

Addressing Overthinking in Large Vision-Language Models via Gated Perception-Reasoning Optimization

Xingjian Diao¹, Zheyuan Liu², Chunhui Zhang¹, Weiyi Wu¹, Keyi Kong¹,
Lin Shi³, Kaize Ding⁴, Soroush Vosoughi¹, Jiang Gui¹

¹Dartmouth College, ²University of Notre Dame, ³Cornell University, ⁴Northwestern University
xingjian.diao.gr@dartmouth.edu

Abstract

Large Vision-Language Models (LVLMs) have exhibited strong reasoning capabilities through chain-of-thought mechanisms that generate step-by-step rationales. However, such slow-thinking approaches often lead to overthinking, where models produce excessively verbose responses even for simple queries, resulting in test-time inefficiency and even degraded accuracy. Prior work has attempted to mitigate this issue via adaptive reasoning strategies, but these methods largely overlook a fundamental bottleneck: visual perception failures. We argue that stable reasoning critically depends on low-level visual grounding, and that reasoning errors often originate from imperfect perception rather than insufficient deliberation. To address this limitation, we propose Gated Perception-Reasoning Optimization (GPRO), a meta-reasoning controller that dynamically routes computation among three decision paths at each generation step: a lightweight fast path, a slow perception path for re-examining visual inputs, and a slow reasoning path for internal self-reflection. To learn this distinction, we derive large-scale failure attribution supervision from approximately 790k samples, using teacher models to distinguish perceptual hallucinations from reasoning errors. We then train the controller with multi-objective reinforcement learning to optimize the trade-off between task accuracy and computational cost under uncertainty. Experiments on five benchmarks demonstrate that GPRO substantially improves both accuracy and efficiency, outperforming recent slow-thinking methods while generating significantly shorter responses.

1 Introduction

The reasoning capabilities of Large Vision-Language Models (LVLMs) have advanced considerably through chain-of-thought (CoT) and related techniques that encourage step-by-step problem decomposition (Wei et al., 2022; Kojima et al., 2022).

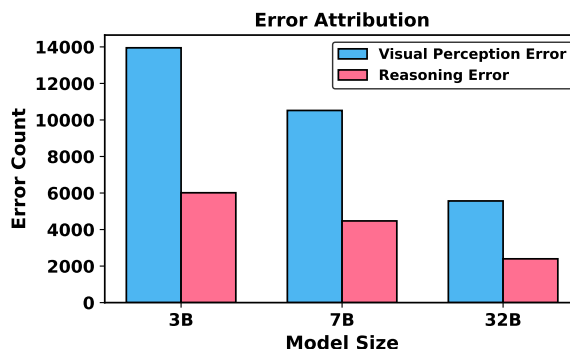


Figure 1: GPT-based error attribution of incorrect predictions from Qwen2.5-VL (Bai et al., 2025) across different model scales. Results are based on large-scale samples from multiple multimodal reasoning benchmarks, such as ViRL39k (Wang et al., 2025), MathV360K (Shi et al., 2024), and Mulberry (Yao et al., 2024a), and show that visual perception failures occur **more than twice** as often as reasoning errors.

This deliberate analytical approach enables models to solve complex mathematical and scientific problems by breaking them into manageable intermediate steps, and has proven effective across diverse domains, from arithmetic word problems to visually grounded scientific reasoning. However, applying such intensive computation uniformly across all inputs leads to a well-documented issue: overthinking. Models often generate verbose rationales for simple questions that could be answered directly, incurring test-time inefficiency and sometimes introducing errors through excessive elaboration (Kahneman, 2011).

Overthinking in LVLMs typically manifests in two patterns. First, models may produce redundant explanatory text that restates obvious visual information without advancing toward a solution (Zou et al., 2025). For instance, when asked to identify a clearly visible object’s color, a model may elaborate on the object’s shape, position, and context before stating the answer. This behavior is especially prevalent in models trained with extensive CoT data, where the training signal favors detailed ex-

planations regardless of problem complexity. Second, models may engage in unnecessary reasoning chains for problems that primarily require direct visual recognition. A simple counting question can trigger multi-step reasoning about spatial relations and numerical properties when straightforward enumeration would suffice.

Our analysis across multiple challenging multimodal reasoning benchmarks shows that **a substantial fraction of errors arise from visual perception failures rather than faulty reasoning** (as illustrated in Figure 1). These failures occur when models misinterpret key visual cues before reasoning begins and are unlikely to be corrected by additional deliberation. This observation motivates our central thesis that adaptive computation in LVLMs should account for perceptual uncertainty alongside reasoning adaptation.

A key challenge in operationalizing this insight is the absence of supervision that distinguishes perceptual failures from reasoning errors. Standard benchmarks provide only final-answer correctness, offering no signal about which cognitive stage failed. To bridge this gap, we derive large-scale failure attribution supervision by mining incorrect predictions from approximately 790k samples drawn from multiple multimodal reasoning benchmarks, such as ViRL39k (Wang et al., 2025), MathV360K (Shi et al., 2024), and Mulberry (Yao et al., 2024a), and using stronger teacher models to attribute errors to perceptual hallucinations or reasoning failures. This supervision provides targeted signals for learning when additional perceptual or reasoning computation is warranted.

Building on this foundation, we introduce Gated Perception–Reasoning Optimization (GPRO), a framework that extends adaptive reasoning with targeted visual re-analysis. At its core, a meta-reasoning controller evaluates the model’s internal state at each token generation step and routes computation through one of three specialized paths: (1) a lightweight fast path for efficient direct generation; (2) a slow perception path that re-examines visual features to resolve perceptual uncertainty; and (3) a slow reasoning path that engages self-reflection to correct logical errors. This fine-grained control allocates computational resources precisely where they are needed, avoiding unnecessary deliberation while preserving robustness on challenging inputs.

The design of GPRO is inspired by cognitive science, where human problem solving combines fast

intuitive responses with slower deliberative reasoning, alongside frequent re-inspection of perceptual inputs under uncertainty (Kahneman, 2011). By explicitly supporting both perceptual re-analysis and reasoning self-correction, GPRO more closely mirrors this flexible cognitive process. Our contributions are summarized as follows:

- We identify visual perception failures as a critical bottleneck in LVLM performance, largely overlooked by existing adaptive reasoning methods.
- We derive large-scale failure attribution supervision from approximately 790k samples to distinguish perceptual hallucinations from reasoning errors.
- We propose the GPRO framework, which dynamically allocates computation between perception and reasoning at token-level granularity via a novel meta-reasoning controller.
- We demonstrate through extensive experiments on five benchmarks that GPRO models achieve strong accuracy with substantially reduced response lengths, establishing an effective approach for adaptive multimodal reasoning.

2 Related Work

2.1 Reasoning in Vision-Language Models

CoT prompting marked a significant advance in enhancing reasoning capabilities of large models (Wei et al., 2022). By generating step-by-step rationales before final answers, models can decompose complex problems into intermediate steps amenable to sequential processing. This paradigm has been extended to multimodal settings, where LVLMs generate interleaved text and visual analysis to solve reasoning problems (Liu et al., 2024; Bai et al., 2025). The key insight is that explicit intermediate steps allow models to leverage their language modeling capabilities for multi-hop reasoning that would be difficult in a single forward pass.

Subsequent work has refined this process through various mechanisms. Self-correction approaches (Madaan et al., 2024) enable models to identify and fix errors in their own reasoning paths through iterative refinement (Zhou et al., 2025). Tree of Thoughts (Yao et al., 2024b) explores multiple reasoning paths simultaneously, enabling backtracking and alternative solution exploration when initial approaches fail. These methods have

pushed performance boundaries on complex reasoning tasks but often incur substantial computational overhead.

Recent work has advanced reasoning capabilities and post-training adaptation in large language models (Comanici et al., 2025; Diao et al., 2025b; Yuan et al., 2025, 2026). Mulberry (Yao et al., 2024a) achieves o1-like reasoning through collective Monte Carlo tree search, enabling systematic exploration of solution spaces. Virgo (Du et al., 2025) explores reproducing slow-thinking mechanisms in multimodal settings through careful data curation and training procedures. LMM-R1 (Peng et al., 2025) employs two-stage rule-based reinforcement learning to enhance reasoning in smaller models. These approaches represent important progress but share a common limitation: they focus primarily on reasoning depth without explicitly addressing perceptual accuracy.

