 8 cm.

DPO-finetuned Qwen3-VL-8B:

We are given a square $FGHI$ with side length 4 cm, and a triangle HIJ that has the **same area** as the square. We are to find the **perpendicular distance** from point J to the line extended through F and G . Since F and G lie on the same horizontal line (as shown in the diagram), this line is horizontal. So we need the **vertical height** from point J to this horizontal line — which is exactly what the arrow labeled with J indicates.

Step 1: Area of the square

The square $FGHI$ has side length 4 cm, so its area is:

$$\text{Area}_{\text{square}} = 4 \times 4 = 16 \text{ cm}^2$$

Since triangle HIJ has the same area:

$$\text{Area}_{\triangle HIJ} = 16 \text{ cm}^2$$

Step 2: Area of triangle HIJ

Triangle HIJ has base HI , which is a side of the square, so:

$$HI = 4 \text{ cm}$$

Let h be the perpendicular distance from point J to the line containing HI (which is the same line as FG , since $FGHI$ is a square). This distance h is exactly what we are asked to find.

The area of a triangle is:

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height}$$

So,

$$16 = \frac{1}{2} \times 4 \times h$$

Simplify:

$$16 = 2h \Rightarrow h = 8$$

Final Answer:

8

GRPO_{text}-finetuned Qwen3-VL-8B:

We are given a square $FGHI$ with side length 4 cm, and a triangle HIJ that has the same area as the square. We are to find the perpendicular distance from point J to the line extended through F and G (which is the base line of the square).

—
****Step 1: Area of the square****

The area of square $FGHI$ is:

$$\text{Area}_{\text{square}} = \text{side}^2 = 4^2 = 16 \text{ cm}^2$$

Since triangle HIJ has the same area, we have:

$$\text{Area}_{\triangle HIJ} = 16 \text{ cm}^2$$

—
****Step 2: Use the formula for the area of a triangle****

The area of a triangle is:

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height}$$

In triangle HIJ , if we take HI as the base, then the height is the perpendicular distance from point J to the line containing HI . But note: the problem asks for the perpendicular distance from J to the line extended through F and G . Since FG is the base of the square and HI is directly above it (as it's a square), the line FG and HI are parallel (both horizontal if we assume standard orientation). So the perpendicular distance from J to the line FG (extended) is the same as the perpendicular distance from J to the line HI — because HI and FG are parallel. Therefore, we can use HI as the base of the triangle, and the height will be the perpendicular distance from J to the line HI (or equivalently, to the line FG).

Length of base $HI = 4$ cm (since it's a side of the square).

So,

$$16 = \frac{1}{2} \times 4 \times h$$

Solve for h :

$$16 = 2h \Rightarrow h = 8$$

—
****Final Answer:****

8

GRPO-finetuned Qwen3-VL-8B:

We are given a square $FGHI$ with side length 4 cm, and a triangle HIJ that has the same area as the square. We are to find the perpendicular distance from point J to the line extended through F and G (which is the base line of the square).

—
****Step 1: Area of the square****

The square $FGHI$ has side 4 cm, so its area is:

$$\text{Area}_{\text{square}} = 4 \times 4 = 16 \text{ cm}^2$$

Since triangle HIJ has the same area, we have:

$$\text{Area}_{\triangle HIJ} = 16 \text{ cm}^2$$

—
****Step 2: Understanding the triangle HIJ****

From the diagram (as described in the problem and standard interpretation):

- Square $FGHI$ is placed with F at bottom left, G at bottom right, H at top right, I at top left. - So, FG is the base on the x-axis, length 4 cm. - HI is the top side of the square, from H to I , horizontal, length 4 cm. - Point J is above the square, connected to H and I , forming triangle HIJ .

So, triangle HIJ has base $HI = 4$ cm (same as square's side), and height from J perpendicular to line HI .

But note: the question asks for the perpendicular distance from J to the line extended through F and G , which is the ****horizontal base line**** (the line containing FG , which is the x-axis in our coordinate system).

So, we need the vertical height from J to the x-axis.

—
****Step 3: Use area of triangle to find height****

Triangle HIJ has base $HI = 4$ cm. Let's denote the height from J to line HI as h_1 . Then:

$$\text{Area} = \frac{1}{2} \times \text{base} \times \text{height} = \frac{1}{2} \times 4 \times h_1 = 2h_1$$

Set equal to 16:

$$2h_1 = 16 \Rightarrow h_1 = 8 \text{ cm}$$

So, the perpendicular distance from J to line HI is 8 cm.

—
****Step 4: Find distance from J to line FG (x-axis)****

Now, since HI is the top side of the square, and the square has height 4 cm, the line HI is 4 cm above the base line FG .

