Bridging Modality Disconnect in Self-Reflection via Closed-Loop Visually Grounded Verification

Haoyu Zhang[1,2], Zhi Gao[1,2,†], Pengxiang Li[1], Xintong Zhang[1], Yuwei Wu[1,2*], Rui Gao[1,2], Mingyang Gao[1], Xuanyan Chen[1], Anquan Cen[1], Che Sun[2*], Yunde Jia[2,1]

[1]Beijing Key Laboratory of Intelligent Information Technology, School of Computer Science & Technology, Beijing Institute of Technology
[2]Guangdong Laboratory of Machine Perception and Intelligent Computing, Shenzhen MSU-BIT University

Abstract

Self-reflection has become a key mechanism for improving reasoning in Vision-Language Models (VLMs), yet this corrective mechanism often fails when resolving complex fine-grained visual ambiguities. This performance degradation stems from the issue of modality disconnect in self-reflection: most existing models execute self-reflection either within textual or latent space, lacking an explicit mechanism to align textual reasoning with visual evidence. In this paper, we propose MIRROR, a closed-loop visual reflection framework comprising four steps: response generation, error identification, region-based visual verification, and revision. In this cycle, the model first generates an initial response, identifies uncertain logical statements in the response that require visual verification, then grounds them in relevant image regions, and finally revises based on the visual evidence. We construct a multi-turn visual reflection dataset ReflectV, which empowers the model with such a reflective capability. Extensive experiments across diverse multimodal benchmarks show that MIRROR improves performance and reduces visual hallucinations by an average of about 7.2%, demonstrating the advantage of transforming self-reflection from open-loop textual revision into closed-loop, visually grounded verification.

1 Introduction

Reasoning in Vision-Language Models (VLMs) has advanced rapidly, evolving from standard instruction following to the incorporation of Chain-of-Thought (CoT) mechanisms (Wei et al., 2022; Xu et al., 2025) that decompose complex problems into intermediate steps. Concurrently, a series of works introduces self-reflection mechanisms that enable models to critique and revise their outputs through additional reasoning steps (Wang et al., 2025d; Wan et al., 2026; Li et al., 2024c). These methods generally follow a "critique-then-refine" protocol, either expanding textual reasoning chains or adding special reflection tokens. Nevertheless, these methods operate entirely in the textual space—the model revises its answer without re-examining the underlying visual evidence.

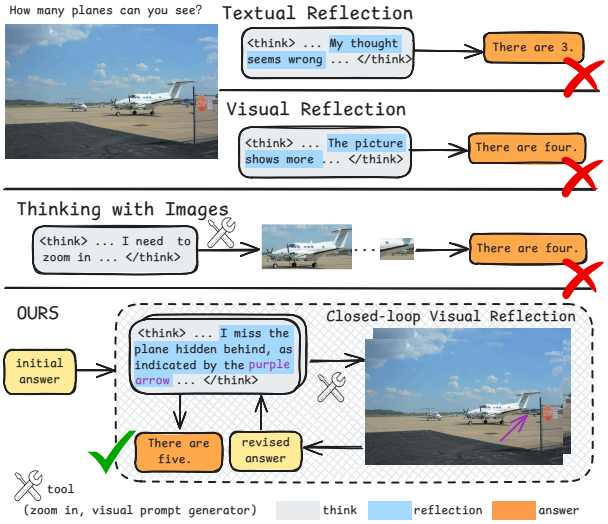


Figure 1: Comparison of existing multimodal reasoning paradigms. MIRROR explicitly seeks visual evidence to achieve closed-loop visually grounded verification.

Recently, some approaches have sought to incorporate visual information into self-reflection (Yang et al., 2026; Jian et al., 2025). Despite improvements, these methods rely on implicit visual refocusing through attention mechanisms rather than explicit visual verification. Crucially, while these works identify that visual attention progressively diminishes as token sequences grow, their implicit feature-guided designs inherently make it difficult to efficiently integrate visual information in the self-reflection process. This bottleneck is further reinforced by the recent analysis (Wu et al.,

* Corresponding author, † Project lead

Proceedings of the 2026 Conference on Empirical Methods in Natural Language Processing, pages 45747–45788
October 24-29, 2026 ©2026 Association for Computational Linguistics

2025), which demonstrates that inference-time self-verification behaviors often fail to yield reliable reasoning improvements. As a result, models often produce response revisions that remain unsupported by the underlying visual evidence—a limitation we formalize as modality disconnect in self-reflection: the model refines its answer without effectively referring to visual evidence. Therefore, self-reflection should explicitly ground reflective reasoning in verified visual evidence.

The paradigm of "Thinking with Images" (Su et al., 2025b) provides a promising foundation for closing the gap between reflective reasoning and visual perception. By enabling region-based inspection, this approach can incorporate visual information during multi-turn reasoning. However, existing methods following this approach (Su et al., 2025a; Zhang et al., 2025b) typically operate in an open-loop manner—invoking visual operations based on the input question, rather than in response to incorrect reasoning steps. As a result, this can lead to redundant or missed visual verification.

In this paper, we propose a framework for Multimodal Iterative Reasoning via Reflection On visual Regions (MIRROR). It addresses modality disconnect in self-reflection by making visual evidence an explicit part of the reflection loop. As shown in Figure 1, MIRROR iteratively performs four stages: response generation, error identification, visual verification, and answer revision. At each turn, the model generates a visual prompt to inspect relevant visual evidence for response uncertainties and revises its response accordingly, coupling reflective reasoning with explicit visual verification. We further construct ReflectV, a visual reflection dataset of approximately 24k samples, ranging from general VQA to complex reasoning tasks, to endow the model with this capability. The dataset combines internal reflection with precise visual cues to teach the model how to perform visual reflection. We fine-tune Qwen2.5-VL (3B/7B), Qwen3-VL-8B, and InternVL3.5-8B (Bai et al., 2025b,a; Wang et al., 2025b) on ReflectV, and evaluate their performance on diverse benchmarks. Experimental results show that MIRROR improves performance and reduces hallucinations, demonstrating the importance of transforming self-reflection from open-loop textual revision into closed-loop visually grounded verification.

In summary, our contributions are as follows:

- We propose MIRROR, a closed-loop visual reflection framework that bridges modality disconnect in self-reflection by transforming open-loop textual revision into visually grounded verification.
- We construct ReflectV, a high-quality dataset of about 24k samples through a multi-agent pipeline. The dataset explicitly models trajectories of error detection, visual verification, and answer correction, teaching the model when to doubt itself and how to seek visually grounded evidence.
- We fine-tune multiple VLM architectures on our dataset to obtain the MIRROR models, and extensive evaluations demonstrate the efficacy and generalizability of our framework.

2 Related Work

2.1 Multimodal Reasoning in VLMs

Multimodal reasoning in VLMs has shifted from visual question answering toward advanced Chain-of-Thought (CoT) mechanisms (Wei et al., 2022; Xu et al., 2025) that perform reasoning through step-by-step intermediate inferences. Research in visual grounding has employed visual prompts—such as bounding boxes and points—to enhance referential precision. Models like Shikra (Chen et al., 2023) and SoM-GPT4V (Yang et al., 2023) demonstrate that explicit visual markers can significantly help in capturing fine-grained details, and STVG-R1 (Zhang et al., 2026a) further extends visual prompting to video grounding with instance IDs as markers. Yet GranFact (Fan et al., 2026) reveals that fine-grained generation is intrinsically more error-prone than coarse-grained generation, suggesting that fine-grained claims should be reliability-prioritized. Beyond visual grounding, recent works have explored a broader *"Thinking with Images"* paradigm (Su et al., 2025b,a; Zhang et al., 2025b), where VLMs actively attend to visual evidence during reasoning to acquire finer-grained visual signals for verification: VisCoT (Shao et al., 2024a) predicts question-relevant bounding boxes and reasons over the localized visual regions, while CogCoM (Qi et al., 2025) performs visual operations such as grounding and zooming through a Chain of Manipulations. AdaptMMBench (Zhang et al., 2026b) evaluates adaptive mode selection between tool-augmented visual reasoning and text reasoning. However, prior works lack a reflection-driven control signal and rarely leverage visual tools to verify response uncertainties against visual

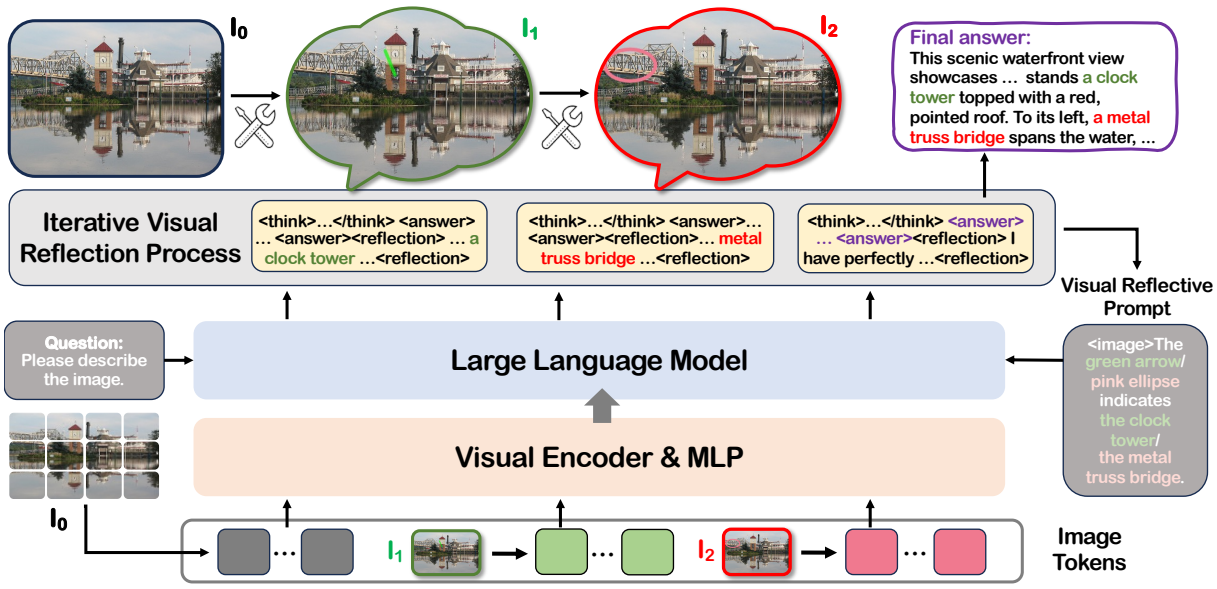


Figure 2: MIRROR performs closed-loop: it iteratively generates an initial response, identifies response uncertainties, invokes the visual prompt generator for verification, and revises based on the rendered visual evidence.

evidence.

2.2 Reflection in Vision-Language Models

Reflection in VLMs aims to improve reasoning reliability by allowing models to critique and revise their own predictions. Recent methods explore this capability through textual self-correction, addressing issues such as logical inconsistency, insufficient utilization of visual information, and hallucination. For instance, Critique Fine-Tuning (Wang et al., 2025d) integrates explicit critique steps into the training objective, while methods like FIRE (Li et al., 2024c) employ "response-feedback-refinement" triplets to enable models to iteratively refine answers based on feedback signals. SRPO (Wan et al., 2026) introduces the <reflection> token and utilizes GRPO (Shao et al., 2024b) to optimize the model's reflective capability. Critic-V (Zhang et al., 2025a) employs a separate VLM critic to identify errors and provide iterative feedback for multimodal response refinement. While these methods strengthen logical reasoning, they often suffer from text-level hallucinations, motivating recent works to shift toward visual reflection. VL-Rethinker (Wang et al., 2026) adopts selective sample replay for effective data filtering and forced rethinking strategies, alleviating the diminishing reflection advantage. Look-Back (Yang et al., 2026) designs a special token <back> to guide the model to re-attend to image information. Reflection-V (Jian et al., 2025) leverages a supervised fine-tuning phase to align reasoning with granular visual details, followed by reinforcement learning with attention-aware rewards to reinforce the model's focus on critical regions. Additionally, some other studies attempt to improve reasoning performance through multi-agent collaboration (e.g., Mulberry (Yao et al., 2026)) or tool-chain expansion (e.g., Thyme (Zhang et al., 2025c)); MAT (Gao et al., 2025) tunes a VLM-driven agent with automatically generated tool-usage trajectories, and ToolEQA (Zhai et al., 2025) performs tool-augmented multi-step reasoning for embodied question answering.

Unlike most existing approaches, which often fail to re-verify visual evidence explicitly, the proposed MIRROR enables visually grounded reflection. By integrating visual tools, our framework transforms reflection from implicit visual refocusing into an active verification process, where the model explicitly reflects and checks specific image regions to support answer revision with visual evidence.

3 Method

As shown in Figure 2, MIRROR transforms one-pass VLM inference into a closed-loop verification process: (i) generate an initial response, (ii) identify uncertainties within the response, (iii) verify visual evidence by invoking a visual prompt generator to mark task-relevant regions, and (iv) refine the final answer based on the updated image.