2.2 Efficiency and Adaptive Computation

The computational cost of exhaustive reasoning has spurred research on adaptive computation strategies. Mixture-of-Experts architectures (Shazeer et al., 2017; Fedus et al., 2022) selectively activate parameter subsets based on input characteristics, enabling capacity scaling without proportional cost increases for every input. The sparse activation pattern provides a natural mechanism for adaptive computation, as different experts can specialize in different input types or reasoning patterns.

Early-exit mechanisms and model cascades represent another approach, routing simpler queries through lightweight processing paths while reserving full model capacity for complex inputs. These methods recognize that not all inputs require the same computational depth and that significant efficiency gains are possible through intelligent routing. Recent work has explored adaptive reasoning strategies specifically for vision-language models. The FAST framework (Xiao et al., 2025) investigates how response length and data distribution affect LVLM performance, developing methods to dynamically adjust reasoning depth based on problem characteristics. Vision-R1 (Huang et al., 2025) incentivizes reasoning capability through reinforcement learning on curated multimodal CoT data. Curr-ReFT (Deng et al., 2025a) proposes curriculum reinforcement finetuning to address training bottlenecks in smaller VLMs. Our work extends this direction by explicitly incorporating perceptual uncertainty alongside reasoning adaptation.

2.3 Reinforcement Learning for Language Models

Reinforcement learning has become central to training adaptive computation frameworks and aligning model behavior with desired objectives (Ma et al., 2026; Hou et al., 2025; Qing et al., 2026; Diao et al., 2025a). Proximal Policy Optimization (Schulman et al., 2017) and related algorithms enable models to optimize policies balancing multiple objectives such as accuracy and efficiency. Reinforcement Learning from Human Feedback (Ouyang et al., 2022) has proven effective for aligning model behavior with human preferences, demonstrating that RL can shape complex behaviors difficult to specify through supervised learning alone.

Recent multimodal reasoning models have applied RL-based approaches with notable success. R1-OneVision (Yang et al., 2025) advances generalized multimodal reasoning through cross-modal formalization trained with reinforcement learning. MM-Eureka (Meng et al., 2025) explores rule-based reinforcement learning for multimodal reasoning at scale. OpenVLThinker (Deng et al., 2025b) demonstrates that iterative self-improvement through alternating supervised finetuning and reinforcement learning can yield sophisticated CoT reasoning capabilities.

Our GPRO framework employs a similar RL-based approach but introduces a more sophisticated state representation incorporating visual perception signals alongside reasoning uncertainty. This enables the controller to trigger dedicated visual re-analysis when perceptual uncertainty is high, a capability absent from prior adaptive reasoning methods that focus solely on reasoning depth.

3 Method

We introduce Gated Perception-Reasoning Optimization (GPRO), a framework that augments standard LVLMs with dynamic, fine-grained control over computational resource allocation. The key innovation is a Gated Perception-Reasoning (GPR) module that replaces selected feed-forward layers in the decoder, enabling token-level decisions about whether to invoke additional perception or reasoning computation.

3.1 Gated Perception-Reasoning Architecture

The GPR module serves as a lightweight replacement for standard FFN layers, strategically inserted at alternating positions in the Transformer decoder

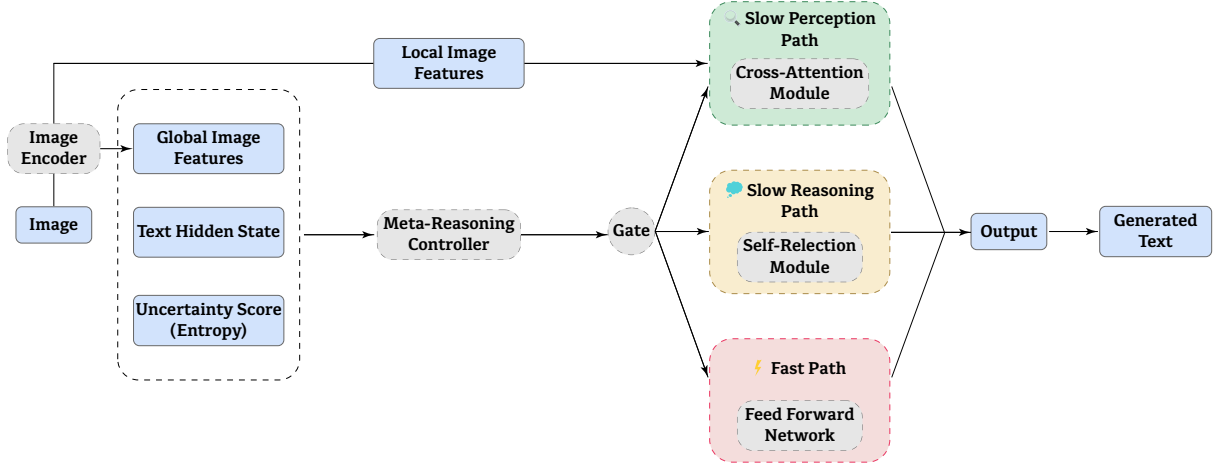


Figure 2: GPRO architecture overview. The meta-reasoning controller receives text hidden states, uncertainty scores, and global image features to route among Fast Path (FFN), Slow Perception Path (Cross-Attention), and Slow Reasoning Path (Self-Reflection).

to balance adaptive computation with base model capabilities. This alternating pattern ensures the model maintains its foundational representational power while gaining flexibility to dynamically adjust its computational strategy. Each GPR module comprises a meta-reasoning controller and three computational paths, as illustrated in Figure 2.

3.1.1 Meta-Reasoning Controller

The controller is a compact 2-layer Transformer that determines path selection for each generated token. Its lightweight design ensures minimal overhead while providing sufficient capacity for nuanced decision-making. At timestep t , the controller receives a state vector s_t that encapsulates the model’s current cognitive state through three complementary signals.

The first signal is the current hidden state h_t from the main decoder, encoding the semantic context of generated text. This captures what the model knows at the current generation step and provides information about the reasoning trajectory so far. The second signal is an uncertainty score U_t computed as predictive entropy over output logits, quantifying the model’s confidence in its next token prediction. High entropy indicates the model is uncertain about how to proceed, suggesting additional computation may be beneficial. The third signal is global image features V_g summarizing the visual context, providing the controller with information about the visual input’s complexity and content.

Based on the concatenated state $s_t = [h_t; U_t; V_g]$, the controller outputs a discrete action $a_t \in \{\text{fast, perception, reasoning}\}$ selecting one of

three computational paths.

3.1.2 Computational Paths

The three paths address distinct computational needs arising during generation:

Fast Path. The default low-cost option utilizing the original FFN layer from the base model. This path is selected when the model is confident in both its perceptual understanding and reasoning trajectory:

$$\text{Fast}(h_t) = \text{GELU}(h_t W_1 + b_1) W_2 + b_2. \quad (1)$$

The fast path preserves the base model’s efficient generation capability for straightforward cases.

Slow Perception Path. Activated when the controller detects high perceptual uncertainty, this path uses the current hidden state as a query for cross-attention over visual features:

$$\text{Perc}(h_t, V) = \text{CrossAttn}(h_t, V, V). \quad (2)$$

This enables the model to re-examine the image, focusing on details relevant to current generation. The cross-attention mechanism allows targeted retrieval of visual information that may have been overlooked or misinterpreted in initial processing.

Slow Reasoning Path. Engaged when logical uncertainty is high, this path passes the current hidden state and recent context to a meta-reasoning module:

$$\text{Reas}(h_t, H_{<t}) = \text{MetaTrans}(h_t, H_{<t}). \quad (3)$$

This supports internal self-reflection, allowing the model to reconsider its reasoning trajectory

without generating additional output tokens. The reasoning path can identify and correct logical errors before they propagate to subsequent generation steps.

3.2 Multi-Objective Training

The meta-reasoning controller is trained via PPO-based reinforcement learning with a multi-objective reward function designed to balance competing desiderata:

$$R(\tau) = R_{\text{task}} + \alpha_c R_{\text{cost}} + \alpha_l R_{\text{cal}}, \quad (4)$$

where α_c and α_l are weighting coefficients controlling the trade-off between accuracy and efficiency.