Therefore, if J is 8 cm above HI , then the total distance from J to the base line FG is:

$$4 + 8 = 12 \text{ cm}$$

—
****Final Answer:****

12

Analysis: All five models successfully process the image without any perception errors. They demonstrate highly accurate OCR and visual entity extraction. Every model correctly reads the 4 cm dimension, identifies the constituent shapes (square $FGHI$ and triangle HIJ), and successfully maps the text constraints to the visual diagram. The failure point across the Base, SFT, DPO, and GRPO_{text} models is strictly a spatial reasoning bottleneck. While their perception is highly accurate, they fail to maintain a coherent internal representation of the geometry, ultimately failing to bridge the modality reasoning gap when transitioning from calculating a local variable (triangle height) to evaluating the global structure (total height). The image-conditioned GRPO approach is the only model here that successfully chains accurate visual perception with rigorous, grounded spatial logic.

E.3. Case Study 3

Image: The image is shown in Figure 10.

Prompt: Hint: Please answer the question and provide the final answer at the end. Question: Out of how many circles is the beaver made of?

Ground-truth: 8

Base Qwen3-VL-8B:



Figure 10. Image of case study 3.

To determine the number of circles in the beaver image, let's break it down:

- The main head is one large yellow circle. - There are two blue ears, each of which is a circle (with a smaller white circle inside, but the ear itself is a circle). - There are two orange eyes, each a small circle. - There is one purple mouth, which is also a circle.

Counting them: - 1 (head) + 2 (ears) + 2 (eyes) + 1 (mouth) = 6 circles.

The hint doesn't change the count; it's just a prompt to answer. The beaver is made of 6 circles.

Final Answer: 6

SFT-finetuned Qwen3-VL-8B:

Let's count the circles in the image:

- There are **two** blue circles on top — these represent the ears. - There is **one** large yellow circle in the center — this is the head. - Inside the yellow circle, there are **two** orange circles — these are the eyes. - There is **one** purple circle — this is the mouth.

Now, adding them up:

2 (ears) + 1 (head) + 2 (eyes) + 1 (mouth) = **6 circles**

Final Answer: **6**

DPO-finetuned Qwen3-VL-8B:

Let's count the circles in the image:

- There are **two** light blue circles on top (the ears). - There is **one** large yellow circle in the center (the head). - There are **two** small orange circles (eyes). - There is **one** purple circle (mouth).

Adding them up: 2 (ears) + 1 (head) + 2 (eyes) + 1 (mouth) = **6 circles**

The beaver is made of **6 circles**.

Final Answer: 6

GRPO_{text}-finetuned Qwen3-VL-8B:

To determine the number of circles in the beaver figure, let's break it down by parts:

- The **head** is one large yellow circle. - There are **two ears**, each represented by a light blue circle (with a white inner circle, but we count the outer blue shape as one circle per ear). - The **eyes** are two small orange

circles. - The **mouth** is one purple circle.

Now, adding them up: - 1 (head) + 2 (ears) + 2 (eyes) + 1 (mouth) = 6 circles

Note: The white inner part of each ear is also a circle, but since the hint and context suggest counting the main shapes that form the beaver, and given the standard interpretation of such puzzles, we usually count the distinct outer shapes. However, if we count **all** circles including the white ones, then:

- Head: 1 - Ears: 2 (blue) + 2 (white) = 4 - Eyes: 2 - Mouth: 1

That would be $1+4+2+1=8$, which is not typical for this style of question.

Given the visual simplicity and common puzzle conventions, the most reasonable answer is to count the **outer, colored shapes** that define the beaver's features.

Thus, the beaver is made of **6 circles**.

Final Answer: 6

GRPO-finetuned Qwen3-VL-8B:

To determine how many circles make up the beaver in the image, let's break it down:

- The **head** is one large yellow circle. - There are **two ears**, each represented by a light blue circle (with a white inner circle, but we count the outer blue circle as one circle per ear). - The **eyes** are two orange circles. - The **mouth** is one purple circle.

Now, counting them:

1. Head: 1 circle 2. Left ear: 1 circle 3. Right ear: 1 circle 4. Left eye: 1 circle 5. Right eye: 1 circle 6. Mouth: 1 circle

Total = $1 + 1 + 1 + 1 + 1 + 1 = 7$ circles

Note: Even though the ears have white inner circles, the hint and common interpretation in such puzzles usually count the outer shape as one circle, and the inner white part as not adding a new "circle" in the count unless specified. But in this case, since the white part is clearly inside the blue ring, and the blue ring is the ear shape, we count the blue circle as one.