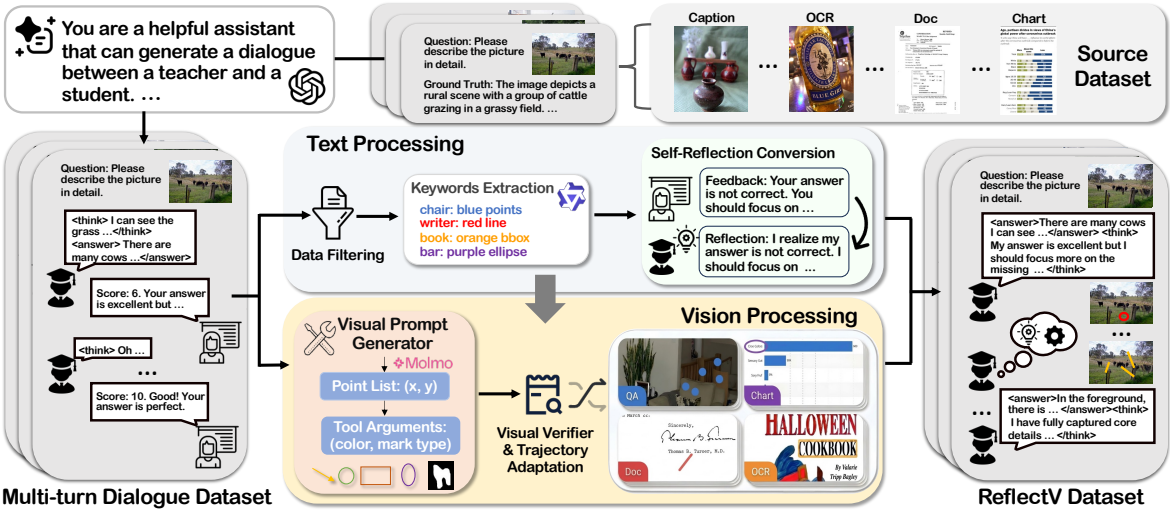


Figure 3: Overview of the ReflectV construction pipeline. Multi-turn reflective trajectories are constructed from diverse multimodal datasets and further processed into visually grounded verification trajectories.

3.1 Formulation

We formulate visual reflection as a multi-turn generation process. Given an image I_0 and a user query q, the objective of the VLM, denoted as π_θ, is to generate a structured reflection trajectory $\mathcal{Y} = \{y_1, y_2, \ldots, y_K\}$ over K interaction turns. At each turn k, the model generates an output tuple $y_k = (a_k, r_k, v_k)$, where a_k is the textual response, r_k specifies the reflection content, and v_k denotes the verification specification. Unlike CoT reasoning, which operates on a fixed visual input, our framework actively verifies visual evidence during reflection. The visual verification specification v_k triggers a visual prompt generator $\mathcal{G}$, yielding an updated image $I_k = \mathcal{G}(I_0, v_k)$ that highlights specific regions for the subsequent turn. For the $(k+1)$-th turn, the model takes the updated image I_k and the interaction history $h_{<k+1} = \{a_j, r_j, v_j\}_{j=1}^{k}$ as input to generate the next response tuple y_{k+1}. This process can be formulated as

$$y_{k+1} \sim \pi_\theta(y_{k+1} \mid I_k, q, h_{<k+1}), \tag{1}$$

where $y_{k+1} = (a_{k+1}, r_{k+1}, v_{k+1})$ represents the newly generated output tuple. This loop continues until the model determines that no further visual verification is needed.

3.2 Closed-Loop Visual Verification

A key innovation of MIRROR is the integration of a visual prompt generator $\mathcal{G}$ for visual reflection. This module allows the model to actively "verify" its response by grounding textual uncertainties into visual cues. The workflow proceeds as follows.

Verification specification generation. When the model identifies content that requires further visual verification, it initiates a specification enclosed within a special token <tool_call> structure v_k =

```
<tool_call>{
  name : "Visual Prompt Generator",
  flag : true/false
  anchor : "keywords..."
  args : {color : "red", mark_type : "point"}
}</tool_call>,
```
(2)

where `name` invokes the visual module, `anchor` specifies the textual region derived from reflection content for localization, and `args` defines the rendering attributes for visual prompts. The `flag` field controls whether visual tools are invoked at each turn: when set to `true`, the model requests grounded visual prompts to verify its response, and when `false`, the current answer is accepted as the final output without further visual verification.

Visual grounding. Upon receiving v_k, the generator $\mathcal{G}$ localizes the anchor, predicts its image coordinates, and overlays the requested marker (e.g., "red point") according to the arguments, producing an updated image $I_k = \mathcal{G}(I_0, v_k)$.

Reasoning refinement. The resulting image I_k, now explicitly highlighting the neglected or misinterpreted regions, is fed back into the VLM. This process forms an iterative feedback loop, forcing the model to re-attend to the updated image (I_k) alongside the query history before generating the next reasoning step.

3.3 Training for Reflective Reasoning

We utilize a visual reflection dataset $\mathbb{D}$ (detailed in Section 4) to train the VLM via supervised fine-tuning. The training objective is to optimize the model's ability to iteratively execute a closed-loop visual reflection process. We formulate the training objective as a multi-turn auto-regressive loss. Formally, for a training sample $(I_0, q, \mathcal{Y})$, we maximize the likelihood of the sequence over K turns:

$$\min_{\theta} \mathbb{E}_{(I_0,q,\mathcal{Y})\sim\mathbb{D}}\Big[-\sum_{k=1}^{K}\log \pi_\theta\big(a_k, r_k, v_k \mid I_{k-1}, q, h_{<k}\big)\Big], \tag{3}$$

where $h_{<k} = \{a_j, r_j, v_j\}_{j=1}^{k-1}$ represents the input history, and I_{k-1} denotes the updated image processed by the visual prompt generator in the previous turn. This objective jointly supervises answer revision (a_k), reflective reasoning (r_k), and visual verification specification generation (v_k).

4 Visual Reflection Dataset Construction

We construct ReflectV as illustrated in Figure 3. The construction process consists of three stages: (i) multi-turn dataset construction, which generates and filters reflective reasoning chains, (ii) visual evidence construction, which converts textual reflection into visual prompts and visually grounded verification trajectories, and (iii) trajectory refinement and adaptation, which produces mixed multi-turn and single-turn supervision for both iterative correction and efficient one-pass inference. Additional details on data sources, pipeline implementation, and data filtering strategy ablations are provided in Appendix A.

4.1 Multi-Turn Dataset Construction

We adopt the data curation pipeline proposed in FIRE (Li et al., 2024c) to construct reflective reasoning trajectories for visually grounded reflection. Instead of relying solely on existing conversational datasets, we transform QA pairs into multi-turn dialogues. Specifically, the teacher model GPT-4o (Hurst et al., 2024) simulates a "student-teacher" interaction based on ground-truth annotations, where the "student" produces intermediate responses containing potential uncertainties, and the "teacher" provides corrective feedback and a score s_t until the correct answer is given. This process converts QA samples into rich trajectories of error identification and answer revision.

To ensure the quality of the generated trajectories, we further apply a filtering strategy based on the teacher score s_t. A trajectory is preserved only if its scores strictly increase across turns ($s_{t+1} > s_t$), the dialogue terminates with a maximum final score ($s_{\text{final}} = 10$), and the final response is semantically aligned with the ground-truth answer, preventing false positives caused by verifier misjudgment.

4.2 Visual Evidence Construction

To explicitly connect reflective reasoning with visual evidence, we convert textual reflection into visually grounded verification trajectories.

First, we extract verification keywords from the reflection content and use them as tool arguments for visual prompting (Section 3.2). For general VQA tasks, entity-centric keywords are directly parsed from the reflection. For dense perception tasks such as OCR and chart understanding, where reflections often lack spatial specificity, we instead derive verification keywords from question entities. To further strengthen alignment between reflection and visual evidence, visual attribute descriptions (e.g., "as indicated by the red point") are injected back into the reflection text.

Second, we employ a self-reflection conversion mechanism to convert external corrective feedback into first-person reflective reasoning. Concretely, an LLM rewrites teacher feedback (e.g., "Your answer is incorrect") into self-reflective thoughts (e.g., "Upon closer inspection, I realize my previous answer was incorrect"). This conversion process teaches the model how to identify potentially incorrect response elements and initiate visual verification during reflection.

4.3 Trajectory Refinement and Adaptation

After visual prompting is applied, we further refine the trajectories to ensure reliable visual grounding and balanced supervision.

First, we perform visual consistency filtering using a visual verifier to assess the semantic alignment between generated visual prompts and reflection content. This step removes instances with ambiguous or incorrectly grounded visual cues.

Second, we adopt a mixed trajectory adaptation strategy. Verified reflective trajectories are preserved as multi-turn supervision for iterative visual verification, while failed or redundant trajectories are simplified into direct QA pairs $(I_0, q \rightarrow a_{\text{final}})$. This mixed supervision strategy improves both closed-loop reflective reasoning and efficient one-

pass inference while reducing over-reliance on unnecessarily long correction chains.

5 Experiment

5.1 Experimental Setting

We conduct experiments across 12 diverse benchmarks. For general capabilities, we employ MM-Vet (Yu et al., 2023), MMStar (Chen et al., 2024), and SeedBench-2-Plus (Li et al., 2024b). In the domain of OCR and document understanding, we utilize TextVQA (Singh et al., 2019), OCRBench (Liu et al., 2024b), and ChartQA (Masry et al., 2022). To assess the models' robustness against hallucinations, we include POPE (Li et al., 2023) and HallusionBench (Guan et al., 2024). Furthermore, we evaluate fine-grained perception using HRBench (Wang et al., 2025c), MME-RealWorld-Lite (Zhang et al., 2025d), and VStarBench (Wu and Xie, 2024), alongside MathVision (Wang et al., 2024) for mathematical reasoning. We employ VLMEvalKit (Duan et al., 2024) to evaluate models on these benchmarks, and use GPT-5-mini (Singh et al., 2025) as the judge to evaluate the correctness of the model's responses for datasets if needed.

5.2 Comparison with Base Models

We compare MIRROR with strong VLMs, including Qwen2.5-VL (Bai et al., 2025b), InternVL3 (Zhu et al., 2025), LLaVA-OneVision (Li et al., 2024a). As shown in Tables 1 and 2, MIRROR consistently outperforms the base model and other strong baselines, while MIRROR (w/o loop) also achieves notable performance gains. We also provide more qualitative visualization in Appendix E and results of other model sizes and architectures in Section 5.4.

General Capabilities and Math Reasoning. On benchmarks evaluating broad multimodal skills and mathematical reasoning, MIRROR demonstrates significant gains over the base models. As illustrated in Figure 4, other models often fail in tasks requiring multi-step logic due to attention drift. In contrast, MIRROR actively identifies and resolves logical discrepancies during complex problem-solving. Notably, despite the absence of mathematical data, MIRROR still achieves notable enhancements in foundational mathematical logic.

OCR & Document Understanding. On text-intensive tasks, MIRROR achieves the best performance among the compared models. MIRROR helps the model explicitly revisit neglected visual details, overcoming the limitations inherent in one-pass inference. We provide a representative correction case in Figure 17.

Hallucination and Fine-grained Perception. Crucially, MIRROR excels in reducing hallucinations (e.g., +13.36 points on HallusionBench) and enhancing fine-grained perception. By anchoring intermediate reasoning in verifiable visual evidence, our method imposes a strong constraint on the trajectory.

5.3 Comparison with Reasoning Models

Implementation Setup. To evaluate the effectiveness of our closed-loop visual reflection mechanism, we compare MIRROR with representative methods from three reasoning paradigms: textual reflection, visual reflection, and Thinking with Images. Specifically, we include VL-Rethinker (Wang et al., 2026) for textual reflection and Look-Back (Yang et al., 2026) for visual reflection, reporting both its solution-level and semantic-level variants. For Thinking with Images, we compare against PixelReasoner (Wang et al., 2025a), DeepEyes (Zheng et al., 2025), and Adaptive-CoF (Zhang et al., 2025b). We select one representative benchmark from each of four major categories and include the SFT variants of several models to ensure a fair comparison.

As shown in Table 3, MIRROR achieves superior performance by addressing the inherent limitations of existing paradigms. First, pure text-based reflection suffers from hallucinations. For instance, VL-Rethinker scores low on POPE (84.19), confirming that self-correction without visual information integration is insufficient. Second, while "Thinking with Images" and "Visual Reflection" methods introduce active visual tools or latent refocusing, they largely operate in an open-loop manner, lacking an error-correction mechanism to rectify perceptual failures. In contrast, MIRROR incorporates a closed-loop visually grounded verification process, consequently surpassing the strongest baselines across all metrics. We also provide more efficiency analysis and visualization in Appendix D.1.

Attention Visualization. To understand why MIRROR achieves stronger reasoning, we visualize the average per-token attention maps in Figure 5. To isolate the model's intrinsic reflection capability, all visualizations are captured from the first reasoning round before any tool invocation. As shown in Figure 5, baseline methods exhibit scattered or misaligned attention patterns, struggling to focus

Table 1: Performance on General and OCR & Document benchmarks. The best and second-best results are highlighted in **bold** and underlined. MIRROR (w/o loop) is an open-loop variant without reflective loop.

Model	General Capabilities			OCR & Document		
	MM-Vet	MMStar	SeedBench-2-Plus	TextVQA-Val	OCRBench	ChartQA-Test
LLaVA-OneVision-7B	48.80	61.70	–	76.10	62.10	80.00
InternVL3-2B	54.95	60.70	64.95	77.00	82.20	76.08
InternVL3-8B	64.27	61.50	69.61	80.51	85.00	79.64
Qwen2.5-VL-3B	47.39	55.87	68.81	79.12	82.60	83.20
Qwen2.5-VL-7B	56.60	61.21	70.88	84.90	83.20	86.08
MIRROR(w/o loop)	59.91	62.80	70.36	85.37	88.30	86.56
MIRROR(ours)	**66.70**	**73.33**	**76.86**	**86.62**	**92.00**	**87.92**

Table 2: Performance comparison on Hallucination, Fine-grained Perception, and Math benchmarks. The best and second-best results are highlighted in **bold** and underlined, respectively. MME-RW indicates MME-RealWorld-Lite.