Task Reward. A sparse signal providing the primary learning signal: $R_{\text{task}} = +1$ for correct final answers, 0 otherwise. This ensures the model prioritizes accuracy as its primary objective.

Cost Reward. Penalizes activation of computationally expensive slow paths:

$$R_{\text{cost}} = - \sum_t (c_p \mathbb{1}_{[a_t=p]} + c_r \mathbb{1}_{[a_t=r]}), \quad (5)$$

where c_p and c_r are cost coefficients for perception and reasoning paths respectively. This encourages reliance on the fast path when additional computation is unnecessary.

Calibration Reward. Ensures uncertainty scores reliably indicate when slow paths are needed:

$$R_{\text{cal}} = - \sum_{t \in \mathcal{E}} (1 - U_t) - \sum_{t \in \mathcal{C}} U_t, \quad (6)$$

where \mathcal{E} and \mathcal{C} denote tokens leading to incorrect and correct answers respectively. This reward encourages the model to be uncertain when it should be (before errors) and confident when it should be (before correct outputs), enabling the controller to make well-informed decisions.

3.3 Training Data Construction

Effective training of the controller requires data that exposes both perceptual and reasoning failure modes. We construct such data through a systematic three-step procedure.

First, we perform failure case mining by running Qwen2.5-VL on a combined dataset of approximately 790K samples drawn from multiple multimodal reasoning benchmarks, such as ViRL39k (Wang et al., 2025), MathV360K (Shi et al., 2024), and Mulberry (Yao et al., 2024a), collecting all

instances where the model produces incorrect answers. This provides a rich set of failure cases spanning diverse visual reasoning challenges.

Second, we conduct failure attribution using GPT-4 to categorize each failure as either a visual perception failure or a reasoning error propagation. The categorization is based on analysis of the question, image, incorrect answer, and ground truth. Visual perception failures are identified when the model’s error can be traced to a misinterpretation of visual elements, while reasoning errors are identified when visual understanding appears correct but the logical steps are flawed. We provide illustrative examples of both failure types in Appendix C.

Third, we construct a training curriculum from the labeled data, oversampling difficult examples to ensure the controller encounters sufficient instances of both failure types (Jia et al., 2025). For the 3B model, we curate approximately 40K training samples, while the 7B model uses approximately 30K samples. The smaller sample size for the larger model reflects its stronger base capabilities requiring less corrective training.

4 Experiments

We evaluate GPRO on five challenging benchmarks to address three questions: (1) How does GPRO compare to state-of-the-art methods in accuracy and efficiency? (2) What are the contributions of individual components? (3) How does the model behave qualitatively?

4.1 Experimental Setup

Baselines. We compare against three categories of baselines. Closed-source models include GPT-4o (Hurst et al., 2024), Claude-3.5 Sonnet (Anthropic, 2024), and Qwen-VL-Max (Bai et al., 2023), representing current commercial capabilities. Base models include Qwen2-VL-7B (Wang et al., 2024b) and Qwen2.5-VL (Bai et al., 2025) at 3B and 7B scales. Recent slow-thinking methods include Mulberry (Yao et al., 2024a), Virgo (Du et al., 2025), Curr-ReFT (Deng et al., 2025a), LMM-R1 (Peng et al., 2025), MM-R1 (Liang et al., 2025), Vision-R1 (Huang et al., 2025), R1-OneVision (Yang et al., 2025), OpenVLThinker (Deng et al., 2025b), and FAST (Xiao et al., 2025).

Benchmarks. We evaluate on five benchmarks requiring sophisticated visual and mathematical reasoning: MathVision (Wang et al., 2024a) for geometric and mathematical problems in visual

Method	MathVision		MathVerse		MathVista		DynaMath		MM-Vet	
	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.	Acc.	Len.
<i>Closed-Source Models</i>										
GPT-4o (Hurst et al., 2024)	30.4	–	49.9	–	63.8	–	<u>63.7</u>	–	80.8	–
Claude-3.5 Sonnet (Anthropic, 2024)	<u>37.9</u>	–	46.3	–	<u>67.7</u>	–	64.8	–	68.7	–
Qwen-VL-Max (Bai et al., 2023)	39.3	–	<u>47.3</u>	–	74.2	–	–	–	<u>73.2</u>	–
<i>Base Qwen2-VL-7B</i>										
Qwen2-VL-7B (Wang et al., 2024b)	18.8	<u>443.0</u>	31.9	<u>388.9</u>	<u>58.2</u>	265.9	<u>39.8</u>	<u>298.4</u>	62.0	132.5
Mulberry (Yao et al., 2024a)	<u>23.4</u>	349.2	39.5	364.3	62.1	<u>275.0</u>	46.8	273.3	<u>43.9</u>	<u>218.3</u>
Virgo (Du et al., 2025)	24.0	–	<u>36.7</u>	–	–	–	–	–	–	–
<i>Base Qwen2.5-VL-3B</i>										
Qwen2.5-VL-3B (Bai et al., 2025)	21.2	450.6	34.6	362.3	62.3	212.9	48.2	270.9	61.3	138.8
Curr-ReFT (Deng et al., 2025a)	20.1	240.1	36.3	121.6	61.9	95.9	43.8	146.4	62.0	117.6
LMM-R1 (Peng et al., 2025)	25.2	447.8	41.8	423.9	63.2	245.0	53.1	341.6	65.9	166.3
FAST-3B (Xiao et al., 2025)	<u>26.8</u>	323.5	<u>43.0</u>	286.3	<u>66.2</u>	158.7	<u>54.4</u>	213.7	64.0	<u>112.7</u>
GPRO-3B (Ours)	27.1	<u>298.6</u>	44.2	<u>265.4</u>	66.8	<u>145.2</u>	55.1	<u>195.3</u>	<u>65.2</u>	108.4
<i>Base Qwen2.5-VL-7B</i>										
Qwen2.5-VL-7B (Bai et al., 2025)	25.6	443.0	46.9	388.9	68.2	189.1	58.0	273.3	67.1	132.5
MM-R1 (Liang et al., 2025)	30.2	324.6	<u>49.8</u>	283.9	71.0	185.6	57.5	254.2	70.6	137.9
Vision-R1 (Huang et al., 2025)	–	–	52.4	–	73.5	–	–	–	–	–
R1-OneVision (Yang et al., 2025)	29.9	692.8	46.4	631.5	64.1	402.5	53.5	560.6	71.6	440.7
OpenVLThinker (Deng et al., 2025b)	29.6	457.2	47.9	398.4	70.2	305.7	57.4	382.1	68.5	312.7
FAST-7B (Xiao et al., 2025)	30.6	204.8	50.6	201.0	<u>73.8</u>	<u>120.7</u>	<u>58.3</u>	<u>164.8</u>	71.2	114.1
GPRO-7B (Ours)	31.2	195.6	48.7	188.4	74.2	115.3	59.2	158.7	70.9	<u>118.8</u>

Table 1: Main results on five reasoning benchmarks. We report accuracy (%) and average response length (tokens). Response lengths are measured using Qwen2.5-VL’s tokenizer. **Bold** results indicate the best performance among all methods, while underlined results indicate the second-best performance among all methods.

contexts; MathVerse (Zhang et al., 2024) for mathematical reasoning with complex visual diagrams; MathVista (Lu et al., 2024) covering diverse mathematical reasoning grounded in visual information; DynaMath (Zou et al., 2024) for dynamic mathematical reasoning; and MM-Vet (Yu et al., 2023) for integrated multimodal capabilities.

Implementation. GPRO models are built on Qwen2.5-VL with GPR modules replacing alternate FFN layers. Training used 8 NVIDIA H100 GPUs with batch size 512, 8 rollouts per question, for 10 epochs (approximately 600 GPU hours). Reward weights were $\alpha_c = 0.1$, $\alpha_l = 0.2$, with learning rate 1×10^{-5} and cosine decay schedule.

4.2 Main Results

Table 1 presents a comprehensive comparison across five multimodal reasoning benchmarks. We report both accuracy (%) and average response length (tokens) to facilitate a dual analysis of performance and computational efficiency.

The empirical results in Table 1 reveal several critical insights regarding the interplay between reasoning depth and computational efficiency.