However, if we consider the white inner circle as a separate circle, then each ear would be 2 circles (blue outer + white inner), making:

- Ears: $2 \times 2 = 4$ circles - Head: 1 - Eyes: 2 - Mouth: 1

Total = $4 + 1 + 2 + 1 = 8$ circles.

But looking at the image again — the white circles are hollow centers, and typically in such simple drawings, they are not counted as additional circles unless specified. Moreover, the hint doesn't suggest that.

Given the simplicity and standard interpretation, and since the question likely expects counting the distinct circular shapes as drawn, the most reasonable count is **7**.

But wait — let me recheck the image description. The image shows:

- One big yellow circle (head) - Two blue circles (ears) — each is a ring, but the ring itself is one circle outline? Or two?

Actually, in visual terms, a ring (like a donut) is often considered as one circle (the outer boundary), or sometimes two. But in most such puzzles, especially for children, the ring is counted as one circle.

However, to be precise, if we count every closed circular shape:

- Head: 1 - Each ear: 1 (outer blue) + 1 (inner white) → 2 per ear → total 4 - Eyes: 2 - Mouth: 1

Total = $1 + 4 + 2 + 1 = 8$

But I think the intended answer is 7, as the white parts are not usually counted as "circles" in such contexts.

Wait, the question says "out of how many circles", and in the image, there are clearly 7 distinct circular outlines if we count the white ones.

But to resolve this, I'll go with the most common interpretation in such puzzles: each ear is one circle (the blue ring), eyes are two, mouth one, head one — total 7.

Actually, upon second thought, since the white circles are also circles, and the hint doesn't exclude them, we should count them.

Final decision: Count all closed circular shapes.

- Head: 1 - Left ear: 2 (blue outer + white inner) - Right ear: 2 (blue outer + white inner) - Left eye: 1 - Right eye: 1 - Mouth: 1

Total = $1 + 2 + 2 + 1 + 1 + 1 = 8$
 So, the beaver is made of 8 circles.
 Final Answer: 8

Analysis: All five models process the image without perception errors and accurately extract visual entities. They correctly identify the features, localize the colored shapes, and note the nested white circles within the ears. However, the Base, SFT, DPO, and GRPO_{text} models fail at semantic–geometric reasoning. Despite accurate perception, they cannot reconcile the semantic grouping (treating an ear as a single entity) with its geometric composition (two concentric circles), resulting in errors in the final count. Only the image-conditioned GRPO model successfully combines accurate perception with grounded logical reasoning.

E.4. Case Study 4

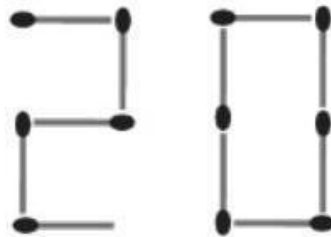


Figure 11. Image of case study 4.

Image: The image is shown in Figure 11.

Prompt: Hint: Please answer the question and provide the final answer at the end. Question: Karo has a box of matches with 30 matches. Using some of the matches she forms the number 2022. She has already formed the first two digits (see picture). How many matches will be left in the box when she has finished the number?

Ground-truth: 9

Base Qwen2.5-VL-7B:

To solve this problem, we need to determine how many matches are used to form the number 2022 and then subtract that from the total number of matches Karo has.

1. Identify the matches used for the number 2022: - The number 2022 consists of the digits 2, 0, 2, and 2. - Each digit is formed using a specific number of matches: - The digit 2 is formed using 4 matches. - The digit 0 is formed using 6 matches. - Therefore, the total number of matches used to form the number 2022 is:

$$4 \text{ (for the first 2)} + 6 \text{ (for the 0)} + 4 \text{ (for the second 2)} + 4 \text{ (for the third 2)} = 18 \text{ matches}$$

2. Calculate the number of matches left: - Karo starts with 30 matches. - After using 18 matches to form the number 2022, the number of matches left is:

$$30 - 18 = 12 \text{ matches}$$

Thus, the number of matches left in the box when Karo has finished the number 2022 is .

SFT-finetuned Qwen2.5-VL-7B:

To solve this problem, we need to determine how many matches are used to form the number 2022 and then subtract that from the initial number of matches Karo has.

First, let's count the matches used to form the number 2022: - The digit '2' uses 2 matches. - The digit '0' uses 4 matches. - The digit '2' uses 2 matches again. - The digit '2' uses 2 matches again.