Model	Hallucination		Fine-grained Perception			Math
	POPE	HallusionBench	HRBench-4K	MME-RW	VStarBench	MathVision
LLaVA-OneVision-7B	78.10	31.60	63.00	–	72.30	18.30
InternVL3-2B	89.60	42.50	61.75	43.88	68.59	21.71
InternVL3-8B	90.37	49.90	70.00	48.83	68.06	20.39
Qwen2.5-VL-3B	86.21	63.09	50.25	42.15	72.77	25.66
Qwen2.5-VL-7B	86.45	68.66	68.87	44.29	75.39	23.36
MIRROR(w/o loop)	87.95	68.24	69.13	46.01	76.44	27.30
MIRROR(ours)	**94.42**	**82.02**	**72.88**	**51.49**	**83.77**	**28.29**

Table 3: Performance comparison with SOTA reasoning methods, all fine-tuned on Qwen2.5-VL-7B. Best and second-best results are **bold** and underlined.

Method	OCRBench	POPE	MME-RW	MM-Vet
Text Reflection				
VL-Rethinker	85.40	84.19	47.21	56.19
Visual Reflection				
LookBack (Solution)	87.50	88.20	49.80	63.50
LookBack (Semantic)	88.60	89.80	50.40	65.10
Thinking with Images				
PixelReasoner-SFT	76.35	80.01	44.73	47.68
PixelReasoner	82.10	86.03	49.70	52.98
DeepEyes	88.10	87.70	49.50	60.28
Adaptive-CoF-SFT	85.62	82.53	50.10	62.73
Adaptive-CoF	86.00	89.30	50.90	66.21
MIRROR	**92.00**	**94.42**	**51.49**	**66.70**

on task-critical regions. In contrast, MIRROR consistently demonstrates concentrated attention on correct areas across both reasoning-intensive and perceptual-grounding tasks. This suggests that the reflective training paradigm improves the model's ability to focus on task-relevant visual regions even before external tool invocation.

5.4 Ablation Experiments

Impact of Reflective Training Paradigm. To disentangle the benefits of our closed-loop reflection mechanism from the intrinsic quality of the domain-specific knowledge in ReflectV, we compare MIRROR against vanilla-SFT variants fine-tuned on the same dataset but reformatted into QA pairs. As shown in Table 4, vanilla-SFT variants outperform the base models due to domain adaptation, yet MIRROR achieves substantially higher gains across all model sizes and architectures, validating the effectiveness of our reflective training paradigm.

Scalability across Model Sizes and Architectures. We further investigate the universality of our approach by applying MIRROR to different model sizes and architectures. As detailed in Table 4, MIRROR achieves consistent gains across all configurations: for 3B/7B/8B variants, it improves over the base model by +13.21/+10.10/+12.6 points on MM-Vet, and by +2.80/+8.80/+2.80 on OCRBench. Beyond the Qwen family, we further fine-tune InternVL3.5-8B (Wang et al., 2025b), an architecturally distinct backbone, with the same recipe on ReflectV. MIRROR-InternVL3.5-8B consistently outperforms both the base model and the vanilla-SFT variant (e.g., +4.43 on OCRBench and

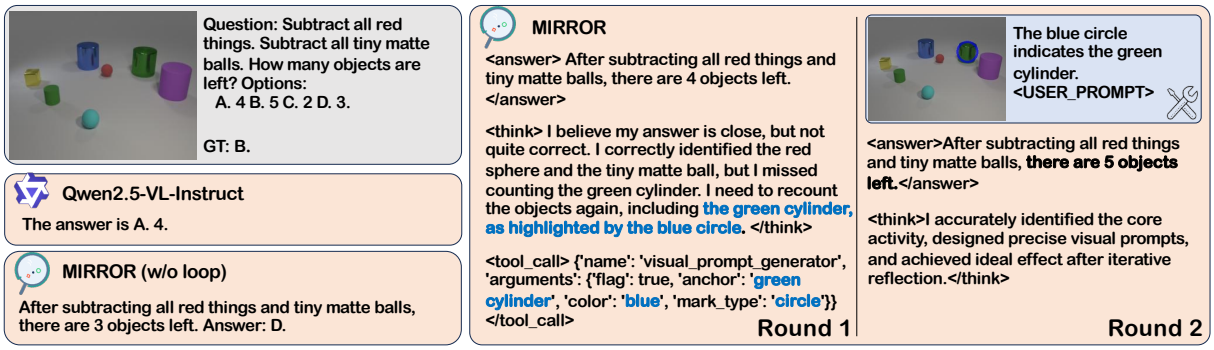


Figure 4: Qualitative examples of iterative visual reflection. Compared to Qwen2.5-VL and MIRROR (w/o loop) (without reflective loop), MIRROR successfully corrects errors via visually grounded verification.

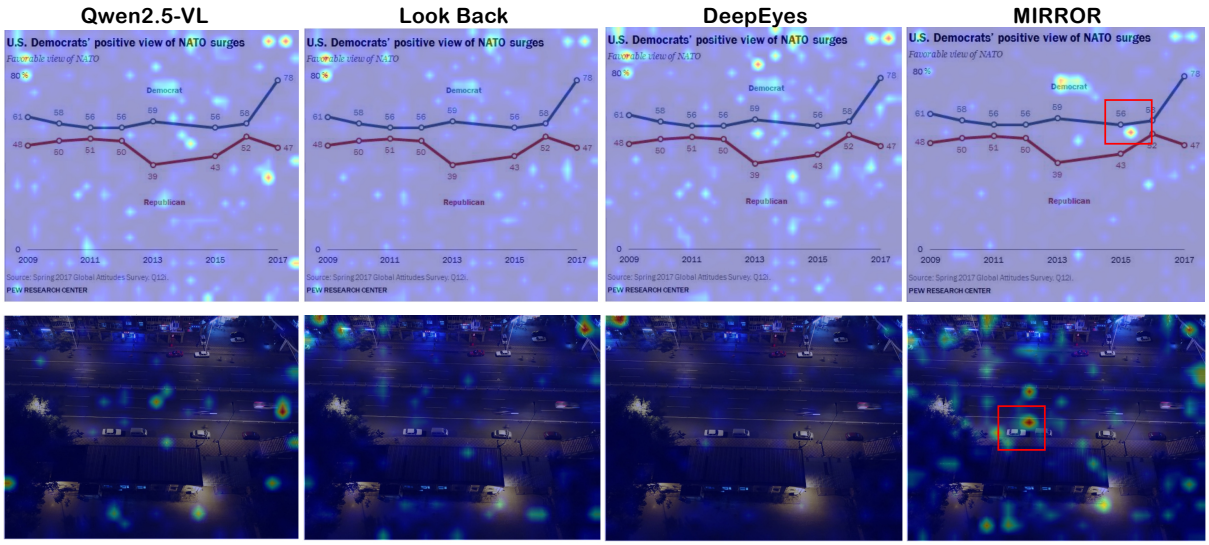


Figure 5: Visualization of per-token attention maps in the first reasoning round (before tool invocation). Correct target regions (the specific value "56" and the "van") are highlighted with red boxes in MIRROR's attention maps.

Table 4: Ablation study on training paradigms, model sizes, and architectures. Best and second-best results are **bold** and underlined.

Method	OCRBench	POPE	VStarBench	MM-Vet
Qwen2.5-VL-3B	82.60	86.20	72.77	47.39
+vanilla-SFT	81.60	86.31	75.92	56.33
MIRROR-3B	**85.40**	**87.74**	**80.63**	**60.60**
Qwen2.5-VL-7B	83.20	86.45	75.39	56.60
+vanilla-SFT	86.90	86.99	78.01	61.01
MIRROR-7B	**92.00**	**94.42**	**83.77**	**66.70**
Qwen3-VL-8B	90.50	88.07	74.35	67.00
+vanilla-SFT	90.80	88.46	75.23	67.52
MIRROR-8B	**93.30**	**95.43**	**86.83**	**79.56**
InternVL3.5-8B	83.24	88.71	70.70	83.13
+vanilla-SFT	84.18	89.20	72.46	84.05
MIRROR-InternVL3.5-8B	**87.67**	**92.59**	**78.20**	**85.61**

+7.50 on VStarBench over the base), indicating the gains are not limited to a single model family. Notably, MIRROR-8B achieves the highest MM-Vet score, demonstrating that our closed-loop visually grounded verification mechanism effectively transfers across both the Qwen and InternVL model families, generalizing well to more VLM backbones with enhanced native reasoning capabilities.

Trajectory Adaptation. To evaluate the effectiveness of our trajectory adaptation strategy, we investigate the impact of ρ, defined as the ratio of multi-turn reflective chains, as shown in Table 5. Training exclusively on multi-turn reflective data ($\rho = 1.0$) leads to a significant degradation compared to the mixed strategy.

Effectiveness of Closed-loop Reflection. As illustrated in Table 6, we compare MIRROR with three variants to isolate the effects of visual tools and iterative closed-loop verification. Qwen2.5-VL-7B-Tools adds the same visual tools to the base

Table 5: Ablation study regarding multi-turn sample ratio ρ for trajectory adaptation on MM-Vet. Best and second-best results are **bold** and underlined.

Ratio ρ	OCR	Math	Spat	Gen	Rec	Avg.
$\rho = 0.5$	61.85	51.54	59.73	45.75	55.61	55.87
$\rho = 0.6$	64.81	56.15	62.13	56.75	61.28	60.32
$\rho = 1.0$	65.19	56.54	61.47	54.12	59.04	58.81
$\rho = 0.75$	**69.90**	**60.77**	**70.80**	**61.62**	**64.40**	**66.70**

Table 6: Ablation study of the closed-loop reflection mechanism on the MM-Vet benchmark (Average Score). **Bold** and underlined indicate the best and second-best performance, respectively.

Configuration	Loop	Tools	MM-Vet
Qwen2.5-VL-7B	-	-	56.60
Qwen2.5-VL-7B-Tools	-	✓	57.90
Qwen2.5-VL-7B-MIRROR	✓	✓	57.10
MIRROR (w/o loop)	-	-	59.91
MIRROR-Tools	-	✓	62.70
MIRROR (ours)	✓	✓	**66.70**

model without reflective training. Qwen2.5-VL-7B-MIRROR applies the MIRROR-style loop in a few-shot setting. MIRROR-Tools disables the iterative loop. We observe two key findings. (1) Tool access alone is insufficient. Simply equipping the base model with visual tools without closed-loop training yields only marginal gains. (2) Verification drives the gains. Removing the iterative loop (MIRROR-Tools) causes a substantial performance drop. This confirms that our closed-loop visual verification, rather than mere tool integration, is the primary driver of MIRROR's success: external models merely translate the VLM's textual reflection into visual prompts, whereas the reflection loop itself is performed entirely by the VLM. Additional efficiency comparisons with other models in Appendix D.2 show that this verification process incurs only modest overhead. Detailed analyses of multi-turn outcomes and failure propagation in tool-triggered trajectories are provided in Appendices B.4 and F.1.

Robustness to Visual Grounding Tools. Although Molmo-7B is adopted as the default grounding tool, MIRROR does not depend on a specific external model. As shown in Table 7, replacing Molmo-7B with a lightweight open-set detector, GroundingDINO-base (Liu et al., 2024a) (∼200M parameters), or a 3B-scale grounding MLLM, Rex-Omni (Jiang et al., 2026), requires no retraining. All variants substantially outperform the tool-equipped open-loop counterpart (57.90): even with the 200M detector, MIRROR achieves 64.22 on MM-Vet, showing that the closed-loop reflection remains robust across tool capacities. Moreover, we train **MIRROR-NativePoint** to remove this dependency entirely, which directly predicts the point coordinates by replacing textual keywords in the training trajectories with verified coordinate annotations. MIRROR-NativePoint achieves 64.68 on MM-Vet (+8.08 over the base model) with zero external parameters, materializing a fully intrinsic reflection mechanism within the VLM. The remaining 2.02-point gap to MIRROR-Molmo reflects a flexible trade-off between absolute performance and zero-dependency deployment.

Table 7: Ablation on the visual grounding tool (MM-Vet). Tool parameters are shown in parentheses. Best results are in **bold**.

Configuration	Loop	External Tool	MM-Vet
Qwen2.5-VL-7B	–	–	56.60
Qwen2.5-VL-7B-Tools	–	Molmo-7B (7B)	57.90
MIRROR-NativePoint	✓	–	64.68
MIRROR-GroundingDINO	✓	DINO-base (∼200M)	64.22
MIRROR-Rex-Omni	✓	Rex-Omni (∼3B)	65.14
MIRROR-Molmo (ours)	✓	Molmo-7B (∼7B)	**66.70**

6 Conclusion

In this work, we have presented MIRROR, a framework that enhances multimodal reasoning by transforming self-reflection from open-loop textual revision into closed-loop, visually grounded verification. We construct ReflectV, a high-quality dataset designed to teach models to explicitly ground their reasoning in specific image regions. Our experiments demonstrate that MIRROR outperforms strong baselines, validating that equipping VLMs with the agency to "look again" explicitly and verify visual evidence is essential for reducing hallucinations and achieving robust reasoning.

Limitations

Despite its effectiveness, MIRROR still has limitations in both capability scope and system design. For capability scope, visual verification remains difficult in abstract domains and fine-grained compositional attribute binding, where the target evidence is hard to represent with simple spatial anchors. We provide a detailed analysis of these cases in Appendix F. For system design, MIRROR currently relies on an external visual prompt generator

to localize anchors and render visual cues, which introduces additional parameters and system complexity beyond the backbone VLM. Future work will focus on more fine-grained verification mechanisms and explore how to internalize region-level reflection and verification through the native visual grounding capabilities of VLMs.