Optimizing the Efficiency-Accuracy Frontier.

First and foremost, GPRO consistently redefines

the Pareto frontier for multimodal reasoning. Unlike prior methods that trade inference speed for accuracy, our approach achieves superior performance while dramatically curtailing token generation. On MathVerse, GPRO-7B improves accuracy by 1.8% over the base Qwen2.5-VL-7B while reducing the average response length by 51.5% (from 388.9 to 188.4 tokens). Similarly, on MathVista, we observe a 6.0% accuracy gain alongside a 39% reduction in tokens. This confirms our core hypothesis: model performance is not strictly proportional to generation length, and intelligent resource allocation can mitigate the overthinking phenomenon inherent in standard CoT processes.

Critique of Unconditional Long-Context Reasoning.

A comparison with recent long-CoT distillation models, such as R1-OneVision and OpenVLThinker, highlights the limitations of unconditional slow thinking. R1-OneVision exhibits extreme token consumption (e.g., 692.8 tokens on MathVision) yet often underperforms our method (29.9% vs. 31.2% for GPRO-7B). This suggests that forcing models to generate extensive reasoning paths for every query introduces noise or hallucination rather than clarity. GPRO’s selective activation mechanism acts as a surgical intervention, allocating computational budget only when

necessary, thereby achieving a $\sim 3.5\times$ reduction in inference cost compared to R1-OneVision while surpassing its accuracy.

Competitiveness with Proprietary Giants. Notably, GPRO-7B demonstrates surprising resilience against closed-source models orders of magnitude larger. On MathVision, GPRO-7B (31.2%) outperforms GPT-4o (30.4%), and on MathVista, it matches the performance of Qwen-VL-Max (74.2%). This result is significant as it indicates that smaller, open-weights models, when equipped with efficient meta-reasoning strategies, can bridge the gap with proprietary SOTA models in specific reasoning-heavy domains.

Scalability Across Model Sizes. Finally, the benefits of GPRO are robust across model scales. GPRO-3B outperforms the strong FAST-3B baseline on 4 out of 5 benchmarks and substantially improves upon the base Qwen2.5-VL-3B (e.g., +9.6% on MathVerse). This scalability suggests that the perception-reasoning decomposition is a fundamental improvement applicable to various architectures, rather than a parameter-scale-dependent optimization. The consistent gains in both the 3B and 7B settings validate the universality of our proposed controller mechanism.

4.3 Ablation Study

To deconstruct the efficacy of GPRO and isolate the impact of its constituent modules, we conducted a component-wise ablation study. Table 2 summarizes the performance contributions of the Slow Perception path, Slow Reasoning path, and the Calibration Reward mechanism.

Configuration	MathVision	MathVerse
Full GPRO-7B	31.2	48.7
w/o Slow Perception	27.8 (-3.4)	44.3 (-4.4)
w/o Slow Reasoning	29.5 (-1.7)	47.0 (-1.7)
w/o Calibration Reward	28.9 (-2.3)	46.2 (-2.5)

Table 2: Ablation study results on MathVision and MathVerse. Numbers in parentheses indicate the absolute accuracy drop compared to the full model configuration.

Dominance of Visual Grounding. The most significant performance degradation occurs upon removing the *Slow Perception Path*, resulting in a sharp decline of 3.4% on MathVision and 4.4% on MathVerse. This finding empirically validates our core hypothesis: visual hallucinations serve as the primary bottleneck in multimodal reasoning.

When the model is deprived of the mechanism to re-examine visual features, initial perceptual errors (e.g., misreading axis scales or misidentifying geometric primitives) propagate irreversibly through the reasoning chain. The disparity between the perception drop (-4.4%) and the reasoning drop (-1.7%) highlights that current VLMs suffer more from "garbage-in" perceptual failures than from logical deduction errors.

Role of Reflexive Reasoning. The *Slow Reasoning Path*, while less critical than perception, contributes a consistent improvement ($\sim 1.7\%$ across benchmarks). This module effectively mitigates logic-level errors by enabling self-correction during complex multi-step derivations. However, its lower relative impact suggests that the base Qwen2.5 model possesses robust inherent reasoning capabilities, and the marginal gain from "thinking longer" is diminishing unless grounded in accurate visual data.

Importance of Uncertainty Calibration. Ablating the *Calibration Reward* leads to a substantial performance drop (2.3%–2.5%), underscoring the necessity of alignment between the controller’s confidence and task difficulty. Without this reward signal, the meta-controller fails to learn an optimal switching policy, often collapsing into a mode of either indiscriminate fast-path usage (under-thinking) or wasteful slow-path activation (over-thinking). The calibration reward effectively regularizes the decision boundary, ensuring that computational overhead is incurred only when model uncertainty is genuinely indicative of potential failure.

Resource Allocation Dynamics. To further understand the controller’s behavior, we analyze the path activation distribution on the MathVision test set. As illustrated in the distribution analysis, GPRO-7B activates the Fast Path for 73% of tokens, allocating the Slow Perception Path (17%) and Slow Reasoning Path (10%) sparsely. This highly skewed distribution confirms that the model has learned a resource-efficient policy, treating slow thinking as a surgical intervention rather than a default mode. Qualitative inspection reveals that Slow Perception is triggered predominantly on high-frequency visual tokens (e.g., numbers in tables, coordinates, legend texts), whereas Slow Reasoning activates during transitional logical connectives (e.g., "therefore", "implies"), demonstrating the semantic awareness of our learned controller.

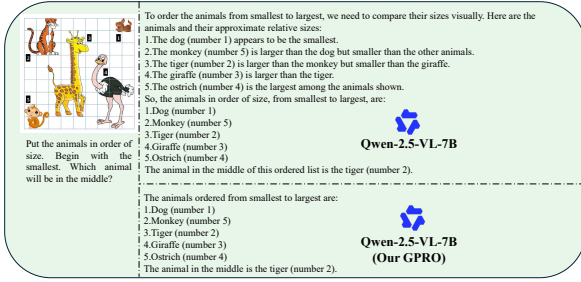


Figure 3: Case Study 1: Animal size ordering. The baseline produces verbose step-by-step comparisons, while GPRO generates a concise direct answer.

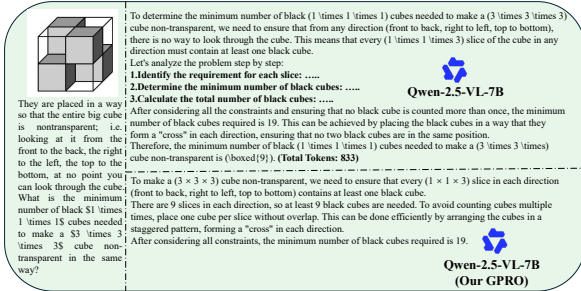


Figure 4: Case Study 2: Spatial reasoning with cubes. The baseline generates 833 tokens of detailed analysis, while GPRO produces an efficient solution.

4.4 Case Study

We present two case studies to illustrate how GPRO mitigates overthinking while preserving accuracy through adaptive resource allocation.

Case 1: Visual Ordering Task. Figure 3 depicts a task that requires ordering animals by size and identifying the middle one. The baseline Qwen2.5-VL-7B produces lengthy explanations for each animal’s relative size (e.g., “the dog appears to be the smallest,” “the monkey is larger than the dog but smaller than the other animals”) before arriving at the answer. This behavior exemplifies overthinking, where detailed reasoning is applied to a task that primarily depends on visual perception. In contrast, GPRO identifies this as a straightforward visual task and predominantly relies on the Fast Path, with selective activation of the Slow Perception Path to verify size relationships when ambiguity arises. As a result, GPRO produces a correct and concise answer, identifying the tiger as the middle animal while generating substantially fewer tokens.

Case 2: Spatial Reasoning. Figure 4 presents a more complex spatial reasoning task that involves counting the number of black cubes required to make a larger cube opaque from all viewing directions. The baseline generates 833 tokens of step-by-step analysis, explicitly enumerating constraints

for each slice and direction before reaching a conclusion. GPRO, in contrast, selectively activates the Slow Perception Path to accurately interpret the 3D structure shown in the image, followed by the Slow Reasoning Path to verify the counting logic. The model correctly identifies the key constraint that each slice along every direction must contain at least one black cube, and arrives at the correct answer with significantly fewer tokens. This example illustrates how GPRO coordinates perception and reasoning by invoking additional computation only when needed, leading to a correct solution with substantially reduced generation length.