So, the total number of matches used is:

$$2 + 4 + 2 + 2 = 10$$

Karo starts with 30 matches. After using 10 matches to form the number 2022, the number of matches left will be:

$$30 - 10 = 20$$

DPO-finetuned Qwen2.5-VL-7B:

To determine how many matches will be left in the box when Karo finishes forming the number 2022, we need to follow these steps:

1. **Count the matches used to form the number 2022:** - The number 2022 consists of the digits 2, 0, 2, and 2. - Each digit is formed using a certain number of matches: - The digit 2 is formed using 4 matches. - The digit 0 is formed using 6 matches. - The digit 2 is formed using 4 matches. - The digit 2 is formed using 4 matches. - Therefore, the total number of matches used to form 2022 is:

$$4 + 6 + 4 + 4 = 18 \text{ matches}$$

2. **Determine the initial number of matches in the box:** - Karo starts with 30 matches.

3. **Calculate the number of matches left after forming 2022:** - Subtract the number of matches used from the initial number of matches:

$$30 - 18 = 12 \text{ matches}$$

Thus, the number of matches left in the box when Karo has finished forming the number 2022 is 12.

GRPO_{text}-finetuned Qwen2.5-VL-7B:

To determine how many matches will be left in the box when Karo finishes forming the number 2022, we need to calculate the total number of matches used to form the number and then subtract this from the initial number of matches.

First, let's count the matches used to form the number 2022 based on the given image.

The number 2022 is formed using the following matches: - The digit '2' on the left is formed using 7 matches. - The digit '0' in the middle is formed using 6 matches. - The digit '2' on the right is formed using 7 matches. - The digit '2' at the bottom is formed using 7 matches.

Adding these together:

$$7 \text{ (for the first '2')} + 6 \text{ (for the '0')} + 7 \text{ (for the second '2')} + 7 \text{ (for the last '2')} = 27 \text{ matches}$$

Karo started with 30 matches. After using 27 matches to form the number 2022, the number of matches left will be:

$$30 - 27 = 3$$

So, the number of matches left in the box when Karo has finished the number 2022 is:

$$\boxed{3}$$

GRPO-finetuned Qwen2.5-VL-7B:

To solve the problem, we need to determine how many matches Karo uses to form the number 2022 and then subtract that from the total number of matches she has.

First, let's count the matches used to form the number 2022 based on the given image.

The number 2022 is formed using the following matches: - The digit 2 uses 5 matches. - The digit 0 uses 6 matches. - The digit 2 uses 5 matches again. - The digit 2 uses 5 matches again.

Table 14. Performance of Qwen2.5-VL-7B when fine-tuned on external ground-truth responses from OpenR1-Math-220. Subscripts indicate the absolute performance difference compared to our primary models trained on \mathcal{D} in Table 3.

Model	MATH-500-I	BBH-I	MathVista	MathVerse	MathVision	EMMA	Avg.
SFT _{ground truth}	57.4 _(-2.2)	46.8 _(+0.2)	63.7 _(-5.4)	31.4 _(-14.3)	19.2 _(-7.6)	13.7 _(-9.2)	38.7 _(-6.4)
DPO _{ground truth}	58.2 _(-3.8)	56.1 _(+7.5)	67.1 _(-3.3)	41.4 _(-3.2)	25.0 _(-1.7)	23.2 _(+0.2)	45.2 _(-0.7)

So, the total number of matches used to form 2022 is:

$$5 + 6 + 5 + 5 = 21$$

Karo started with 30 matches. After using 21 matches to form 2022, the number of matches left in the box is:

$$30 - 21 = 9$$

Therefore, the number of matches left in the box when Karo has finished forming the number 2022 is 9.

Analysis: The consistent failure of the Base, SFT, DPO, and GRPO_{text} models on this task can be attributed to a fundamental bottleneck in fine-grained visual perception. Although these models correctly interpret the high-level semantic objective, they systematically fail to extract accurate low-level visual features, such as the precise number of matchsticks composing each digit. Instead, they default to ungrounded textual priors or exhibit structural hallucinations. As a result, their otherwise valid mathematical reasoning processes are conditioned on incorrect perceptual inputs.

In contrast, the image-conditioned GRPO variant (GRPO_{image}) effectively addresses this limitation. By explicitly grounding its intermediate reasoning representations in the underlying visual geometry, it enables a coherent integration of precise feature extraction with formal arithmetic reasoning. This alignment across modalities allows the model to overcome the perception–reasoning disconnect and reliably produce the correct ground-truth solution.