Acknowledgments

This work was supported by the Natural Science Foundation of China (NSFC) under Grant No. 62406009, the Shenzhen Science and Technology Program under Grant No. JCYJ20241202130548062, the Natural Science Foundation of Shenzhen under Grant No. JCYJ20230807142703006, the Fundamental Research Funds for the Central Universities No. 2025CX11016, and the Key Research Platforms and Projects of the Guangdong Provincial Department of Education under Grant No. 2023ZDZX1034.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. 2023. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, and 1 others. 2025. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 91–104.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, and 1 others. 2024. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201.

Xiaomeng Fan, Wei Wu, Yuwei Wu, Zhi Gao, Shiyu Luo, Mingyang Gao, Haoyu Zhao, Zhenxin Diao, Yuxuan Ba, Lijia Feng, and 1 others. 2026. Reliability-prioritized fine-grained generation in multimodal large. *arXiv preprint arXiv:2606.29573*.

Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaojian Ma, Tao Yuan, Yue Fan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. 2025. Multi-modal agent tuning: Building a vlm-driven agent for efficient tool usage. In *International Conference on Learning Representations*, volume 2025, pages 13354–13385.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. volume 1, page 3.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

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*.

Pu Jian, Junhong Wu, Wei Sun, Chen Wang, Shuo Ren, and Jiajun Zhang. 2025. Look again, think slowly: Enhancing visual reflection in vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 9262–9281.

Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. 2026. Detect anything via next point prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25472–25483.

Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. 2018. Dvqa: Understanding data visualizations via question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*.

Pengxiang Li, Zhi Gao, Bofei Zhang, Tao Yuan, Yuwei Wu, Mehrtash Harandi, Yunde Jia, Song-Chun Zhu, and Qing Li. 2024c. Fire: A dataset for feedback integration and refinement evaluation of multimodal models. *Advances in Neural Information Processing Systems*, 37:101618–101640.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. pages 292–305.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

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

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, and 1 others. 2024a. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European conference on computer vision*, pages 38–55. Springer.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. 2024b. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE.

Ji Qi, Ming Ding, Weihan Wang, Yushi Bai, Qingsong Lv, Wenyi Hong, Xu Bin, Lei Hou, Juanzi Li, Yuxiao Dong, and 1 others. 2025. Cogcom: A visual language model with chain-of-manipulations reasoning. In *International Conference on Learning Representations*, volume 2025, pages 11090–11110.

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, and 1 others. 2025. Sam 2: Segment anything in images and videos. 2025:28085–28128.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024b. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. 2019. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and 1 others. 2025a. Openthinkimg: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*.

Zhaochen Su, Peng Xia, Hangyu Guo, Zhenhua Liu, Yan Ma, Xiaoye Qu, Jiaqi Liu, Yanshu Li, Kaide Zeng, Zhengyuan Yang, and 1 others. 2025b. Thinking with images for multimodal reasoning: Foundations, methods, and future frontiers. *arXiv preprint arXiv:2506.23918*.

Zhongwei Wan, Zhihao Dou, Che Liu, Yu Zhang, Dongfei Cui, Qinjian Zhao, Hui Shen, Jing Xiong, Yi Xin, Yifan Jiang, and 1 others. 2026. Srpo: Enhancing multimodal llm reasoning via reflection-aware reinforcement learning. *Advances in Neural Information Processing Systems*, 38:153676–153713.

Haozhe Wang, Chao Qu, Zuming Huang, Wei Chu, Fangzhen Lin, and Wenhu Chen. 2026. Vl-rethinker: Incentivizing self-reflection of vision-language models with reinforcement learning. *Advances in Neural Information Processing Systems*, 38:30865–30891.

Haozhe Wang, Alex Su, Weiming Ren, Fangzhen Lin, and Wenhu Chen. 2025a. Pixel reasoner: Incentivizing pixel-space reasoning with curiosity-driven reinforcement learning. *arXiv preprint arXiv:2505.15966*.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025b. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. 2025c. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7907–7915.

Yubo Wang, Xiang Yue, and Wenhu Chen. 2025d. Critique fine-tuning: Learning to critique is more effective than learning to imitate. *arXiv preprint arXiv:2501.17703*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Mingyuan Wu, Meitang Li, Jingcheng Yang, Jize Jiang, Kaizhuo Yan, Zhaoheng Li, Hanchao Yu, Minjia Zhang, and Klara Nahrstedt. 2025. Aha moment revisited: Are vlms truly capable of self verification in inference-time scaling? *arXiv preprint arXiv:2506.17417*.

Penghao Wu and Saining Xie. 2024. V*: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13084–13094.

Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. 2025. Llava-cot: Let vision language models reason step-by-step. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2087–2098. IEEE.

A. Yang and 1 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Shuo Yang, Yuwei Niu, Yuyang Liu, Yang Ye, Bin Lin, and Li Yuan. 2026. Look-back: Implicit visual refocusing in mllm reasoning. 40(14):11694–11702.

Huanjin Yao, Jiaxing Huang, Wenhao Wu, Jingyi Zhang, Yibo Wang, Shunyu Liu, Yingjie Wang, Yuxin Song, Haocheng Feng, Li Shen, and 1 others. 2026. Mulberry: Empowering mllm with o1-like reasoning and reflection via collective monte carlo tree search. *Advances in Neural Information Processing Systems*, 38:29918–29952.

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*.

Mingliang Zhai, Hansheng Liang, Xiaomeng Fan, Zhi Gao, Chuanhao Li, Che Sun, Xu Bin, Yuwei Wu, and Yunde Jia. 2025. Multi-step reasoning for embodied question answering via tool augmentation. *arXiv preprint arXiv:2510.20310*.

Di Zhang, Jingdi Lei, Junxian Li, Xunzhi Wang, Yujie Liu, Zonglin Yang, Jiatong Li, Weida Wang, Suorong Yang, Jianbo Wu, and 1 others. 2025a. Critic-v: Vlm critics help catch vlm errors in multimodal reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9050–9061. IEEE.

Xiaowen Zhang, Zhi Gao, Licheng Jiao, Lingling Li, and Qing Li. 2026a. Stvg-r1: Incentivizing instance-level reasoning and grounding in videos via reinforcement learning. *arXiv preprint arXiv:2602.11730*.

Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and 1 others. 2025b. Adaptive chain-of-focus reasoning via dynamic visual search and zooming for efficient vlms. *arXiv preprint arXiv:2505.15436*.

Xintong Zhang, Xiaowen Zhang, Jingrong Wu, Zhi Gao, Shilin Yan, Zhenxin Diao, Kunpeng Gao, Xuanyan Chen, Yuwei Wu, Yunde Jia, and 1 others. 2026b. Adaptmmbench: Benchmarking adaptive multimodal reasoning for mode selection and reasoning process. *arXiv preprint arXiv:2602.02676*.

Yi-Fan Zhang, Xingyu Lu, Shukang Yin, Chaoyou Fu, Wei Chen, Xiao Hu, Bin Wen, Kaiyu Jiang, Changyi Liu, Tianke Zhang, and 1 others. 2025c. Thyme: Think beyond images. *arXiv preprint arXiv:2508.11630*.

YiFan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, and 1 others. 2025d. Mme-realworld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? 2025:89655–89701.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. Llamafactory: Unified

efficient fine-tuning of 100+ language models. pages 400–410.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. 2025. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A Dataset Setting

A.1 Data Sources

We construct a high-quality visual reflection instruction-tuning dataset comprising approximately 24k samples. The data sources are curated from diverse public benchmarks, including COCO (Lin et al., 2014), LLaVA (Liu et al., 2023), GQA (Hudson and Manning, 2019), TextVQA (Singh et al., 2019), OCRVQA (Mishra et al., 2019), DOCVQA (Mathew et al., 2021), DVQA (Kafle et al., 2018), and ChartQA (Masry et al., 2022). These samples cover a wide range of visual domains and reasoning complexities, serving as the foundation for our fine-tuning process.

For benchmarks that appear in both our training data and evaluation (TextVQA and ChartQA), we strictly use the training splits for ReflectV construction and the validation/test splits for evaluation, following the standard protocol of each benchmark. Specifically, TextVQA-Val and ChartQA-Test are held out for evaluation and never seen during training.

To ensure balanced capability acquisition, we categorize these sources into four primary domains: General QA, Scene Text (OCR), Document Understanding, and Chart Reasoning. As illustrated in Figure 6(a), this domain-specific distribution is strategically balanced to retain generalizability while boosting fine-grained perception. Furthermore, the statistics regarding the data filtering process and trajectory depth are detailed in Figure 6(b) and (c), respectively.

A.2 Data Construction Pipeline

In the multi-turn dialogue construction stage (Section 4.1), GPT-4o (Hurst et al., 2024) serves as the teacher model that generates the multi-turn corrective feedback and quality scores. Specifically, we employ Qwen2.5-7B (Yang et al., 2024) to generate the initial textual self-reflection chains and extract keywords. Molmo-7B (Deitke et al., 2025) is then employed to generate precise point coordinates for the mentioned keywords, which serve as prompts for SAM 2 (Ravi et al., 2025) (optional) to produce fine-grained segmentation masks. Finally, Qwen2.5-VL-7B (Bai et al., 2025b) acts as a visual verifier to ensure the generated cues are semantically aligned with the image content.

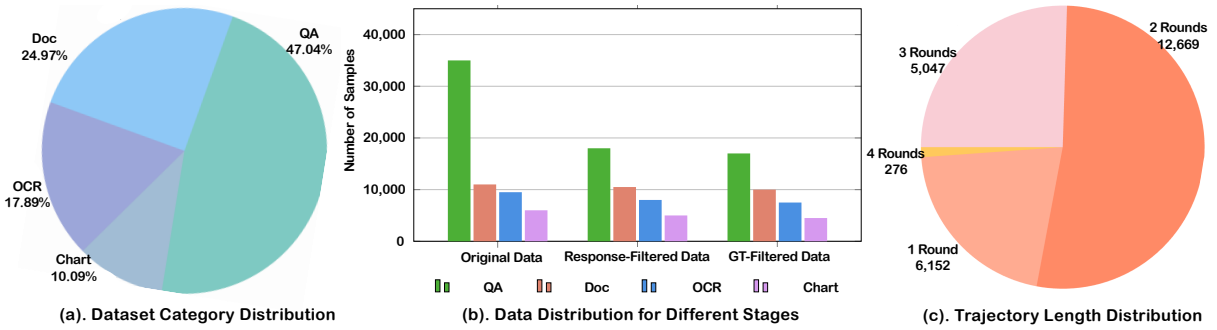


Figure 6: **Statistics and distribution of the ReflectV dataset construction.** (a). **Domain Distribution:** The composition of the raw data spans four distinct capabilities: General QA, Document Understanding (Doc), Scene Text (OCR), and Chart Reasoning. (b). **Filtering Pipeline:** The data volume retention across the three construction stages (Original $\rightarrow$ Response-Filtered $\rightarrow$ GT-Filtered), illustrating the rigorous quality control process described in Section 4.1. (c). **Trajectory Depth:** We set the default mixing ratio to $\rho = 0.75$ for our training. The distribution of samples based on the number of self-reflection rounds required for convergence illustrates the varying complexity of error correction.

A.3 Data Filtering Strategy

To demonstrate the effectiveness of our filtering strategy and the high quality of our data, we conduct a comparative quality scoring of the filtered, unfiltered, and discarded subsets, and further validate our approach by benchmarking our model against a baseline trained on raw, unfiltered data.

First, we conduct a blind quality assessment involving both human experts and GPT-4o-mini. We randomly sample 100 trajectories from three subsets: the discarded noisy data (D_{noise}), the raw unfiltered data (D_{raw}), and our final filtered data (D_{clean}). Each sample is scored on a scale of 1 to 5 based on Logical Coherence and Visual Validity.

Table 8: Comparative quality assessment. We evaluate samples from Discarded (D_{noise}), Unfiltered (D_{raw}), and Filtered (D_{clean}) subsets. The results demonstrate that our pipeline effectively isolates high-quality trajectories.

Dataset	Human		GPT-4o-mini		Average	
	Logic	Visual	Logic	Visual	Logic	Visual
D_{noise}	2.14	1.38	2.30	1.45	2.22	1.42
D_{raw}	3.56	3.12	3.65	3.20	3.61	3.16
D_{clean}	**4.78**	**4.65**	**4.82**	**4.70**	**4.80**	**4.68**

As reported in Table 8, D_{clean} consistently achieves the highest scores (Avg. 4.74), significantly surpassing the raw baseline (D_{raw}, 3.39). Conversely, the discarded subset (D_{noise}) exhibits the lowest performance, particularly in visual validity (1.42). This distinct stratification confirms that our pipeline effectively identifies and prunes samples with hallucinated or ungrounded visual prompts, ensuring high-quality training data. For more visualizations of D_{noise}, please refer to Figure 8.

Second, we construct a baseline named MIRROR-Raw. This variant is fine-tuned on the unrefined dataset ($\sim$35k samples) obtained before our multi-turn dialogue data filtering and visual verification stage, containing noisy instances where visual reflection is either absent or fails verification.

Table 9: Ablation study on data filtering strategy conducted on MM-Vet. We compare MIRROR-Raw with MIRROR-1.0 and our final MIRROR-0.75 ($\rho = 0.75$). The best and second-best results are highlighted in **bold** and underlined, respectively.