Together, these cases demonstrate the core advantage of GPRO: dynamically allocating computation based on task demands, avoiding unnecessary verbosity for simple inputs while engaging deeper analysis only when required.

5 Conclusion

We introduce Gated Perception–Reasoning Optimization (GPRO), a gated adaptive computation framework for LVLMs. This work is motivated by the observation that visual perception failures constitute a critical yet underexplored bottleneck in LVLm performance. Such failures often co-occur with overthinking and limit the effectiveness of deeper reasoning alone. GPRO leverages failure attribution supervision to distinguish perceptual errors from reasoning errors, and uses this signal to guide a meta-reasoning controller that routes token-level generation among fast execution, visual re-examination, and reasoning refinement. Experiments on five challenging benchmarks show that GPRO improves accuracy while consistently reducing response length, demonstrating an effective approach to adaptive multimodal reasoning.

6 Limitations

While GPRO demonstrates strong effectiveness on vision–language reasoning benchmarks, several limitations remain. First, our failure attribution supervision relies on GPT-based labeling, and its quality therefore depends on the reliability of the teacher model’s judgments. Although this labeling strategy enables large-scale supervision, GPT-generated attributions may still contain occasional ambiguities, especially in cases where perception and reasoning errors are closely intertwined. Second, GPRO adopts a discrete routing scheme over a fixed set of computation paths. Exploring finer-

grained or continuous control over perception and reasoning interventions may offer additional flexibility, which we leave to future work.

Ethical Considerations

All experiments presented in this study were conducted using publicly available datasets and models licensed for academic research purposes. To the best of our knowledge, this work does not present any ethical concerns.

Acknowledgment

This study is supported by the Department of Defense under Grant No. HT9425-23-1-0267 and in part by the National Science Foundation under Grant No. 2452367.

References

- Anthropic. 2024. [The claude 3 model family: Opus, sonnet, haiku](#).
- Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. [Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond](#). *arXiv preprint arXiv:2308.12966*.
- Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. [Qwen2. 5-vl technical report](#). *arXiv preprint arXiv:2502.13923*.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Naveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *arXiv preprint arXiv:2507.06261*.
- Huilin Deng, Hongchen Shi, Yicheng Zhu, Junfeng Yin, Shen Zheng, Zilong Liu, and 1 others. 2025a. [Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning](#). *arXiv preprint arXiv:2503.07065*.
- Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. 2025b. [Openvlthinker: An early exploration to complex vision-language reasoning via iterative self-improvement](#). *arXiv preprint arXiv:2503.17352*.
- Xingjian Diao, Chunhui Zhang, Keyi Kong, Weiyi Wu, Chiyu Ma, Zhongyu Ouyang, Peijun Qing, Soroush Vosoughi, and Jiang Gui. 2025a. [Soundmind: RL-incentivized logic reasoning for audio-language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Xingjian Diao, Chunhui Zhang, Weiyi Wu, Zhongyu Ouyang, Peijun Qing, Ming Cheng, Soroush Vosoughi, and Jiang Gui. 2025b. [Temporal working memory: Query-guided segment refinement for enhanced multimodal understanding](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*.
- Yifan Du, Zikang Li, Yifan Ding, Junyu Feng, Xinyu Zuo, Xudong Zhao, Longteng Tian, Rui Zheng, Zhaopeng Wen, Minlie Huang, and 1 others. 2025. [Virgo: A preliminary exploration on reproducing o1-like mllm](#). *arXiv preprint arXiv:2501.01904*.
- William Fedus, Barret Zoph, and Noam Shazeer. 2022. [Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity](#). *Journal of Machine Learning Research*.
- Xinhai Hou, Shaoyuan Xu, Manan Biyani, Moyan Li, Jia Liu, Todd C Hollon, and Bryan Wang. 2025. [Codev: Code with images for faithful visual reasoning via tool-aware policy optimization](#). *arXiv preprint arXiv:2511.19661*.
- Wenyi Huang, Bohan Jia, Zijie Zhai, Zhiqi Cai, Shaobin Gao, Zhi Wang, Luoyi Chen, and 1 others. 2025. [Vision-r1: Incentivizing reasoning capability in multimodal large language models](#). *arXiv preprint arXiv:2503.06749*.
- Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. [Gpt-4o system card](#). *arXiv preprint arXiv:2410.21276*.
- Yaning Jia, Chunhui Zhang, Xingjian Diao, Xiangchi Yuan, Zhongyu Ouyang, Chiyu Ma, and Soroush Vosoughi. 2025. [What makes a good curriculum? disentangling the effects of data ordering on llm mathematical reasoning](#). *arXiv preprint arXiv:2510.19099*.
- Daniel Kahneman. 2011. *Thinking, Fast and Slow*. Farrar, Straus and Giroux.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large language models are zero-shot reasoners](#). In *Advances in Neural Information Processing Systems*.
- Qiankun Liang, Yuxuan Chen, Ziming Liu, Shaobin Huang, and 1 others. 2025. [Mm-r1: Unleashing the power of unified multimodal large language models for personalized image generation](#). *arXiv preprint arXiv:2508.11433*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. [Visual instruction tuning](#). In *Advances in Neural Information Processing Systems*.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. [Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts](#). *arXiv preprint arXiv:2310.02255*.

- Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. 2026. [Fipo: Eliciting deep reasoning with future-kl influenced policy optimization](#). *arXiv preprint arXiv:2603.19835*.
- Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegrefe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2024. [Self-refine: Iterative refinement with self-feedback](#). *Advances in Neural Information Processing Systems*.
- Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, and 1 others. 2025. [Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning](#). *arXiv preprint arXiv:2503.07365*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. [Training language models to follow instructions with human feedback](#). In *Advances in Neural Information Processing Systems*.
- Yingzhe Peng, Gongrui Liu, Cheng Zhang, Haoran Xu, Jiawei Liu, Xin Li, Ning Ding, Yu Qiao, Jie Liu, and 1 others. 2025. [Lmm-r1: Empowering 3b lmm with strong reasoning abilities through two-stage rule-based rl](#). *arXiv preprint arXiv:2503.07536*.
- Peijun Qing, Puneet Mathur, Nedim Lipka, Varun Manjunatha, Ryan Rossi, Franck Dernoncourt, Saeed Hassanpour, and Soroush Vosoughi. 2026. [Cluster-r1: Large reasoning models are instruction-following clustering agents](#). *arXiv preprint arXiv:2603.23518*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *arXiv preprint arXiv:1707.06347*.
- Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). *arXiv preprint arXiv:1701.06538*.
- Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. [Math-llava: Bootstrapping mathematical reasoning for multimodal large language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024*.
- Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhui Chen. 2025. [Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning](#). *arXiv preprint arXiv:2504.08837*.
- Ke Wang, Junting Wang, Jingyi Shao, Zimu Shi, Wenya Guan, Weijie Liu, Xuefeng Wang, and Rui Zhong. 2024a. [Measuring multimodal mathematical reasoning with math-vision dataset](#). *arXiv preprint arXiv:2402.14804*.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024b. [Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution](#). *arXiv preprint arXiv:2409.12191*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.
- Wenyi Xiao, Lin Gan, Wangui Dai, Wenhao He, Ziwei Huang, Haoyuan Li, Yichao Zhu, Wentao Zhu, Ruofan Jiang, Weijia Shao, and 1 others. 2025. [Fast-slow thinking grpo for large vision-language model reasoning](#). *arXiv preprint arXiv:2504.18458*.
- Yi Yang, Xiaocui Yin, Shuo Wang, Yifu Chen, Yingying Li, Wenjie Wang, Yuhao Zhong, Jiaqi Deng, and 1 others. 2025. [R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization](#). *arXiv preprint arXiv:2503.10615*.
- Huanjin Yao, Jiaying Wu, Wenhao Wang, Jingyi Dong, Yibo Liang, Shunyu Zhu, Yingjie Wang, Yuxin Tan, Haoran Liu, Jianye Wang, and 1 others. 2024a. [Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search](#). *arXiv preprint arXiv:2412.18319*.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2024b. [Tree of thoughts: Deliberate problem solving with large language models](#). In *Advances in Neural Information Processing Systems*.
- Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. [Mm-vet: Evaluating large multimodal models for integrated capabilities](#). *arXiv preprint arXiv:2308.02490*.
- Xiangchi Yuan, Xiang Chen, Tong Yu, Dachuan Shi, Can Jin, Wenke Lee, and Saayan Mitra. 2025. [Mitigating forgetting between supervised and reinforcement learning yields stronger reasoners](#). *arXiv preprint arXiv:2510.04454*.
- Xiangchi Yuan, Dachuan Shi, Chunhui Zhang, Zheyuan Liu, Shenglong Yao, Soroush Vosoughi, and Wenke Lee. 2026. [Behavior knowledge merge in reinforced agentic models](#). *arXiv preprint arXiv:2601.13572*.
- Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and 1 others. 2024. [Math-verse: Does your multi-modal llm truly see the diagrams in visual math problems?](#) *arXiv preprint arXiv:2403.14624*.