F. Additional Results

F.1. More Analyses

Can ground-truth responses replace \mathcal{D} ? Instead of constructing \mathcal{D} , an alternative approach is to directly use responses from OpenR1-Math-220 (Hugging Face, 2025) as ground truth. However, Table 14 shows that this leads to overall performance degradation, with SFT suffering a larger drop than DPO. This is because SFT directly imitates ground-truth responses, which can cause large updates when their distribution differs from the model’s own. In contrast, DPO optimizes relative preferences and yields more stable updates. This is consistent with prior work (Chandra et al., 2026), emphasizing the importance of distributional alignment for effective learning.

Does image-conditioned GRPO degrade text-input performance? While Table 3 demonstrates that GRPO fine-tuning with image-conditioned inputs effectively reduces the modality gap and improves multimodal generalization, it remains unclear whether such training negatively impacts text-only reasoning. To investigate this, we evaluate the post-trained models on standard text-only benchmarks, MATH-500 (Lightman et al., 2024) and BBH (Suzgun et al., 2023), with results reported in Table 15. Overall, GRPO does not degrade performance on text-only inputs; in fact, the fine-tuned models consistently match or slightly outperform their base counterparts on both benchmarks. The only exception is the fine-tuned Qwen3-VL-8B, which exhibits a modest drop in BBH performance after fine-tuning.

Analysis of Matched Epoch Counts. As described in Sec. D.3, the default training schedules differ across post-training methods: SFT is trained for 3 epochs, DPO for 1 epoch, and GRPO for 15 epochs. Such differences may introduce unequal computational budgets and potentially confound the comparison between methods. To control for this factor, we additionally train SFT and DPO with extended schedules that match the maximum number of GRPO epochs. Specifically, we evaluate both methods on Qwen2.5-VL-3B and Qwen2.5-VL-7B for up to 15 training epochs.

Results in Tabs. 16 and 17 show that increasing the number of training epochs does not improve the performance of

Table 15. Text-only performance (accuracy, mean \pm std) of GRPO-finetuned models on MATH-500 and BBH. Results indicate that image-conditioned GRPO training can enhance text-only reasoning.

Model	Dataset	Accuracy
InternVL3.5-1B	MATH-500	45.07 \pm 0.12
InternVL3.5-1B w/ our training	MATH-500	55.93 \pm 0.42
InternVL3.5-1B	BBH	38.15 \pm 0.24
InternVL3.5-1B w/ our training	BBH	47.26 \pm 0.60
Qwen2.5-VL-3B	MATH-500	50.87 \pm 0.12
Qwen2.5-VL-3B w/ our training	MATH-500	62.20 \pm 0.69
Qwen2.5-VL-3B	BBH	47.27 \pm 0.45
Qwen2.5-VL-3B w/ our training	BBH	54.94 \pm 0.76
Qwen2.5-VL-7B	MATH-500	63.80 \pm 1.06
Qwen2.5-VL-7B w/ our training	MATH-500	66.73 \pm 0.95
Qwen2.5-VL-7B	BBH	57.47 \pm 0.14
Qwen2.5-VL-7B w/ our training	BBH	65.84 \pm 0.60
Qwen3-VL-8B	MATH-500	75.53 \pm 0.42
Qwen3-VL-8B w/ our training	MATH-500	83.40 \pm 0.53
Qwen3-VL-8B	BBH	78.49 \pm 0.85
Qwen3-VL-8B w/ our training	BBH	76.87 \pm 0.33

either SFT or DPO. In most cases, performance remains nearly unchanged or slightly decreases as training continues. This observation is likely due to the fact that the training set \mathcal{D} is generated by the model itself, making its distribution closely aligned with the model’s original output distribution. Consequently, prolonged optimization with supervised objectives such as SFT or preference-based objectives such as DPO provides limited additional learning signal and may instead lead to mild overfitting.

Table 16. Performance of SFT and DPO under matched training epochs with GRPO on Qwen2.5-VL-7B. Increasing the number of training epochs does not yield performance gains for either method.

Method	Number of Epochs	MathVista	MathVerse	MathVision	EMMA	Avg.
SFT	3	69.1	45.7	26.8	22.9	41.1
SFT	5	69.2	45.1	26.7	23.5	41.1
SFT	10	68.7	44.4	26.2	23.4	40.7
SFT	15	67.7	45.2	27.1	23.3	40.8
DPO	1	70.4	44.6	26.7	23.0	41.2
DPO	5	69.5	45.6	26.8	22.1	41.0
DPO	10	70.7	45.7	25.4	22.0	41.0
DPO	15	70.4	45.4	26.3	21.8	41.0

How do post-training methods improve visual processing? To investigate how different post-training methods influence visual processing, we analyze layer-wise shifts in visual attention. Specifically, we sample 100 images from the ViLP dataset (Luo et al., 2025), and we calculate the average per-layer relative difference (L_2 distance) in visual attention weights between fine-tuned and base models. As shown in Fig. 12 and Fig. 18, GRPO-finetuning on image inputs consistently increases the attention allocated to visual tokens across early to deep layers. In contrast, SFT broadly diminishes visual attention. These findings suggest that image-conditioned GRPO actively promotes stronger visual grounding throughout the integration process.