Model	OCR	Math	Spat	Gen	Rec	Overall
MIRROR-Raw	56.39	41.54	55.20	38.63	55.88	55.87
MIRROR-1.0	<u>65.19</u>	<u>56.54</u>	<u>61.47</u>	<u>54.12</u>	<u>59.04</u>	<u>58.81</u>
MIRROR-0.75	**69.90**	**60.77**	**70.80**	**61.62**	**64.40**	**66.70**

As presented in Table 9, we assess the efficacy of our visual consistency filtering by comparing the unrefined baseline, MIRROR-Raw, against MIRROR-1.0 ($\rho = 1.0$). It is crucial to note that both variants share an identical structural composition—consisting exclusively of multi-turn self-reflection trajectories ($\rho = 1.0$)—which isolates data quality as the sole variable. Despite MIRROR-Raw utilizing a significantly larger corpus ($\sim$35k samples), it underperforms the filtered MIRROR-1.0 ($\sim$24k samples) across all metrics.

Specifically, MIRROR-Raw lags behind the structure-aligned MIRROR-1.0 (55.87 vs. 58.81) and exhibits a substantial deficit compared to our optimal hybrid model, MIRROR-0.75 (55.87 vs. 66.70). This massive performance gap confirms

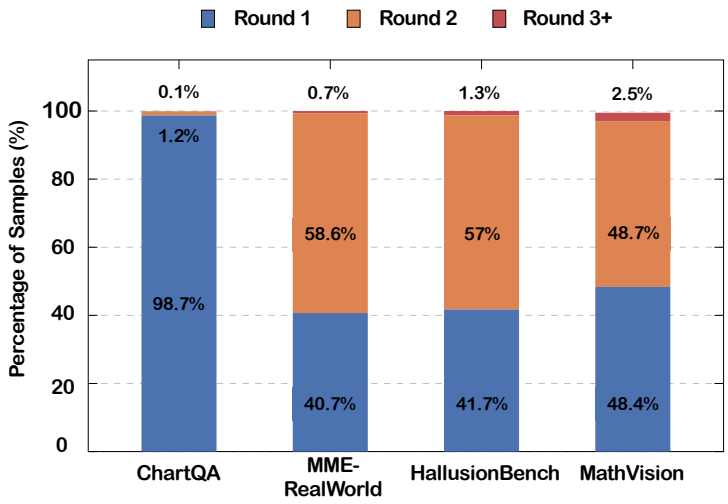


Figure 7: **Distribution of inference trajectory lengths across representative benchmarks.** The stacked bars illustrate the percentage of samples resolved within 1, 2, or 3+ rounds.

that data purity is paramount. Retaining ∼11k noisy samples—where visual grounding is hallucinated or absent—introduces conflicting signals that severely hamper the model's potential. In contrast, removing this noise enables the precise optimization seen in MIRROR-0.75.

B Implementation Details

B.1 Training Details.

We utilize LLaMA-Factory (Zheng et al., 2024) to perform Supervised Fine-Tuning (SFT) on the Qwen2.5-VL-7B (Bai et al., 2025b). The training is conducted on 2 × NVIDIA H100 GPUs for 3.0 epochs. To ensure training efficiency and memory optimization, we utilize the DeepSpeed ZeRO-3 optimization and employ Low-Rank Adaptation (LoRA) (Hu et al., 2022) for parameter-efficient fine-tuning. The LoRA rank is set to $r = 32$ with the scaling factor $\alpha = 128$. Training setup of Qwen3-VL-8B (Bai et al., 2025a) follows Qwen2.5-VL-7B.

B.2 Inference Details.

We perform all evaluations using the SFT-adapted MIRROR model, and we use the same hardware settings in all experiments. To ensure high-quality visual grounding, the visual prompt generator is powered by Molmo-7B (Deitke et al., 2025), which predicts precise point coordinates based on the model's textual reflection. For mask generation, we employ the SAM 2.1 (Ravi et al., 2025) model, specifically loading the large-scale checkpoint `sam2.1_hiera_l.pt` to ensure robust segmentation performance even in complex visual scenes. Note that SAM 2.1 is an optional module used only to convert point coordinates into masks for mask-style prompts; point-style prompts do not rely on it.

Specifically, a verification flag dictates whether visual tools are invoked at each turn. In cases of tool failure or empty returns, the original image is passed forward, allowing MIRROR to gracefully default to standard one-pass reasoning.

B.3 Trajectory Length Analysis.

To understand the model's dynamic behavior, we conduct a statistical analysis of the inference trajectories across evaluated benchmarks, as illustrated in Figure 7. We observe that the model adaptively adjusts its reasoning depth based on the inherent difficulty of the task.

- Efficiency in information extraction. For straightforward queries where visual evidence is explicit, the model tends to converge efficiently. For instance, on ChartQA, which primarily involves direct value extraction, 98.7% of samples are resolved within a single round, minimizing unnecessary computational overhead.

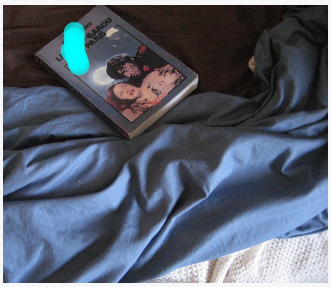
- Rigor in complex perception. In contrast, for tasks requiring fine-grained verification or hallucination mitigation, the reasoning trajectory becomes significantly longer. On MME-RealWorld and HallusionBench, the model engages in a second round of visual reflection for 58.6% and 57.0% of cases, respectively. Notably, MathVision exhibits the highest proportion of trajectories exceeding three rounds

Question: What kind of book might one expect to read in this scene, and what is its theme?



<answer>
One might expect to read a romance novel in this scene. The theme could be centered around love and relationships, as suggested by the cover image of a man and a woman.
<answer>
<think>
I believe my answer correctly identifies that the book has a French title and that the bed is unmade, suggesting recent use. However, I realize the theme of the book is not accurately identified. The cover image and the title 'Le Loup-Garou de Paris' hint at a different genre. I need to consider the elements on the cover more closely, as indicated by the cyan points in the image, to determine the correct theme.
</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'French title, unmade bed, urban horror theme', 'color': 'cyan', 'mark_type': 'point'}}
</tool_call>



<answer>
One might expect to read a horror novel in this scene. The theme could be centered around werewolves and supernatural events, as suggested by the title and cover image.
</answer>
<think>
Much better! I correctly identified that the book is likely a horror novel and that its theme involves werewolves and supernatural events. I could improve my answer by mentioning the specific elements of fear and suspense that are typical of the horror genre, and how the unmade bed, indicated by the magenta ellipses in the image, might suggest the book was recently read or set down.
</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'unmade bed, werewolves, fear, suspense novel', 'color': 'magenta', 'mark_type': 'ellipse'}}
</tool_call>

Continued on next page.

(a) **Part 1 of the case from D_{noise}.**

Continuation (continued):

<answer>
In this scene, one would expect to read a horror book with a French title. The book is placed on an unmade bed, suggesting that it might have been recently picked up, read, or set down. Its theme is centered around werewolves, indicating that the story includes elements of fear, suspense, and potentially supernatural creatures, typical of a horror genre. One may find this book to be thrilling and engaging, particularly if they are a fan of horror or paranormal stories.
</answer>
<think>
There is high consistency between my answer, image details and visual prompts, complete details and clear logic.
</think>



```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': false, 'anchor':
'None', 'color': 'None', 'mark_type': 'None'}}
</tool_call>
```

(b) **Part 2 of the case from D_{noise}.**

Figure 8: Case 1 from D_{noise}.

(2.5%), reflecting the need for iterative correction in multi-step logical derivation.

Overall, with 80.2% of total samples resolved in the first turn, this distribution confirms that MIRROR effectively balances efficiency and rigor, engaging in deep visual reflection only when the complexity of the query demands it.

B.4 Multi-turn Reflection Outcome Analysis

To provide a fine-grained understanding of how multi-turn visual reflection affects prediction quality, we analyze the triggered multi-turn subset of MM-Vet samples for which MIRROR enters at least one additional verification round. This subset accounts for 19.8% of all MM-Vet samples, indicating that the model invokes reflection selectively rather than forcing multi-turn reasoning on every query. For a controlled comparison, we evaluate both loop-enabled models on this same triggered subset, while single-pass models (Qwen2.5-VL-7B, MIRROR w/o loop, MIRROR-Tools) are excluded because they do not produce correction trajectories.

We categorize each multi-turn sample into one of four outcome types based on the correctness transition between the initial response (Round 1) and the final response:

- **Maintained Correct** (C→C): The initial response was correct and remained correct after reflection.
- **Corrected** (W→C): The initial response was wrong, but multi-turn visual reflection successfully identified the error and produced the correct answer.
- **Degraded** (C→W): The initial response was correct, but reflection incorrectly changed it to a wrong one.
- **Uncorrected** (W→W): The initial response was wrong, and reflection failed to correct it.

Table 10: Reflection outcome analysis on the triggered multi-turn subset of MM-Vet. Distribution of correctness transitions (%) on the triggered multi-turn subset of MM-Vet.

Method	**C→C**	**W→C ↑**	**C→W ↓**	**W→W ↓**
Qwen2.5-VL-7B-MIRROR	28.0	28.5	15.0	28.5
MIRROR (ours)	**25.0**	**69.0**	**3.5**	**2.5**

Table 10 details the correctness transition distribution within the triggered multi-turn subset on MM-Vet. Compared to the few-shot baseline, trained MIRROR exhibits a closed-loop verification policy. First, it demonstrates precise reflection triggering: the vast majority of its triggered sessions (**71.5**% total) target initially incorrect samples (W → C and W → W). Within this critical

subset, MIRROR achieves an outstanding error-correction rate of $\mathbf{69.0}\%$, effectively turning failure cases into correct outputs. Second, our framework maintains superior multi-turn robustness, heavily suppressing harmful degradation where C $\rightarrow$ W transitions plummet from 15.0% to just $\mathbf{3.5}\%$. This confirms that MIRROR successfully learns a highly reliable, evidence-driven policy that preserves initially correct answers while rectifying visual misalignments.

To further understand the remaining failure cases (6.0% of the triggered subset), we identify three contributing factors: error misdiagnosis—self-reflection pinpoints correct elements as errors, imprecise visual grounding—the visual prompt generator produces misleading spatial markers, and persistent reasoning errors—both reflection and visual prompts are correct but the VLM still fails to derive the right answer. We quantify each factor's contribution using the few-shot baseline. For C$\rightarrow$W (3.5%), the initial answer was already correct, so errors stem solely from misdiagnosis or imprecise grounding. The baseline C$\rightarrow$W of 15.0% is reduced by 76.7%, confirming that learned reflection suppresses false positive triggers, with the residual 3.5% dominated by misdiagnosis errors. For W$\rightarrow$W (2.5%), all three factors can contribute. The baseline W$\rightarrow$W of 28.5% is reduced by 91.2%, demonstrating that training collectively addresses all factors, with persistent reasoning errors likely accounting for the largest residual share as they are hardest to eliminate. Overall, the total triggered-subset error drops from 43.5% to 6.0%—an 86.2% relative reduction—meaning only 1.2% of all MM-Vet samples are affected by any reflection failure, confirming MIRROR's robustness.

C Prompt

C.1 Prompt for Dataset Construction

To construct the training data for the ReflectV dataset, we employ a multi-stage prompting strategy to transform raw annotations into grounded, first-person self-reflection trajectories.

First, we generate task-specific visual cues. For data-intensive tasks like Charts, OCR, and Doc, we utilize a specific prompt to extract key subjects and entities as shown in Figure 9. For general QA tasks, we employ a two-step process: in Figure 10, first identifying key physical objects mentioned in the feedback, and subsequently merging them into a concise caption. Next, we use a caption insertion prompt, as shown in Figure 11, to seamlessly blend these visual cues (e.g., specific marks or coordinates) into the corrective feedback, ensuring the text is visually grounded. Finally, in Figure 12, to simulate an internal self-correction process, we apply a role conversion prompt that transforms the content from a "teacher's evaluation" perspective into a "student's first-person reflection" (e.g., changing "Your response is wrong" to "I think my response is wrong").

C.2 Prompt for Iterative Reasoning

To equip the model with self-reflection and visual verification capabilities, we design a structured system prompt and user prompt that enforces a rigorous output protocol, as shown in Figure 13 and Figure 14.

D More Efficiency Analysis

To evaluate the practicality of MIRROR, we conduct a comparative analysis of inference efficiency on the MM-Vet dataset, employing average inference time per sample and average token consumption as metrics.

D.1 Comparison with Reasoning Models.

As detailed in Table 11, we benchmark our method against representative baselines from three paradigms: Text Reflection (VL-Rethinker), Visual Reflection (Look-Back), and Thinking with Images (PixelReasoner, DeepEyes, Adaptive-CoF).

Table 11: Efficiency comparison on the MM-Vet dataset. We report the average time cost (seconds) and token consumption per sample. **Bold** and underlined indicate the best and second-best efficiency, respectively.

Method	Time (s) ↓	Tokens ↓
Text Reflection		
VL-Rethinker	5.56	229.33
Visual Reflection		
Look-Back (Semantic)	89.34	307.90
Look-Back (Solution)	300.86	427.22
Thinking with Images		
PixelReasoner-SFT	25.03	241.60
PixelReasoner	64.12	273.95
DeepEyes	5.06	197.51
Adaptive-CoF-SFT	5.41	206.90
Adaptive-CoF	<u>4.02</u>	<u>143.51</u>
MIRROR (ours)	**3.73**	**112.58**

Compared to the Text Reflection model, our method reduces inference time by 32.9% (3.73 s vs. 5.56 s) and token consumption by 50.9% (112.58 vs. 229.33). This significant reduction suggests

Please generate a short phrase based on the given <question>, focusing on the **subject** (the core object/person/thing the question is about), the **key modifying words** (especially words related to data statistics like "total", "average", "sum", "difference", "median", "maximum", "minimum") and key associated information (such as time, category, attribute, etc) present in the question.

