CAVE: A Structured Credit Assignment Approach for Fragmented Visual Evidence Reasoning

Tengda Guo[1]* **Jie Leng**[1]* **Hanlei Li**[1]* **Yaoyuan Liang**[1]† **Qingyue Zhang**[1] **Dian Yang**[1]
Mingyu Zhang[2] Yuhua Fu[3] Shao-Lun Huang[1]†
[1]Tsinghua University [2]Peking University [3]Zhejiang University of Technology
{gtd25, leng-j20, lihl25}@mails.tsinghua.edu.cn

Abstract

Vision-Language Models (VLMs) have achieved strong performance on general multimodal reasoning, yet remain challenged in integrating nonlocal visual information to support semantically underdetermined visual reasoning. We describe this challenge as Fragmented Visual Reasoning. To this end, we propose Credit Assignment for Visual Evidence (CAVE), a structured process-reward method based on GRPO for interleaved visual reasoning. Specifically, CAVE evaluates the contribution of intermediate steps at the action level via three complementary reasoning process signals: belief update, evidence acquisition, and adaptive focus control, thereby guiding the model to optimize each reasoning action and learn more reliable visual reasoning strategies. Meanwhile, we construct TRACER-Bench, which covers four nonlocal and semantically confusable reasoning dimensions and provides key intermediate evidence to supervise reasoning paths. Experiments demonstrate that CAVE substantially improves performance on tasks requiring fragmented visual evidence integration, covering both public benchmarks and our newly introduced TRACER-Bench, while retaining competitive performance on general multimodal evaluations. Further analyses reveal that CAVE effectively improves the visual reasoning capacity and exhibits stronger robustness under longer-range and deeper cross-region dependencies.

1 Introduction

Vision-Language Models (VLMs) have made substantial progress in multimodal reasoning tasks [1–4], largely benefiting from the semantic priors accumulated through large-scale pretraining and the chain-of-thought capabilities of language models [5, 6]. However, when tasks primarily rely on basic visual capabilities with low language priors, existing VLMs still exhibit clear brittleness. Representative tasks in Tunnel Vision [7, 8] provide a concrete example: even simple human-solvable problems can require chaining evidence across multiple, potentially distant image regions, where current VLMs perform poorly. Building on this observation, we study a broader setting in which task-critical evidence is spatially fragmented and semantically weakly separable across regions. We refer to this challenge as Fragmented Visual Reasoning (FVR). As illustrated in Figure 1(a), FVR creates a reasoning dilemma: if VLMs rely only on coarse global visual perception, their vision-side representations often fail to robustly capture long-range dependencies and subtle local differences, thereby biasing reasoning paths toward a verbalized subset of visual evidence. As a result, models may produce coherent but visually ungrounded reasoning chains, leading to error propagation in subsequent reasoning steps.

*Equal contribution.
†Corresponding authors.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

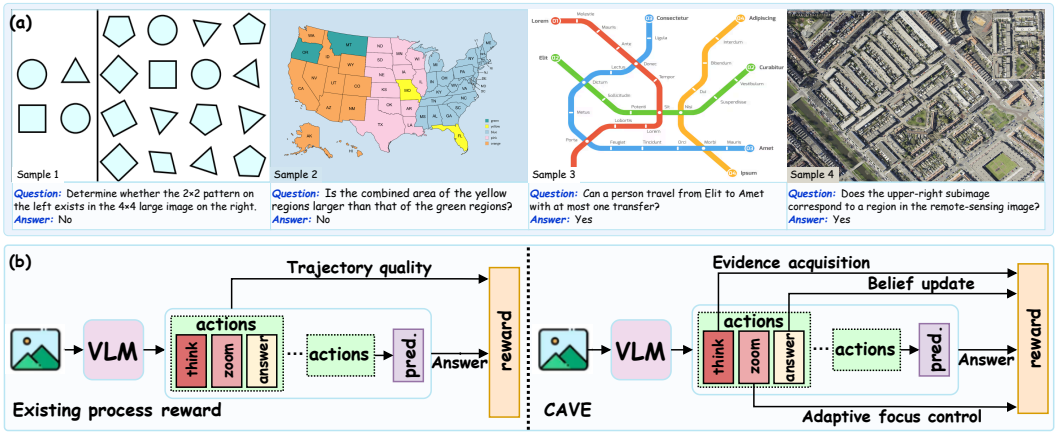


Figure 1: (a) Samples of Fragmented Visual Reasoning. The examples cover Geometry, Geography, Route, and Remote-Sensing tasks. Samples 1–3 require organizing confusable and subtle evidence across separate regions, while Sample 4 requires matching nonlocal relations in a noisier real image. (b) Comparison between existing process-reward methods and CAVE. Existing methods usually rely on trajectory-level or answer-side optimization, whereas CAVE assigns action-level structured credits according to the reasoning contributions of intermediate visual-evidence reasoning steps.