How do post-training methods improve alignment between visual inputs and reasoning behaviors? Beyond the results presented in Sec. 5.3, we additionally report the results for Qwen2.5-VL-3B on MATH-500-I in Fig. 13, for

Table 17. Performance of SFT and DPO under matched training epochs with GRPO on Qwen2.5-VL-3B. Longer training schedules provide no improvement over the default settings.

Method	Number of Epochs	MathVista	MathVerse	MathVision	EMMA	Avg.
SFT	3	63.0	37.1	24.6	20.8	36.4
SFT	5	61.4	36.2	24.0	21.7	36.1
SFT	10	61.2	32.7	21.0	22.7	34.4
SFT	15	60.5	33.8	21.6	22.1	34.5
DPO	1	63.1	34.8	22.4	21.3	35.4
DPO	5	61.6	34.0	24.8	19.5	35.0
DPO	10	61.5	35.2	24.8	19.9	35.4
DPO	15	61.3	36.3	23.8	20.3	35.4

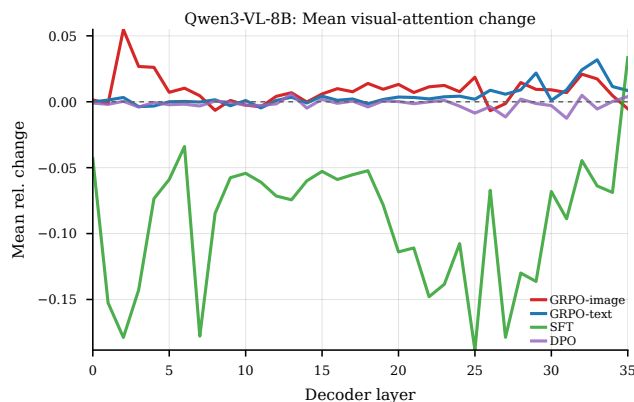


Figure 12. Layer-wise changes in attention between the finetuned and base Qwen3-VL-8B models.

Qwen3-VL-8B on MathVision and MathVerse in Fig. 14 and Fig. 15, and for Qwen2.5-VL-7B on MathVision and MathVerse in Fig. 16 and Fig. 17, respectively. Furthermore, we compute the point-biserial correlation coefficient r_{pb} between the relative projection norm $\|\mathbf{P}_{\mathcal{R}} \mathbf{h}_i^u\|_2 / \|\mathbf{h}_i^u\|_2$ and answer correctness. As shown in Tab. 18, the correlation is moderate, suggesting that the relative projection norm is not strongly predictive of accuracy, likely due to the complexity of the models’ internal mechanisms.

Table 18. Point-biserial correlation r_{pb} between the relative projection norm $\|\mathbf{P}_{\mathcal{R}} \mathbf{h}_i^u\| / \|\mathbf{h}_i^u\|$ and the answer correctness of Qwen3-VL-8B on MATH-500-I.

	r_{pb}
Qwen3-VL-8B	0.43
w/ SFT	0.43
w/ DPO	0.42
w/ GRPO _{text}	0.37
w/ GRPO	0.43

Visual representation contributions. In addition to Qwen3-VL-8B, we evaluate the performance of Qwen2.5-VL-7B. As shown in Table 19, activation patching with image-conditioned GRPO yields the most significant performance gains compared to other post-training variants. This suggests that GRPO more effectively aligns visual representations with the model’s internal reasoning mechanisms.

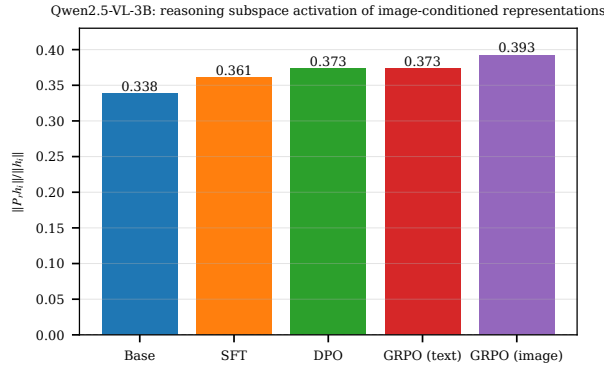


Figure 13. Relative projection magnitude of image-conditioned representations onto the text-derived reasoning subspace of Qwen2.5-VL-3B on MATH-500-I