Requirements:

1. **Core Elements**: Must include both the clear subject from the question and the key modifying words (e.g., "total", "average", "maximum") if mentioned.

2. **Key Details**: Incorporate essential associated information (e.g., time, location, product type) related to the subject if specified.

3. **Avoid Redundancy**: Do not include interrogative words like "how many", "how much", "what" or explanatory words like "the amount of".

4. **Avoid Over-Simplicity**: Do not answer with only 1-2 words; the phrase should clearly link the modifying word, subject and context.

Examples:

Example1:

<question>: How many bikes are there in 2018?

Your answer: bikes in 2018

Example2:

<question>: What is the total revenue of Company A in 2021 and 2022?

Your answer: total revenue of Company A in 2021 and 2022

Example3:

<question>: What was the average monthly temperature in City C from 2019 to 2021?

Your answer: average monthly temperature in City C from 2019 to 2021

Example4:

<question>: What is the difference in sales volume between 2020 and 2023 for Product B?

Your answer: difference in sales volume of Product B in 2020 and 2023

Example5:

<question>: What's the maximum value in the green line?

Your answer: maximum value in the green line

Figure 9: Subject Extraction Prompt: Extracts core subjects and key modifiers from questions to form concise visual descriptions for Chart, OCR, and Doc understanding tasks.

that our targeted visual verification mechanism effectively circumvents the verbose and redundant self-correction chains often required by purely text-based reflection.

Compared to Thinking with Images models and Visual Reflection models, MIRROR significantly outperforms PixelReasoner (64.12 s), DeepEyes (5.06 s), and Look-Back (89.34 s / 300.86 s) in terms of latency. Even compared to the highly efficient Adaptive-CoF, MIRROR maintains a distinct advantage, further reducing time cost by 7.2% and token usage by 21.5%. These findings confirm that MIRROR not only enhances reasoning accuracy but also ensures an efficient inference process, making it highly suitable for practical deployment.

Table 12: Efficiency breakdown of ablation configurations on MM-Vet. Time and token costs are per sample. **Bold** and underlined indicate the best and second-best, respectively.

Configuration	Time (s) ↓	Tokens ↓
Qwen2.5-VL-7B	**2.81**	**85.50**
Qwen2.5-VL-7B-Tools	6.65	180.40
Qwen2.5-VL-7B-MIRROR	4.69	156.30
MIRROR (w/o loop)	<u>3.25</u>	<u>98.50</u>
MIRROR-Tools	5.74	161.70
MIRROR (ours)	3.73	112.58

You are given a question, a response, and feedback. Please find out the key objects in the <feedback> with the help of <question> and <response> to help refine <response>. Identify only a few concise noun phrases from <feedback> that can be localized in the image. Return only the following format, with no additional text: <object1>|<object2>|...|<objectn>.

Example 1:

<question>: "What rare species are depicted in the wildlife photograph?"

<response>: "Thought: I see some grey animals along with another creature I can not recognize. Answer: Just some grey animals."

<feedback>: "The image features grey cats and a white spotted leopard with piercing eyes, lying on a rocky outcrop amidst snow-capped peaks."

Your Answer: grey cats|a white spotted leopard.

Example 2:

<question>: "What is in the glass on the left?"

<response>: "Thought: I can see the left glass but I don't know what is in the glass. Answer: Sorry, I don't know."

<feedback>: "The liquid in the left glass is yellow which is similar to some beverages."

Your Answer: left glass|yellow liquid.

You are given feedback<feedback> made of some key objects in this format:

<object1>|<object2>|...|<objectn>. Please help me combine them into a short caption. Remember only return a short caption of no more than 10 words.

For example:

<feedback> is "grey cats\a white spotted leopard."

Your Answer: grey cats and a white spotted leopard.

<feedback> is "left glass|yellow liquid."

Your Answer: yellow liquid in left glass.

Figure 10: Visual Caption Generation Prompt: A two-step pipeline that first identifies key physical objects from feedback and then synthesizes them into short descriptive captions for QA tasks.

D.2 Component Ablation Comparison.

Table 12 further breaks down the efficiency of each ablation configuration introduced in the main paper. Compared to the base model (Qwen2.5-VL-7B), the full MIRROR system adds only 0.92 s and 27.08 tokens per sample, demonstrating that the closed-loop verification process incurs a modest overhead while delivering substantial accuracy gains (+10.10 on MM-Vet). Notably, MIRROR (w/o loop) achieves the second-best efficiency (3.25 s, 98.50 tokens), suggesting that even without external visual tools, the trained verification policy alone is highly efficient.

E More Qualitative Visualization

Beyond quantitative metrics, we provide comprehensive qualitative comparisons across diverse complex reasoning scenarios.

Comparison with base model. Figures 15 and 16 illustrate the reasoning trajectories of our model versus the base model Qwen2.5-VL-7B, highlighting our model's ability to self-correct.

Comparison with competitive baselines. We further compare MIRROR with two representative reasoning paradigms: (i) the "Thinking with Images" approach (PixelReasoner, DeepEyes, Adaptive-CoF), and (ii) the textual self-reflection approach (VL-Rethinker). As illustrated in Figures 17 to 20, MIRROR demonstrates superior grounding and error correction capabilities.

You are given **feedbacks**<feedback>, **marks**<mark> and **captions**<caption>. The captions help to show some **marks**<mark> on the image. **Please insert the captions along with the marks into the feedbacks and ensure that the new information blends seamlessly with the original content, maintaining the overall flow and coherence of the feedbacks. BUT DO NOT show the** <groundtruth> **and answer in a diverse way. If** <number> **is 1**, you may use word like "a red point" or "the red point". **But if** <number> **is more than 1**, you may use word like "red points" or "the red points"

Example:
<feedback>: "Your observation that the flower is vibrant is partly right. However, the flower is not red. Take a closer look at the petals and think if there is another bright color that more accurately describes it."
<mark>: "green point"
<caption>: "petals"
<number>: 1

Your answer can be one of the following examples:

Example 1: "Your observation that the flower is vibrant is partly right. However, the flower is not red. Take a closer look at the petals and think if there is another bright color that more accurately describes it and can be seen with a green point in the <image>."

Example 2: "Your observation that the flower is vibrant is partly right. However, the flower is not red. Take a closer look at the petals, as indicated by the green point in the <image>, and think if there is another bright color that more accurately describes it."

Example 3: "Your observation that the flower is vibrant is partly right. However, the flower is not red. Check the petals closely, the area highlighted by a green point in the <image>, and consider if there is another bright color that more accurately describes it."

Example 4: "Your observation that the flower is vibrant is partly right. However, the flower is not red. Inspect the petals, noted by the green point in the <image>, and think if there is another bright color that more accurately describes it."

Figure 11: Caption Insertion Prompt: Seamlessly integrates generated visual captions and spatial marks (e.g., points) into textual feedback to ensure visual grounding.

F Limitation

To provide a comprehensive understanding of MIRROR's boundaries, we present qualitative examples of failure cases in the figures below. These cases correspond to the two primary limitations discussed in the Limitations section: the difficulty in grounding abstract concepts and the lack of granularity in complex attribute binding.

Limited applicability in abstract domains. Despite these improvements, MIRROR still faces challenges in symbolic derivation involving complex spatial mapping. Figure 21 illustrates a geometry problem requiring the calculation of a circle's radius using the Pythagorean theorem. While the model correctly performs the symbolic derivation in the text, its attempt to invoke the visual prompt generator is ineffective. The model tries to ground the concept of "diameter calculation and radius division" into visual points. However, mathematical operations and logical variables are abstract concepts that lack a direct spatial mapping in the 2D image plane. Consequently, the generated visual cues (red points) provide no informative feedback for verifying the correctness of the arithmetic steps.

Coarse-grained attribute binding. Furthermore, MIRROR encounters difficulties in fine-grained attribute binding under compositional constraints. As shown in Figure 22, the user queries for a specific flower defined by a compositional constraint: "five petals and three leaves." Because the visual tool fails to strictly enforce these count-based attributes, the model incorrectly selects option E. Instead of isolating the specific target, the tool generates coarse visual markers (green points) on all flower instances in the image. This indicates that the visual prompt generator struggles to distinguish complex attribute combinations, defaulting to a generic object detection behavior (detecting "flowers") rather than precise constraint verification.

F.1 Quantitative Failure Analysis

To complement the qualitative cases above and to quantify how errors propagate through the closed-loop pipeline, we manually examine all 846 failed, tool-triggered trajectories, which account for 4.7% of the full evaluation set, and classify each case according to the earliest incorrect stage.

You are given a **SENTENCE**. Please rewrite it to adopt a **first-person perspective** that reflects personal thinking or a subjective statement (utilizing "I" or "we") and diversify your answer. Your answer **MUST** be written in the **first-person perspective**. If the **SENTENCE** start with words like "Score: 1", "Feedback:", please remove these words like Example 2,3,4,5.

For example:

Example 1

SENTENCE: Your response is wrong. The data for "Q3" shows a higher growth rate than "Q4". Check the bar chart again, especially the blue bars representing quarterly growth, and pick the quarter with the highest growth.

Your answer: I think my response is wrong. The data for "Q3" shows a higher growth rate than "Q4". Check the bar chart again, especially the blue bars representing quarterly growth, and pick the quarter with the highest growth.

Example 2

SENTENCE: Score: 3. Feedback: Correct! Your conclusion is wrong. The "Online Sales" segment outperforms "In-Store Sales" this quarter. Check the pie chart again, especially the blue section for online data, and confirm the top-performing segment.

Your answer: I believe my conclusion is wrong. The "Online Sales" segment outperforms "In-Store Sales" this quarter. Check the pie chart again, especially the blue section for online data, and confirm the top-performing segment.

Example 3

SENTENCE: Score: 8. Your conclusion is incorrect. The plant grows taller under "Full Sun" than "Partial Shade". Refer to the growth chart again, marked by the red lines for each light condition, and identify the optimal environment.

Your answer: I suspect my conclusion isn't correct. The plant grows taller under "Full Sun" than "Partial Shade". Refer to the growth chart again, marked by the red lines for each light condition, and identify the optimal environment

---

Example 4

SENTENCE: Feedback: Correct! Your conclusion is inaccurate. The "18-25 Age Group" has higher engagement than "26-35". Review the user data again, specifically the green bars for engagement rates, and name the most engaged group.

Your answer: I'm starting to think my conclusion is inaccurate. The "18-25 Age Group" has higher engagement than "26-35". Review the user data again, specifically the green bars for engagement rates, and name the most engaged group.

Example 5

SENTENCE: Score: 6. Feedback: Correct! Your conclusion is mistaken. The chemical "Compound B" has a longer shelf life than "Compound A". Look at the stability test results again, as shown by the purple timelines in the table, and confirm the more stable compound.

Your answer: I fear my conclusion may be mistaken. The chemical "Compound B" has a longer shelf life than "Compound A". Look at the stability test results again, as shown by the purple timelines in the table, and confirm the more stable compound.

Figure 12: Self-Reflection Conversion Prompt: Rewrites teacher-side evaluations into first-person student self-reflections (e.g., changing "Your response is wrong" to "I think my response is wrong").

Table 13: Stage-wise failure analysis of the 846 failed tool-triggered trajectories (4.7% of the full evaluation set). Each case is assigned to the earliest incorrect stage.

Failure Stage	Description	Percentage
Incorrect anchor	Wrong, ambiguous, or incomplete visual target	22.5%
Incorrect grounding	Correct target but incorrect localization	33.8%
Incorrect reasoning	Correct target and localization but incorrect revision	43.7%

As reported in Table 13, grounding-related errors (incorrect anchor and incorrect grounding) account for 56.3% of the failures, while the dominant stage is incorrect reasoning (43.7%), where the visual evidence is correctly identified and localized but the model still fails to derive the correct answer. This indicates that most failures stem from the reasoning capacity of the backbone VLM rather than from error propagation of the external grounding tool. Note that SAM 2.1 is only used to convert point coordinates into masks for mask-style prompts, so it is not involved in this analysis.

Furthermore, focusing on the 655 failure cases from the incorrect grounding and incorrect reasoning stages, we categorize each case by failure type, as shown in Table 14. Attribute binding refers to failures involving multiple objects, attributes, counts, or relations; abstract reasoning covers symbolic or logical steps that cannot be directly mapped to a visual region; Others includes the remaining failures, such as directly localizable perception errors and general answer-revision errors. Attribute binding (42.1%) and abstract reasoning (19.7%) are precisely the two limitations discussed above, providing a quantitative foundation for the capability boundary of MIRROR.

You are an intelligent assistant with self-reflection and visual verification capabilities.

Core Task:

1. First, provide a direct answer to the user's image-related query based on your initial observation.

2. Second, engage in a "post-hoc reflection" to strictly evaluate the accuracy of your initial answer and analyze visual evidence.

3. Third, determine if visual grounding tools are needed to verify or correct your answer.

Follow these rules strictly:

1. Output Structure & Order:

You must output exactly three parts in this specific order:

<answer>Your Initial Answer</answer>

<think>Your Reflection & Visual Analysis</think>

<tool_call>JSON Function Call</tool_call>

---

2. Reflection Requirements (<think>):

- In <think>, you must critique your <answer>. Ask yourself: "Is this answer grounded in the image? Did I miss details?"

- You must explicitly identify specific visual regions that support or contradict your answer.