Yefan Zhou, Austin Xu, Yilun Zhou, Janvijay Singh, Jiang Gui, and Shafiq Joty. 2025. [Variation in verification: Understanding verification dynamics in large language models](#). *arXiv preprint arXiv:2509.17995*.

Chengke Zou, Xingang Zhang, Rui Zhao, Wei Li, Junchi Guo, and Wentao Zhu. 2024. [Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models](#). *arXiv preprint arXiv:2411.00836*.

Xin Zou, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Kening Zheng, Sirui Huang, Junkai Chen, Peijie Jiang, Jia Liu, Chang Tang, and Xuming Hu. 2025. [Look twice before you answer: Memory-space visual re-tracing for hallucination mitigation in multimodal large language models](#). In *International Conference on Machine Learning*.

A Benchmarks

- **MathVision** (Wang et al., 2024a): MathVision is a multimodal mathematical reasoning benchmark composed of problems collected from real-world math competitions. It spans 16 mathematical disciplines and five difficulty levels, covering diverse visual reasoning scenarios such as geometric constructions, diagrams, graphs, and symbolic annotations. MathVision requires precise visual grounding followed by multi-step mathematical reasoning, making it particularly sensitive to errors caused by incorrect perception rather than insufficient reasoning.
- **MathVerse** (Zhang et al., 2024): MathVerse is a visual mathematics benchmark that evaluates mathematical reasoning grounded in diagrams. It systematically varies the amount of textual and visual information, forcing models to rely on accurate visual perception rather than textual shortcuts. The benchmark is particularly effective at exposing failures where incorrect visual grounding undermines multi-step reasoning.
- **MathVista** (Lu et al., 2024): MathVista is a benchmark for evaluating mathematical reasoning in visual contexts, comprising problems drawn from 28 existing multimodal datasets and 3 newly curated ones, covering tasks such as geometry, algebra, logic, and scientific reasoning over diverse visual inputs like diagrams, charts, and plots. It is designed to stress fine-grained visual grounding and multi-step mathematical reasoning, where state-of-the-art vision language models still lag behind human performance.
- **DynaMath** (Zou et al., 2024): DynaMath is a dynamic visual mathematics benchmark designed

to evaluate the robustness of mathematical reasoning in vision-language models. Each problem is defined as a program that generates multiple concrete variants by altering visual and numerical conditions, such as function parameters, geometric configurations, or graph structures. Unlike prior static benchmarks, DynaMath assesses a model’s ability to generalize across variants of the same underlying problem rather than performance on isolated instances.

- **MM-Vet** (Yu et al., 2023): MM-Vet is a comprehensive multimodal benchmark designed to evaluate the integrated capabilities of large vision-language models. It covers a diverse set of tasks including visual recognition, spatial reasoning, OCR, mathematical reasoning, and commonsense understanding. The benchmark emphasizes multi-faceted reasoning grounded in visual inputs and is widely used for holistic multimodal evaluation.

B Baselines

- **GPT-4o** (Hurst et al., 2024): GPT-4o is a large proprietary multimodal foundation model developed by OpenAI, capable of jointly processing visual and textual inputs for a wide range of perception and reasoning tasks. It serves as a strong closed-source baseline for vision-language reasoning, demonstrating robust performance across visual understanding, mathematical reasoning, and multimodal question answering benchmarks.
- **Claude-3.5 Sonnet** (Anthropic, 2024): Claude-3.5 Sonnet is a state-of-the-art proprietary vision-language model developed by Anthropic, designed for high-quality multimodal understanding and reasoning. The model exhibits strong performance on tasks requiring integrated visual perception and multi-step reasoning, and is commonly used as a competitive closed-source baseline in recent multimodal evaluations.
- **Qwen-VL-Max** (Bai et al., 2023): Qwen-VL-Max is a large-scale proprietary vision-language model from the Qwen family, designed for high-capacity multimodal perception and reasoning. It combines strong visual encoders with advanced cross-modal alignment to support tasks such as visual understanding, OCR, and multimodal question answering. Qwen-VL-Max serves as a

competitive closed-source baseline for evaluating multimodal reasoning performance.

- **Qwen2-VL** (Wang et al., 2024b): Qwen2-VL is an open-weight vision–language model that introduces substantially improved visual encoding and multimodal fusion compared to earlier Qwen-VL models. It enhances visual grounding and cross-modal interaction, enabling stronger performance on a wide range of vision–language understanding and reasoning tasks. Due to its robustness and openness, Qwen2-VL is commonly used as a foundational model in multimodal research.
- **Qwen2.5-VL** (Bai et al., 2025): Qwen2.5-VL further advances the Qwen2-VL series by improving visual perception accuracy, instruction following, and reasoning stability. It is trained with enhanced data curation and optimization strategies, resulting in stronger and more consistent performance on challenging multimodal reasoning benchmarks. As a strong open-weight LLM, Qwen2.5-VL is widely adopted as a base model in recent multimodal reasoning studies.
- **Mulberry** (Yao et al., 2024a): Mulberry is a multimodal large language model designed to acquire o1-like reasoning and reflection capabilities. It introduces Collective Monte Carlo Tree Search (CoMCTS), which performs structured search over intermediate reasoning steps by collaboratively expanding and evaluating reasoning paths across multiple models. The resulting reasoning trees are used to construct a large-scale multimodal dataset with explicit intermediate states, enabling supervised training of models with strong multi-step reasoning and self-reflection abilities.
- **Virgo** (Du et al., 2025): Virgo is a multimodal slow-thinking model that aims to reproduce o1-like reasoning behavior in vision–language settings through explicit long-form reasoning. It is built by fine-tuning a strong base MLLM (Qwen2-VL) with long chain-of-thought data, primarily text-based reasoning trajectories distilled from existing slow-thinking language models. Rather than introducing new architectural modules or controllers, Virgo relies on instruction tuning to elicit extended step-by-step reasoning during inference. This approach enables Virgo to perform complex multimodal reasoning by generating lengthy natural-language thoughts, but applies slow thinking uniformly across inputs without explicit mechanisms to distinguish perceptual uncertainty from reasoning errors.
- **Curr-ReFT** (Deng et al., 2025a): Curr-ReFT is a post-training paradigm for improving reasoning and generalization in small- and medium-scale vision-language models via reinforcement learning. It combines Curriculum Reinforcement Learning with difficulty-aware reward schedules, which progressively transition from simple visual judgment to complex open-ended reasoning, and a rejected-sample-based self-improvement stage that preserves core visual and language capabilities. By structuring reinforcement learning along increasing task complexity, Curr-ReFT enables stable optimization and enhances multimodal reasoning without modifying the underlying model architecture.
- **LMM-R1** (Peng et al., 2025): LMM-R1 is a reasoning-enhancement framework for large multimodal models based on rule-based reinforcement learning. It adopts a two-stage training strategy, first strengthening foundational reasoning using text-only verifiable data, and then generalizing this reasoning capability to multimodal tasks. By decoupling reasoning skill acquisition from multimodal perception, LMM-R1 improves reasoning accuracy in vision-language settings without relying on large-scale high-quality multimodal reasoning annotations.
- **MM-R1** (Liang et al., 2025): MM-R1 is a reasoning-enhanced unified multimodal large language model designed for personalized image generation. It introduces a cross-modal Chain-of-Thought (X-CoT) mechanism that explicitly decomposes generation into an understanding stage and a generation stage, enabling the model to reason about subject identity and attributes before synthesis. MM-R1 further applies reinforcement learning with grouped reward optimization to improve subject fidelity and text-image alignment. Unlike subject-specific fine-tuning methods, MM-R1 supports zero-shot personalization by leveraging intrinsic multimodal reasoning.
- **Vision-R1** (Huang et al., 2025): Vision-R1 is a reinforcement-learning-based multimodal reasoning model that aims to induce chain-of-thought style reasoning in LLMs. It first per-

forms cold-start initialization using a large automatically constructed multimodal CoT dataset, obtained via modality bridging between an MLLM and a text-only reasoning model. The model is then further optimized with Group Relative Policy Optimization (GRPO), combined with Progressive Thinking Suppression Training to mitigate excessive and incorrect long reasoning chains. Vision-R1 focuses on improving complex multimodal reasoning, particularly in visual mathematical tasks, by encouraging structured and reflective reasoning behavior.