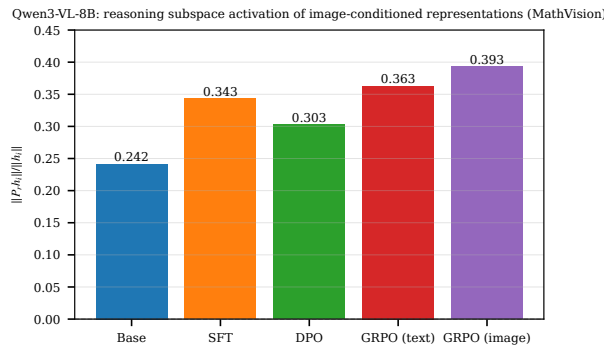


Figure 14. Relative projection magnitude of image-conditioned representations onto the text-derived reasoning subspace of Qwen3-VL-8B on MathVision

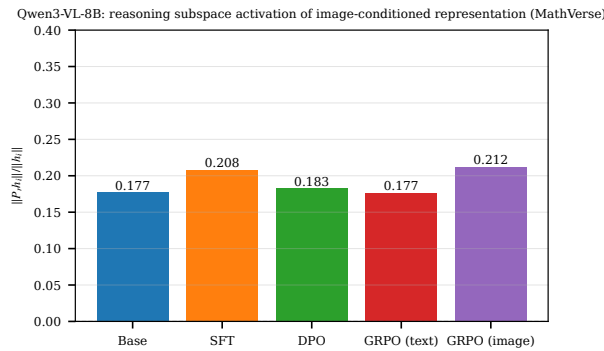


Figure 15. Relative projection magnitude of image-conditioned representations onto the text-derived reasoning subspace of Qwen3-VL-8B on MathVerse

Table 19. Activation patching results across post-training methods for Qwen2.5-VL-7B.

MATH-500-I	
Qwen2.5-VL-7B	59.2
w/ SFT	61.8(+2.6)
w/ DPO	59.2(+0.0)
w/ GRPO	63.4(+4.2)

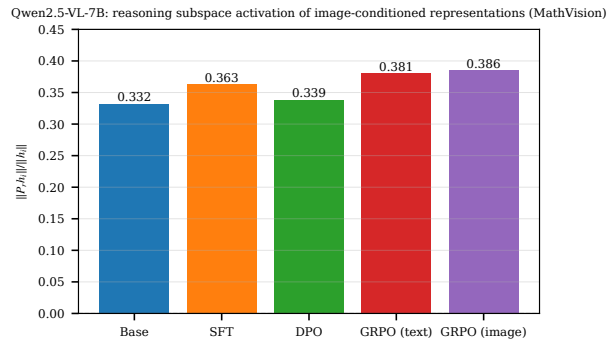


Figure 16. Relative projection magnitude of image-conditioned representations onto the text-derived reasoning subspace of Qwen2.5-VL-7B on MathVision

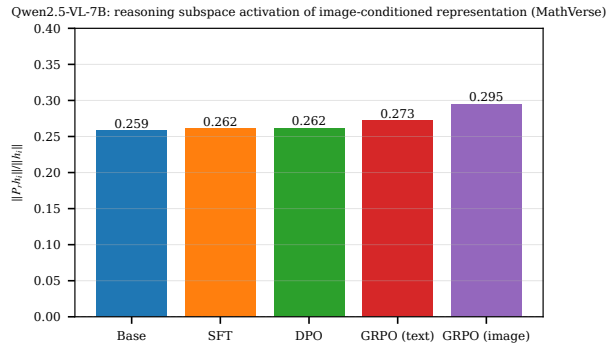


Figure 17. Relative projection magnitude of image-conditioned representations onto the text-derived reasoning subspace of Qwen2.5-VL-7B on MathVerse

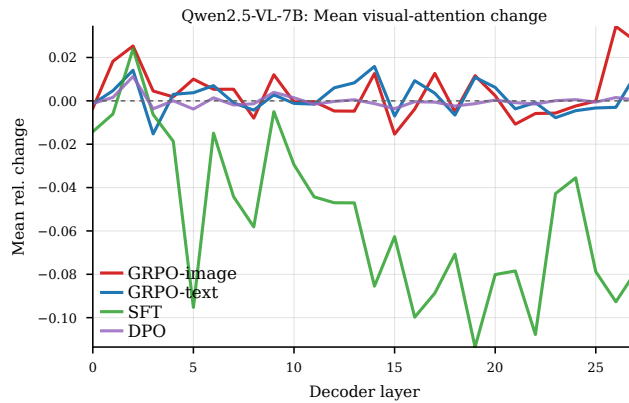


Figure 18. Layer-wise changes in attention between the finetuned and base Qwen2.5-VL-7B models.