- Describe these regions using specific mark types (arrow/point/circle/line/ellipse/mask/bbox) and colors.

- **CRITICAL:** The text description of the visual object you analyze here MUST match the 'anchor' used in the tool call later.

---

3. Tool Call & Consistency:

- Use the `visual_prompt_generator` function.

- flag: Set to `True` if you detect errors, ambiguity, or need verification. Set to `False` ONLY if the answer is completely correct and verified.

- anchor: This string MUST be consistent with the visual description provided in the <think> section.

- mark_type & color: Must match the visualization logic described in <think>.

---

4. Function Signature:

Continued on next page.

(a) **Part 1 of system prompt.**

Function Signature (continued):

```
<tool_call>
{
  "type": "function",
  "function": {
    "name": "visual_prompt_generator",
    "description": "Generates a visual overlay to verify specific image regions
identified during reflection.",
    "parameters": {
      "properties": {
        "flag": {
          "type": "boolean",
          "enum": [True, False],
          "description": "True if the answer needs verification/correction; False
if the answer is final and correct."
        },
        "anchor": {
          "type": "string",
          "description": "The exact text description of the visual element
analyzed in <think> (e.g., 'red sports car', 'y-axis label')."
        },
        "color": {
          "type": "string",
          "enum": ["red", "green", "blue", "cyan", "magenta", "pink", "orange",
"purple", "gold", "brown"],
          "description": "Color of the visual prompt."
        },
        "mark_type": {
          "type": "string",
          "enum": ["arrow", "point", "circle", "line", "ellipse", "mask",
"bbox"],
          "description": "Geometric shape of the visual prompt."
        }
      },
      "required": ["flag", "anchor", "color", "mark_type"],
      "type": "object"
    }
  }
}
</tool_call>
```

Output Format Example:

<answer>The car is red.</answer><think>I stated the car is red. Looking closer at the hood of the car highlighted by the red bbox, I see reflections that might suggest it's actually orange. I need to verify the color of the car.</think><tool_call>{"name": "visual_prompt_generator", "arguments": {"flag": true, "anchor": "hood of the car", "color": "red", "mark_type": "bbox"}}</tool_call>

(b) **Part 2 of system prompt.**

Figure 13: System prompt during training.

Remember: Output your Answer first, then Think, and finally Tool Call.

Strict format: <answer>...</answer><think>...</think><tool_call>...</tool_call>

Figure 14: User prompt during training.

Question: What is the color of the car?

Choices: (A) white (B) red (C) blue (D) green

GT: (B) red



Qwen2.5-VL-7B:

The image does not contain a car. It features a serene landscape with a lake, forest, and mountains. Therefore, there is no car to determine its color. The correct answer is: **None of the above**.

MIRROR:

<answer>

The car is red.

</answer>

<think>

I believe my answer is correct. The car is indeed red. However, I could have provided more detail to support my answer. For example, I could mention the specific shade of red and how it stands out against the background.

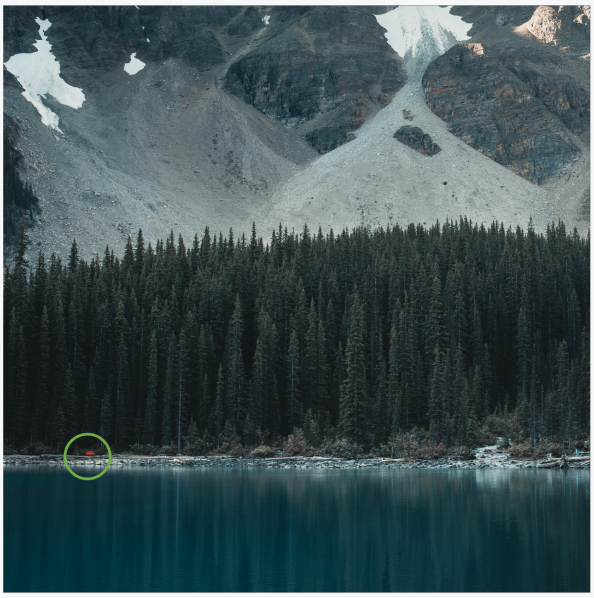
</think>

```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor':
'red car standing out against the background', 'color': 'green', 'mark_type':
'circle'}}
</tool_call>
```

Continued on next page.

(a) **Part 1 of comparison case 1 with Qwen2.5-VL-7B.**

Continuation (continued):



The green circle indicates the red car.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer>...</answer> <think>... </think><tool_call>...</tool_call>

<answer>

The car is red.

</answer>

<think>

I accurately identified the core activity, designed precise visual prompts, and achieved ideal effect after iterative reflection.

</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': false, 'anchor': 'None', 'color': 'None', 'mark_type': 'None'}}
</tool_call>

(b) **Part 2 of comparison case 1 with Qwen2.5-VL-7B.**

Figure 15: Comparison case 1 with Qwen2.5-VL-7B.

Question: Is there a total of two dogs in the image? Please answer yes or no.



GT: Yes.

Qwen2.5-VL-7B:

No, there are not two dogs in the image. There appear to be three dogs lying on the bed.

MIRROR:

<answer>

No, there are more than two dogs in the image. </answer>

<think>

I believe my answer is partially correct in that there are multiple dogs in the image, but I need to count them more carefully. Look closely at the image again and try to count the number of dogs present.

</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'multiple dogs in the image against the background', 'color': 'cyan', 'mark_type': 'point'}}
</tool_call>



The cyan points indicate multiple dogs in the image against the background.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer>...</answer> <think>... </think><tool_call>...</tool_call>

<answer>

Yes, there are two dogs in the image. </answer>

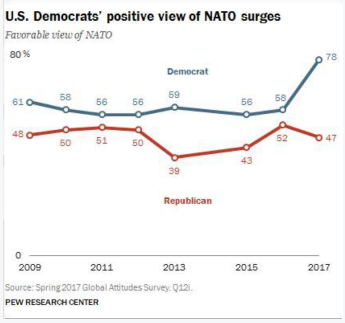
<think>

I accurately identified the core activity, designed precise visual prompts, and achieved ideal effect after iterative reflection.

</think>

Figure 16: Comparison case 2 with Qwen2.5-VL-7B.

Question: How many points have 56 value in blue graph?



GT: 3

VL-Rethinker-7B:

To determine how many points have a value of 56 in the blue graph, let's follow these steps:
1. **Identify the blue graph**: The blue graph appears to be a line graph with points plotted at various values on the x-axis and y-axis.
2. **Locate the value 56 on the y-axis**: Find where the y-axis value is 56.
3. **Count the points**: Look for the points on the graph that align with the y-axis value of 56.
By examining the graph, it appears that there are two points where the y-value is 56.
Therefore, **the number of points that have a value of 56 in the blue graph is 2**.

PixelReasoner-7B:

To determine how many points have a value of 56 in the blue graph, we need to follow these steps:

1. Identify the blue graph: The blue graph represents the "Democrat" data. 2. Locate the value 56 on the y-axis: The y-axis is labeled "Favorable view of NATO" and ranges from 0% to 80%. 3. Find the points on the blue graph that correspond to the value 56: We need to look for the horizontal lines that intersect the y-axis at 56%.

By examining the graph: - The blue graph has a point at 56% in 2011. - The blue graph has a point at 56% in 2013.

There are no other points on the blue graph that have a value of 56%.

Therefore, **the number of points with a value of 56 in the blue graph is 2**.

DeepEyes-7B:

<think> To determine how many points have a value of 56 in the blue graph, I need to examine the data points for the "Democrat" line. The blue graph shows the favorable view of NATO for Democrats over the years. I can see that the value 56 appears at two points: in 2011 and 2013. </think>

<image_zoom_in_tool> Upon zooming in, the blue graph shows two points with a value of 56 in the years 2011 and 2013. </image_zoom_in_tool>

<answer> **There are 2 points that have a value of 56 in the blue graph.** </answer>

Continued on next page.

(a) **Part 1 of comparison case 1 with reasoning paradigms.**

Continuation (continued):

Adaptive-CoF:

<think> To determine how many points have a 56 value in the blue graph, I need to analyze the graph carefully. The blue line represents the "Democratic" view of NATO. I observe that the blue line has two points marked with the value 56. One point occurs in 2011 and the other in 2012, as shown on the x-axis. Thus, there are 2 points with a value of 56 in the blue graph. </think>

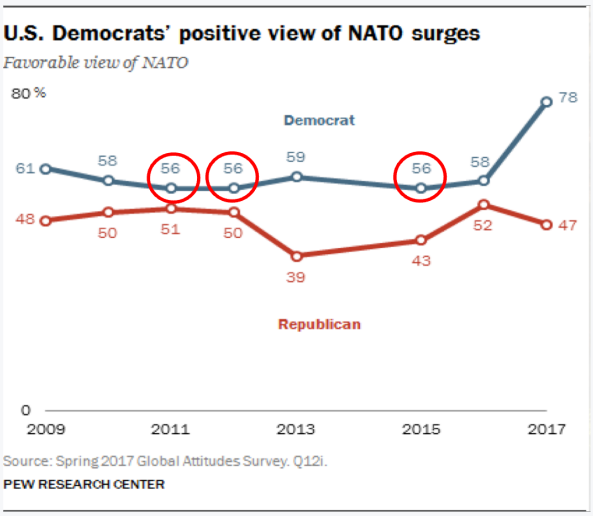
<answer> **2** </answer>

MIRROR:

<answer> There are 2 points with a value of 56 on the blue graph. </answer>

<think> I believe my answer is close, but not entirely correct. I need to carefully count all the points on the blue graph that have a value of 56. Look at each point on the graph, as indicated by the red circles in the image, and ensure I haven't missed any. </think>

```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'each
point on the graph with a value of 56', 'color': 'red', 'mark_type': 'circle'}}
</tool_call>
```



The red circles indicate each point on the graph with a value of 56.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer>...</answer> <think>... </think><tool_call>...</tool_call>

Continued on next page.

(b) **Part 2 of comparison case 1 with reasoning paradigms.**

Continuation (continued):

<answer> **There are 3 points with a value of 56 on the blue graph.** </answer>

<think> I have provided a targeted and practical response that meets the core needs of the user. </think>

```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': false, 'anchor':
'None', 'color': 'None', 'mark_type': 'None'}}
</tool_call>
```

(c) **Part 3 of comparison case 1 with reasoning paradigms.**

Figure 17: Comparison case 1 with reasoning paradigms.

Question: Which of the following is not directly connected to the subpages in the spider diagram?

Choices: A. Images B. Privacy policy C. Navigation links D. Contact info

GT: D



VL-Rethinker-7B:

To determine which option is not directly connected to the subpages in the spider diagram, let's analyze each option in the context of the diagram:

A. Images - The diagram shows that images are connected to the footer, not the subpages. B. Privacy policy - The diagram shows that the privacy policy is connected to the subpages. C. Navigation links - The diagram shows that navigation links are connected to both the header and the subpages. D. Contact info - The diagram shows that contact info is connected to the footer, not the subpages.

From this analysis, we can see that the options that are directly connected to the subpages are: - Privacy policy - Navigation links

The options that are not directly connected to the subpages are: - Images - Contact info

Since the question asks for the option that is not directly connected to the subpages, **the correct answer is: A**.

PixelReasoner-7B:

To determine which option is not directly connected to the subpages in the spider diagram, let's analyze the connections step by step:

1. **Images**: The diagram shows "Images" connected to the "Footer" section, not directly to the "Subpages". 2. **Privacy policy**: The diagram shows "Privacy policy" connected to the "Subpages" section. 3. **Navigation links**: The diagram shows "Navigation links" connected to the "Header", "Footer", and "Subpages" sections. 4. **Contact info**: The diagram shows "Contact info" connected to the "Footer" section, not directly to the "Subpages".

Given this analysis, "Images" and "Contact info" are not directly connected to the "Subpages" section. However, "Images" is more clearly not directly connected as it is specifically linked to the "Footer".

Therefore, the final answer is A.

Continued on next page.

(a) **Part 1 of comparison case 2 with reasoning paradigms.**

Continuation (continued):

DeepEyes-7B:

<think> In the spider diagram, the subpages section is directly connected to the following items: RSS options, search, newsletter box, privacy policy, contact info, images, and navigation links. The option that is not directly connected to the subpages is "PSI," which stands for "Privacy Info" in some contexts, but in this diagram, it is not listed. Therefore, the correct answer is "Privacy policy". </think>

<image_zoom_in_tool> Not applicable in this text-based environment. </image_zoom_in_tool>

<answer> **B. Privacy policy** </answer>

Adaptive-CoF-7B:

<think> > To determine which option is not directly connected to the subpages in the spider diagram, I need to analyze the connections between the subpages and the other elements. The diagram clearly shows that the subpages are directly connected to "Images," "Contact info," and "Navigation links." However, "Privacy policy" is not directly connected to the subpages. Instead, it is connected to the "Footer" section. Therefore, the correct answer is B. Privacy policy. </think>

<answer> **B. Privacy policy** </answer>

MIRROR:

<answer> B. Privacy policy. </answer>

<think> I believe my answer is incorrect. While I correctly identified that the spider diagram shows various elements of a website, I need to focus on the connections specifically to the 'Subpages' section. Look closely at the elements directly linked to 'Subpages' and see which one is not connected to it. </think>

```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor':
'elements directly linked to "Subpages", 'color': 'cyan', 'mark_type': 'line'}}
</tool_call>
```

Continued on next page.