- **R1-OneVision** (Yang et al., 2025): R1-OneVision is a multimodal reasoning model that explicitly bridges visual perception and language reasoning via a cross-modal formalization pipeline. R1-OneVision converts images into structured textual representations (e.g., formal descriptions of objects, layouts, and relations), enabling precise language-based reasoning over visual content. The model is trained with supervised fine-tuning followed by reinforcement learning to stabilize step-by-step multimodal reasoning and improve generalization across diverse reasoning tasks.
- **MM-R1** (Liang et al., 2025): MM-R1 is a reasoning-enhanced unified MLLM designed for personalized image generation. MM-R1 introduces a cross-modal Chain-of-Thought strategy that explicitly decomposes generation into visual understanding and conditioned synthesis, and further applies GRPO-based reinforcement learning to align subject fidelity and text consistency, enabling zero-shot personalization without subject-specific fine-tuning.
- **OpenVLThinker** (Deng et al., 2025b): OpenVLThinker is an open-source vision–language model designed to induce *R1-style complex reasoning* in multimodal settings. OpenVLThinker is built on top of Qwen2.5-VL and equips the model with explicit planning, reflection, and verification behaviors by iteratively alternating supervised fine-tuning on chain-of-thought traces and reinforcement learning. Compared to standard LVLMs, it exhibits substantially stronger visual mathematical and logical reasoning while maintaining stable visual grounding.
- **FAST** (Xiao et al., 2025): FAST proposes a fast–slow thinking framework for LVLMs based

on a modified GRPO algorithm. It estimates multimodal question difficulty using both model-dependent signals (pass@k) and image complexity, and applies difficulty-aware length rewards and KL regularization to adaptively control reasoning depth. As a result, the model performs short, efficient reasoning on easy problems while enabling deeper reasoning on hard ones, achieving higher accuracy with significantly reduced response length compared to prior slow-thinking methods.

C Examples of Failure Attribution

We provide examples of the two failure types used in our error attribution pipeline. Visual perception failures occur when the model misreads, misidentifies, or incorrectly grounds key visual evidence, as shown in Figures 5 and 6. In contrast, reasoning errors occur when the model correctly captures the relevant visual information but makes mistakes in the subsequent inference or calculation process, as illustrated in Figures 7 and 8. This distinction is important because perception failures call for re-examining the image, whereas reasoning errors require revisiting the intermediate reasoning steps.

D Extended Case Study Figures

We include enlarged versions of the case study figures from the main paper in Figures 9 and 10 for improved readability and closer inspection of the visual details and model outputs.

Visual Perception Failure Example 1



Question

What is the total number of bears in the image?

Answer by Qwen

To determine the total number of bears in the image, let's count them step by step:

1. Identify the bears in the image. Bears are typically represented with a rounded shape and sometimes with a face.

2. Count the bears from left to right:

- There are two bears at the beginning.
- There are two more bears in the middle.
- There is one bear at the end. Adding these up:
- First set of bears: 2
- Middle set of bears: 2
- Last bear: 1

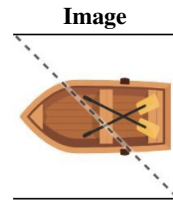
Total number of bears = $2 + 2 + 1 = 5$ So, the total number of bears in the image is 5.

Error Attribution by GPT

"reasoning": "The model's error is fundamentally rooted in a misinterpretation of the visual data. The image contains a total of 3 bears: two near the beginning (positions 3 and 4) and one at the very end (position 10). The model correctly identifies the first set and the last bear, but it explicitly states there are 'two more bears in the middle.' In reality, the objects in the middle are a red train and a green plane. Because the model has hallucinated the existence of bears where there are none (or misidentified other objects as bears), this is a failure of visual grounding rather than a failure of arithmetic logic. The calculation ($2+2+1=5$) is mathematically consistent with the model's flawed visual premises.", "category": "visual_perception"

Figure 5: Example of a **visual perception failure** made by Qwen and the corresponding GPT-based error attribution.

Visual Perception Failure Example 2



Question

Are the oars arranged symmetrically in relation to the dotted line?

Answer by Qwen

To determine if the oars are arranged symmetrically in relation to the dotted line, follow these steps:

1. Identify the Symmetry Line: The dotted line in the image is a diagonal line passing through the center of the boat, dividing it into two triangular sections.
2. Examine the Oars: The oars are positioned with their handles at the stern (the back) of the boat and their blades pointing forward. Each oar has a distinct shape and position.
3. Compare the Oars to the Symmetry Line:
 - The first oar on the left side of the image is positioned such that its handle is closer to the left side of the boat and its tip is closer to the right side.
 - The second oar on the right side of the image is positioned such that its handle is closer to the right side of the boat and its tip is closer to the left side.
4. Evaluate Symmetry:
 - For the oars to be symmetrically arranged relative to the dotted line, the oar on the left side should mirror the oar on the right side across the line. The first oar's position is not mirrored by the second oar's position. The first oar's handle is on the left side and the second oar's handle is on the right side, and the first oar's tip is on the right side and the second oar's tip is on the left side.

Given the above analysis, the oars are not arranged symmetrically in relation to the dotted line.

Error Attribution by GPT

"reasoning": "The model's error lies in its visual assessment of the spatial relationship between the oars and the dotted line. The model explicitly claims that 'The first oar's position is not mirrored by the second oar's position.' However, visual inspection confirms that the oars are arranged in a crossed formation that is symmetrical with respect to the diagonal dotted line (forming an axis of symmetry). The model fails to correctly perceive this geometric alignment, misinterpreting the orientation and reflection of the objects within the visual space. Since the failure is in recognizing the visual property of symmetry present in the image, rather than a misunderstanding of the concept of symmetry itself, it is a perception failure.",
"category": " visual_perception "

Figure 6: Example of a **visual perception failure** made by Qwen and the corresponding GPT-based error attribution.

Reasoning Failure Example 1

Image



Question

If the spinner is spun 100 times, how many times can we expect it to land on pink?

Answer by Qwen

To determine how many times we can expect the spinner to land on pink after 100 spins, we need to follow these steps:

1. Identify the probability of landing on pink:

- The spinner is divided into two sections: one pink and one blue.
- The pink section appears to be a smaller portion of the circle compared to the blue section.
- By visually estimating, the pink section appears to be about 1/4 of the circle.

2. Convert the visual estimate to a probability:

- If the pink section is 1/4 of the circle, the probability (P) of landing on pink is ($\frac{1}{4}$).

3. Calculate the expected number of times the spinner will land on pink over 100 spins:

- The expected number of times (E) is calculated by multiplying the total number of spins by the probability of landing on pink.

- ($E = 100 \times \frac{1}{4}$)

4. Perform the multiplication:

- ($E = 100 \times 0.25 = 25$)

Therefore, we can expect the spinner to land on pink approximately 25 times out of 100 spins.