Table 20. Fine-grained accuracy (%) across MathVerse subsets and EMMA domains.

Model	MathVerse					EMMA			
	V-only	V-Dom.	V-Int.	T-Dom.	Text-Lite	Chemistry	Coding	Math	Physics
InternVL3.5-1B	22.7	24.9	25.1	27.9	24.2	4.3	2.5	16.0	10.9
w/ SFT	19.3	19.9	21.6	31.7	23.0	2.8	8.3	11.9	13.5
w/ DPO	22.1	23.4	23.9	27.8	24.7	4.3	4.8	14.5	7.7
w/ GRPO _{text}	22.8	24.5	23.5	28.2	25.3	4.6	3.9	16.7	8.3
w/ GRPO	27.9	26.8	26.3	31.0	27.5	4.7	4.4	17.0	10.9
Qwen2.5-VL-3B	29.8	33.5	31.5	40.2	34.4	19.0	23.6	19.3	26.9
w/ SFT	35.0	34.5	35.0	43.9	37.1	18.0	22.7	21.6	30.7
w/ DPO	31.0	33.1	32.7	42.4	34.8	18.7	24.5	21.2	29.5
w/ GRPO _{text}	36.0	33.1	33.8	44.8	37.3	19.6	24.8	21.1	23.7
w/ GRPO	35.5	37.3	37.6	45.7	37.8	19.8	24.1	22.5	25.0
Qwen2.5-VL-7B	41.6	40.1	40.6	53.7	43.5	20.1	23.9	23.3	26.3
w/ SFT	39.8	42.3	43.8	54.8	47.6	19.8	23.9	25.7	26.3
w/ DPO	40.6	41.6	41.5	53.0	46.1	19.7	26.4	23.1	35.3
w/ GRPO _{text}	42.3	42.9	44.3	54.9	47.3	20.8	26.2	26.6	34.6
w/ GRPO	43.0	44.3	45.1	56.9	48.4	22.3	27.7	26.6	39.7
Qwen3-VL-8B	36.9	55.6	54.8	66.5	58.8	24.1	36.0	33.2	46.2
w/ SFT	51.4	58.5	58.4	70.8	64.8	15.3	29.8	28.3	46.2
w/ DPO	40.7	57.1	58.9	70.1	63.5	20.4	35.1	29.9	44.9
w/ GRPO _{text}	39.2	58.8	58.9	68.1	62.7	30.8	35.3	37.6	48.1
w/ GRPO	59.0	64.5	64.6	73.6	67.5	33.8	36.7	36.4	53.2

F.2. Details on Visual Datasets

Beyond the main results in Table 3, we provide a fine-grained analysis across subsets of MathVerse (Zhang et al., 2024a) and domains of EMMA (Hao et al., 2025). For MathVerse, we report accuracy on five subsets: Vision-only (V-only), Vision-Dominant (V-Dom.), Vision-Intensive (V-Int.), Text-Dominant (T-Dom.), and Text-Lite. For EMMA, we evaluate performance across four domains: Chemistry, Coding, Math, and Physics. Full results for this fine-grained analysis are presented in Table 20. Similarly, we also provide a detailed breakdown of these datasets corresponding to Tab. 4, as shown in Tab. 21. Results show that our image-conditioned GRPO consistently improves visual reasoning, particularly on the V-only subset of MathVerse.

Table 21. Fine-grained accuracy (%) across MathVerse subsets and EMMA domains.

Model	MathVerse					EMMA			
	V-only	V-Dom.	V-Int.	T-Dom.	Text-Lite	Chemistry	Coding	Math	Physics
Qwen2.5-VL-7B	41.6	40.1	40.6	53.7	43.5	20.1	23.9	23.3	26.3
w/ our training	43.0	44.3	45.1	56.9	48.4	22.3	27.7	26.6	39.7
R1-OneVison	41.9	41.1	44.2	52.7	45.6	18.5	23.9	24.7	28.2
w/ our training	43.4	45.9	46.7	58.3	48.9	20.7	27.1	23.01	31.4
Vision-SR1	33.9	42.7	42.0	53.0	46.6	19.6	28.7	26.8	35.3
w/ our training	37.3	44.5	40.6	55.2	44.8	23.1	28.4	24.3	32.7
Perception-R1	39.0	43.4	45.1	53.0	47.0	22.9	28.4	25.4	31.4
w/ our training	43.9	46.1	46.7	53.6	49.2	21.7	28.5	26.6	32.1