(b) **Part 2 of comparison case 2 with reasoning paradigms.**

Continuation (continued):



The cyan lines indicate elements directly linked to 'Subpages'.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer>...</answer> <think>... </think><tool_call>...</tool_call>

<answer> **D. Contact info** </answer>

<think> I accurately identified the core activity, designed precise visual prompts, and achieved ideal effect after iterative reflection. </think>

```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': false, 'anchor':
'None', 'color': 'None', 'mark_type': 'None'}}
</tool_call>
```

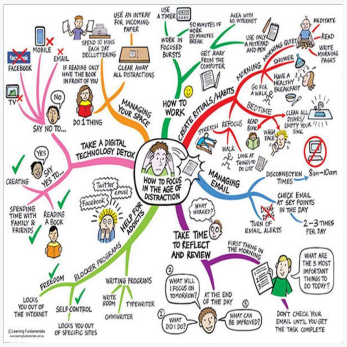
(c) **Part 3 of comparison case 2 with reasoning paradigms.**

Figure 18: Comparison case 2 with reasoning paradigms.

Question: Which tool is depicted as an option for helping to restrict internet access?

Choices: A. Self Control B. Freedom C. Blocker Programs D. Twitter

GT: B



VL-Rethinker-7B:

To determine the correct answer, let's analyze each option in the context of the image and the question:

A. Self Control - The image shows "SELF CONTROL" under the "BLOCKER PROGRAMS" section, which suggests it is a tool for restricting internet access. This seems like a plausible option.

B. Freedom - The image shows "FREEDOM" under the "BLOCKER PROGRAMS" section as well, indicating it is another tool for restricting internet access. This is also a strong candidate.

C. Blocker Programs - This is a general term that could refer to any tool that blocks access to the internet or specific sites. The image specifically lists "SELF CONTROL" and "FREEDOM" under this category, so it's a valid option, but it's more of a category rather than a specific tool.

D. Twitter - Twitter is a social media platform and not a tool for restricting internet access. It is listed in the image but is not related to the question.

Given this analysis, the options that are depicted as tools for helping to restrict internet access are "Self Control" and "Freedom."

Since the question asks for "which tool" is depicted, and both options A and B are correct, we need to choose one. Typically, the question would be designed to have a single correct answer, so we should choose the one that is more prominently or specifically listed under the "BLOCKER PROGRAMS" section.

The image clearly lists "SELF CONTROL" under the "BLOCKER PROGRAMS" section, making it the more specific and correct answer.

Final answer: A

PixelReasoner-7B:

To determine the correct answer, let's analyze the image step by step:

1. The image is a mind map titled "How to Focus in the Age of Distraction." 2. It includes various strategies and tools to help manage distractions and improve focus. 3. The section labeled "BLOCKER PROGRAMS" is connected to "LOCKS YOU OUT OF THE INTERNET" and "LOCKS YOU OUT OF SPECIFIC SITES." 4. This suggests that blocker programs are tools designed to restrict internet access.

Now, let's look at the options provided: A. Self Control - This is a strategy mentioned in the image but not specifically tied to internet access restriction. B. Freedom - This is also a strategy mentioned in the image but not specifically tied to internet access restriction. C. Blocker Programs - This matches the description in the image. D. Twitter - This is a social media platform and not a tool for restricting internet access.

Based on this analysis, the correct answer is: C.

Continued on next page.

(a) **Part 1 of comparison case 3 with reasoning paradigms.**

Continuation (continued):

DeepEyes-7B:

<think> The image is a mind map titled "How to Focus in the Age of Distraction." It includes various strategies to manage distractions, such as taking a digital technology detox, creating rituals/habits, managing your space, and handling email. The options provided are A. Self Control, B. Freedom, C. Blocker Programs, and D. Twitter. The image shows "FREEDOM" and "BLOCKER PROGRAMS" as tools to help with internet access restrictions. Since the question asks for the tool depicted as an option for helping to restrict internet access, the correct answer should be either B. Freedom or C. Blocker Programs. The image specifically highlights "BLOCKER PROGRAMS" as a tool for internet restriction. </think>

<image_zoom_in_tool> ... </image_zoom_in_tool>

<answer> **C. Blocker Programs** </answer>

Adaptive-CoF-7B:

<think> To determine which tool is depicted as an option for helping to restrict internet access, I need to analyze the image step by step. First, I observe the central theme of the image, which is about managing distractions and focusing in the age of technology. This suggests that the tools mentioned are related to digital technology. Next, I look for specific mentions of tools that can restrict internet access. The image shows two options: "FREEDOM" and "BLOCKER PROGRAMS." Both of these options are clearly labeled and connected to the theme of managing distractions. Since the question asks for the tool depicted as an option for restricting internet access, and both "FREEDOM" and "BLOCKER PROGRAMS" are relevant, I need to identify the one that is specifically mentioned as a tool for restricting internet access. The label "LOCKS YOU OUT OF THE INTERNET" is associated with "FREEDOM," indicating that it is the tool designed to restrict internet access. Therefore, the correct answer is C. </think>

<answer> **C. Blocker Programs** </answer>

MIRROR:

<answer> The correct answer is C. Blocker Programs. </answer>

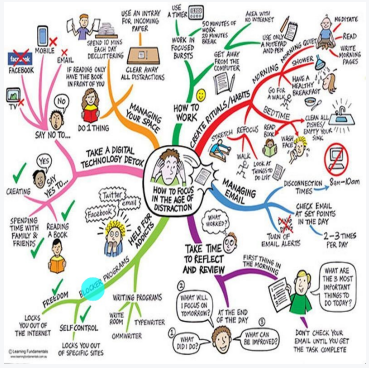
<think> I believe my answer is partially correct. I correctly identified that the image mentions tools to help restrict internet access. However, I need to be more specific about the tool. Look closely at the options provided and the labels in the image to identify the correct one. </think>

```
<tool_call>
{‘name’:‘visual_prompt_generator’, ‘arguments’: {‘flag’: true, ‘anchor’:
‘Blocker Programs’, ‘color’: ‘cyan’, ‘mark_type’: ‘point’}}
</tool_call>
```

Continued on next page.

(b) **Part 2 of comparison case 3 with reasoning paradigms.**

Continuation (continued):



The cyan point indicates Blocker Programs.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer>...</answer> <think>... </think><tool_call>...</tool_call>

<answer> **B. Freedom** </answer>

<think> I accurately identified the core activity, designed precise visual prompts, and matched visual prompts with visual skills, qualified answer. </think>

```
<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': false, 'anchor':
'None', 'color': 'None', 'mark_type': 'None'}}
</tool_call>
```

(c) **Part 3 of comparison case 3 with reasoning paradigms.**

Figure 19: Comparison case 3 with reasoning paradigms.

Question: What will the van do in the image? The choices are listed below:

Choices: A. stopping B. no response C. turn left D. turn right E. The image does not feature the van.

GT: A



VL-Rethinker-7B:

E.

PixelReasoner-7B:

To determine the van's future action, let's analyze the image step by step:

1. **Identify the van**: The van is located in the center of the image, on the right side of the road. 2. **Observe the van's position**: The van appears to be stationary, as there are no visible signs of motion blur or other vehicles overtaking it. 3. **Check for traffic signals or signs**: There are no visible traffic lights or signs that would indicate the van should stop or turn. 4. **Look at the surrounding vehicles**: The van is surrounded by other vehicles that are also stationary or moving in the same direction, suggesting no immediate need to stop or turn.

Given these observations, the van is likely to continue its current state of being stationary or moving forward in the same direction.

E

DeepEyes-7B:

<think> The image shows a van parked on the side of the road. There are no indications that the van is moving or about to move. The van is stationary, and there is no visible action that suggests it will perform any of the listed actions. Therefore, the best answer is that the van will not respond to any of the given options. </think>

<image_zoom_in_tool> Not needed for this question. </image_zoom_in_tool> <answer> **E** </answer>

Adaptive-CoF-7B:

<think> To determine what the van will do, I need to analyze its current status. However, the van is not explicitly shown or mentioned in the image, making it impossible to identify its behavior. Therefore, I cannot determine the van's actions based on the visual information. </think>

<answer> **B** </answer>

Continued on next page.

(a) **Part 1 of comparison case 4 with reasoning paradigms.**

Continuation (continued):

MIRROR:

<answer> A. </answer>

<think> I believe my answer is correct. The van is positioned near the edge of the road, and it appears to be in a position where it might stop. However, to be more confident, I should consider the context of the image and the van's position relative to the road and other vehicles. </think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'the van near the edge of the road, possibly stopping', 'color': 'red', 'mark_type': 'ellipse'}}
</tool_call>



The red ellipse indicates the van near the edge of the road, possibly stopping.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer>...</answer> <think>... </think><tool_call>...</tool_call>

<answer> **A.** </answer>

<think> I accurately identified the core activity, designed precise visual prompts, and achieved ideal effect after iterative reflection. </think>

(b) **Part 2 of comparison case 4 with reasoning paradigms.**

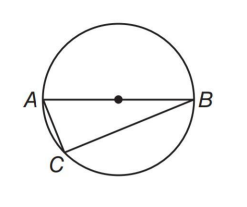
Figure 20: Comparison case 4 with reasoning paradigms.

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

Question: $\overline{AB}$ is a diameter,$AC = 8$ inches, and $BC = 15$ inches. Find the radius of the circle.

Choices: (A) 7.5 (B) 8 (C) 8.5 (D) 17

GT: (C) 8.5



MIRROR:

<answer>

Let's use the Pythagorean theorem to find the length of the diameter$\overline{AB}$. The diameter is the hypotenuse of the right triangle $\triangle ABC$.

So,

$$\overline{AB} = \sqrt{AC^2 + BC^2} = \sqrt{8^2 + 15^2} = \sqrt{64 + 225} = \sqrt{289} = 17$$

inches. The radius is half of the diameter, so the radius is $\frac{17}{2} = 8.5$ inches. The correct option is C.

</answer>

<think>

I believe my approach to using the Pythagorean theorem to find the diameter is correct, but there seems to be a mistake in my calculation. I need to double-check my arithmetic to ensure the diameter is calculated accurately. Then, I will calculate the radius by dividing the diameter by 2. I will use the red points in the image to verify my calculations.

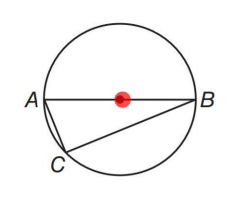
</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'diameter calculation and radius division', 'color': 'red', 'mark_type': 'point'}}
</tool_call>

Continued on next page.

(a) **Part 1 of Qualitative failure case 1.**

Continuation (continued):



The red point indicates diameter calculation and radius division.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer/>...</answer > <think >... </think ><tool_call >...</tool_call >

<answer>

Let's use the Pythagorean theorem to find the length of the diameter $\overline{AB}$. The diameter is the hypotenuse of the right triangle $\triangle ABC$.

So,

$$\overline{AB} = \sqrt{AC^2 + BC^2} = \sqrt{8^2 + 15^2} = \sqrt{64 + 225} = \sqrt{289} = 17$$

inches. The radius is half of the diameter, so the radius is $\frac{17}{2} = 8.5$ inches. **The correct option is C.**

</answer>

<think>

I accurately identified the core activity, designed precise visual prompts, and achieved ideal effect after iterative reflection.</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': false, 'anchor': 'None', 'color': 'None', 'mark_type': 'None'}}
</tool_call>

(b) **Part 2 of Qualitative failure case 1.**

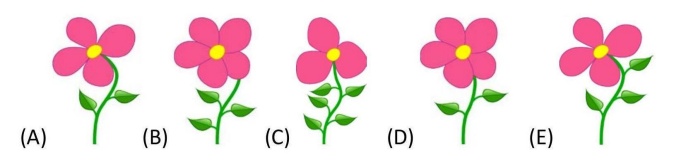
Figure 21: **Limited Applicability in Abstract Domains:** When addressing mathematical problems, the model attempts to verify symbolic logic (e.g., radius calculation) visually. However, since abstract derivations cannot be spatially grounded, the resulting visual prompts (red points) are uninformative and do not aid in verifying the calculation.

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

Question: The ladybird would like to sit on his flower. The flower has five petals and the stem has three leaves. On which flower should the ladybird sit?

Choices:(A) A (B) B (C) C (D) D (E) E

GT: (B) B



MIRROR:

<answer>E</answer>

<think>

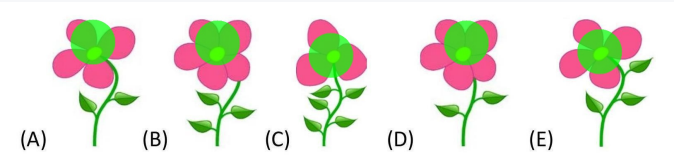
I believe my answer is correct. The flower with five petals and three leaves is indeed option E. However, I should also mention that the ladybird would like to sit on this flower, as indicated in the question.

</think>

<tool_call>
{'name':'visual_prompt_generator', 'arguments': {'flag': true, 'anchor': 'flower with five petals and three leaves', 'color': 'green', 'mark_type': 'point'}}
</tool_call>

The green points indicate flower with five petals and three leaves.

Remember: Output your Answer first, then Think, and finally Tool Call. Strict format: <answer/>...</answer > <think >... </think ><tool_call >...</tool_call >



Answer

<answer>**E**</answer>

<think>

I accurately identified the core activity, designed precise visual prompts, and achieved ideal effect after iterative reflection.

</think>

Figure 22: **Coarse-grained Attribute Binding:** This case demonstrates the limitations of the current visual verification mechanism.

Table 14: Failure type distribution over the 655 failure cases from the incorrect grounding and incorrect reasoning stages.

Failure Type	**Percentage**
Attribute binding	42.1%
Abstract reasoning	19.7%
Others	38.2%