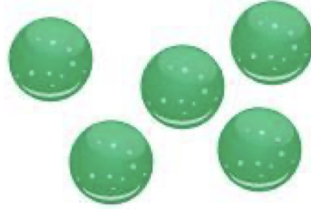
Error Attribution by GPT

"reasoning": "The model correctly identifies the key visual components: a spinner divided into pink and blue sections. It also correctly retrieves the visual fact that the pink section is smaller than the blue one. However, the error occurs in the logical step of **geometric estimation**. The model arbitrarily assigns a value of '1/4' (90 degrees) to the pink section without applying proper spatial reasoning to verify if the angle is actually a right angle (it is clearly acute, likely around 45-60 degrees). The failure lies not in seeing the color or the sector itself, but in the analytical reasoning process of quantifying that visual area into a probability fraction before performing the final calculation.", "category": "reasoning"

Figure 7: Example of a **reasoning failure** made by Qwen and the corresponding GPT-based error attribution.

Reasoning Failure Example 2

Image



Question

If three marbles are to be selected without replacement, what is the probability that all three are green?

Answer by Qwen

To solve the problem of finding the probability that all three selected marbles are green, we need to follow these steps:

Step 1: Determine the total number of marbles From the image, we can see there are 5 marbles in total.

Step 2: Determine the number of green marbles From the image, we can see that all 5 marbles are green.

Step 3: Calculate the total number of ways to select 3 marbles out of 5 This is a combination problem where the order of selection does not matter. The number of ways to choose 3 marbles out of 5 is given by the combination formula $\binom{n}{r} = \frac{n!}{r!(n-r)!}$, where (n) is the total number of items, (r) is the number of items to choose, and $(!)$ denotes factorial.

Here, $(n = 5)$ and $(r = 3)$:

$[\binom{5}{3} = \frac{5!}{3!(5-3)!} = \frac{5!}{3! \cdot 2!} = \frac{5 \times 4 \times 3!}{3! \times 2 \times 1} = \frac{5 \times 4}{2 \times 1} = 10]$ So, there are 10 different ways to choose 3 marbles out of 5.

Step 4: Calculate the number of ways to select 3 green marbles out of 5 Since all 5 marbles are green, there is only one way to choose all 3 marbles to be green. This is a straightforward selection since there are no other colors:

$[\binom{5}{3} = 1]$

Step 5: Calculate the probability The probability (P) of selecting all 3 green marbles is the ratio of the number of favorable outcomes to the total number of possible outcomes:

$[P(\text{all 3 green}) = \frac{\text{Number of ways to choose 3 green marbles}}{\text{Total number of ways to choose 3 marbles}} = \frac{1}{10}]$

Final Answer: The probability that all three selected marbles are green is:

$[\frac{1}{10}]$

Error Attribution by GPT

"reasoning": "The model's visual perception is accurate. It correctly identifies from the image that 'there are 5 marbles in total' and 'all 5 marbles are green.' These visual facts are correct. The error occurs purely in the logical reasoning phase, specifically in Step 4. The model incorrectly reasons that if all marbles are green, there is 'only one way' to choose 3 green marbles. In reality, if all available marbles are green, *every* possible selection of 3 marbles results in 3 green marbles. Therefore, the number of favorable outcomes (choosing 3 green from 5 green) is identical to the total number of outcomes (choosing 3 from 5), which is 10. The probability should be $10/10 = 1$. The model's logic that there is only 'one way' to get the result despite correctly identifying the input set size is a flaw in probabilistic reasoning, not visual perception.", "category": "reasoning"

Figure 8: Example of a **reasoning failure** made by Qwen and the corresponding GPT-based error attribution.

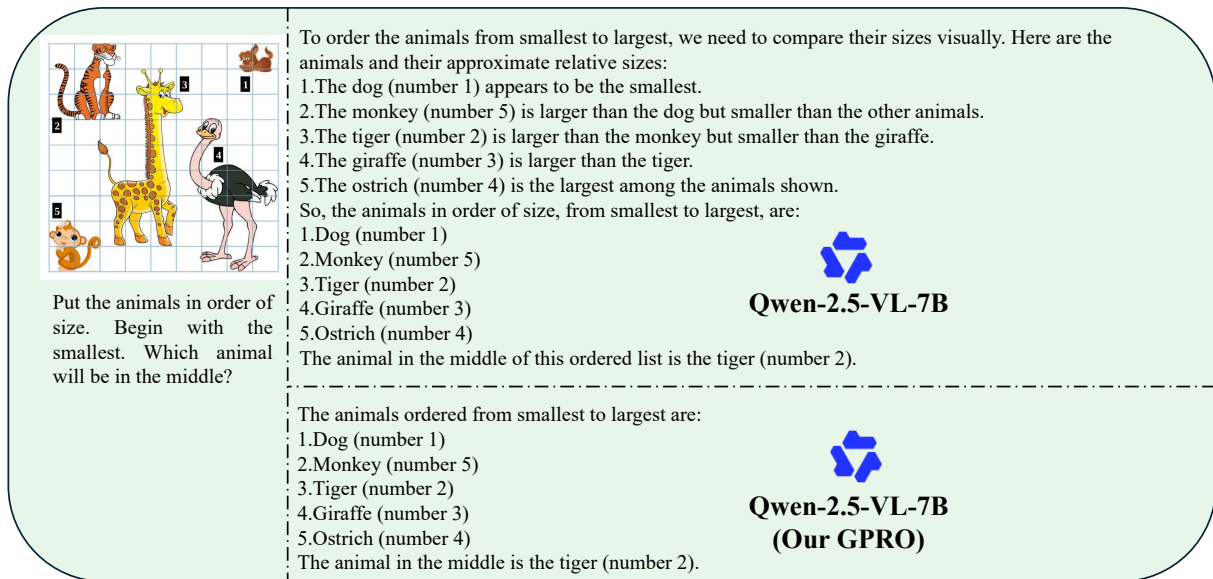


Figure 9: Case Study: Animal size ordering. The baseline produces verbose step-by-step comparisons, while GPRO generates a concise direct answer.

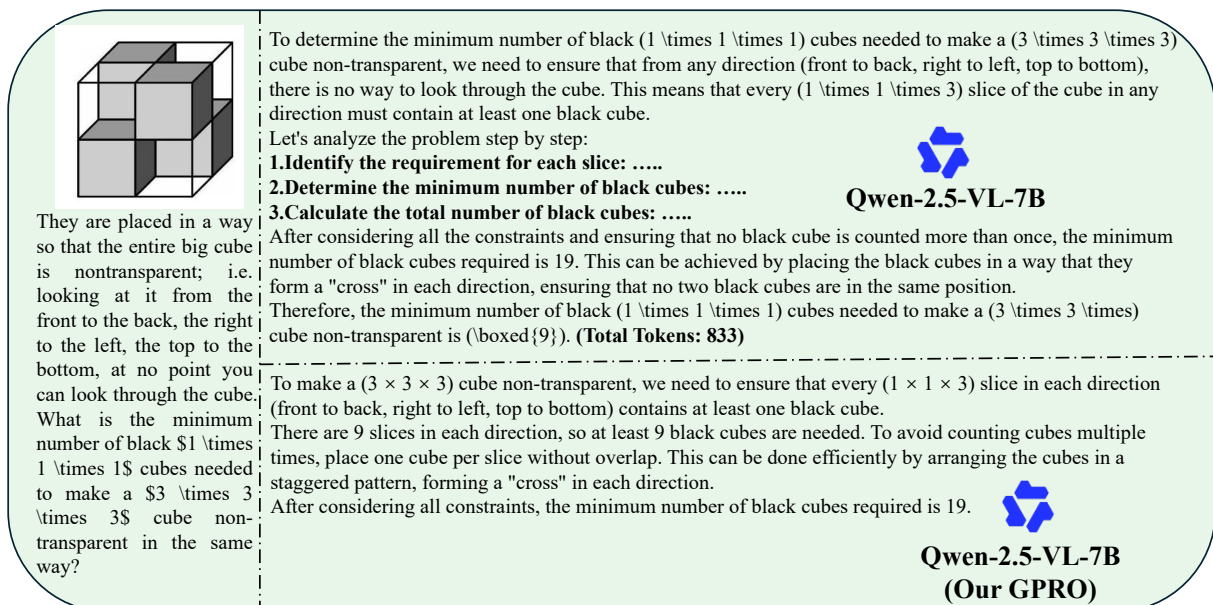


Figure 10: Case Study: Spatial reasoning with cubes. The baseline generates 833 tokens of detailed analysis, while GPRO produces an efficient solution.