This challenge naturally connects to recent interleaved multimodal reasoning methods, which introduce a "Think–with–Image" paradigm, enabling models to progressively acquire visual information through localized re-observation and providing a more natural solution form for such tasks [9–12]. However, even though existing process-reward methods attempt to alleviate sparse feedback in long-trajectory training [13, 14], they usually evaluate intermediate steps from the perspective of final-answer improvement or overall trajectory quality, failing to provide action-specific process supervision and thus leaving models without targeted guidance to learn reliable strategies for each distinct reasoning behavior in visual evidence exploration, as shown in Figure 1(b) left.

To address this, we propose Credit Assignment for Visual Evidence (CAVE), which performs action-level structured credit assignment over multi-round reasoning trajectories for visual evidence exploration [15], as shown in Figure 1(b) right. CAVE evaluates intermediate reasoning with three complementary signals: belief update, evidence acquisition, and adaptive focus control. These signals respectively assess progress toward the correct answer, recovery of task-critical visual details, and the appropriateness of visual region focusing, thereby explicitly guiding the model to learn more reliable evidence discovery, reasoning state updating, and goal-directed exploration behaviors. To enable controlled evaluation and process-level analysis, we construct TRACER-Bench, which covers three abstract scenarios and one real-world scenario: Rule-Switching Navigation, Nonsemantic Tracing, Embedded Matching, and Remote-Sensing Subimage Matching. These scenarios require cross-regional evidence with low semantic separability, and TRACER-Bench provides corresponding key evidence for offline analysis.

In summary, our main contributions are as follows: 1) We introduce Fragmented Visual Reasoning, a challenging task characterized by low semantic separability across multiple visual regions, together with TRACER-Bench for systematic evaluation across diverse scenarios with verifiable explicit visual-evidence chains. 2) We propose CAVE, a structured credit assignment process-reward method based on GRPO for interleaved visual reasoning. 3) Experiments show that CAVE effectively improves model performance on benchmarks with fragmented vision, achieves competitive results on general multimodal evaluations, and effectively improves the visual reasoning capacity.

2 Related work

VLMs visual reasoning and interleaved visual reasoning. VLMs have achieved significant progress on general multimodal reasoning tasks. Representative works such as Flamingo and LLaVA have advanced the development of general-purpose VLMs through interleaved image-text modeling and related techniques [1–4, 16, 17]. However, existing VLMs remain fragile in tasks requiring detailed discrimination and nonlocal evidence. Tunnel Vision, BabyVision, and related benchmarks reveal this issue from the perspectives of long-range region association and basic visual cognition, respectively [7, 8, 18–20]. To improve reasoning capability, methods such as DDCoT and LLaVA-

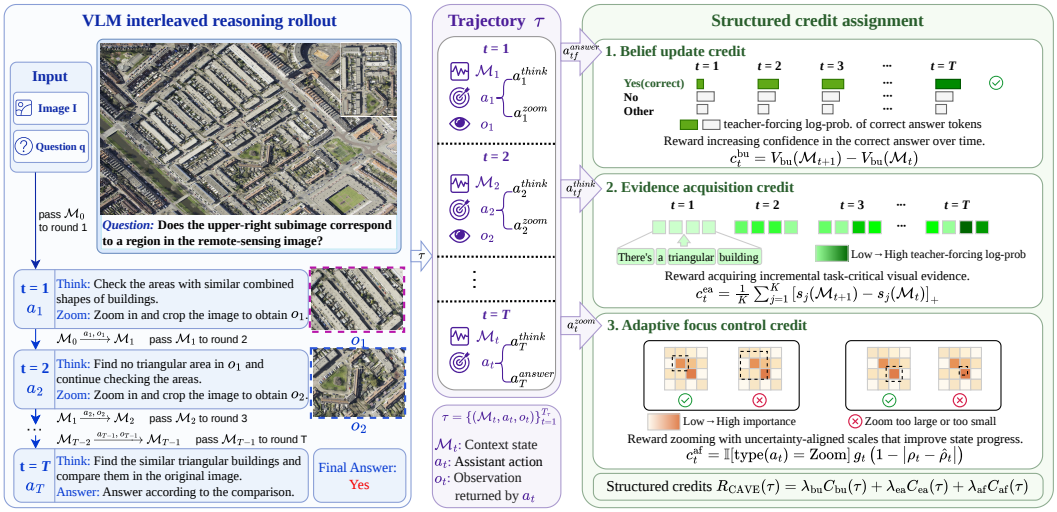


Figure 2: Overview of CAVE. Given an input pair, the policy performs fragmented visual evidence reasoning. CAVE assigns structured process credits to belief update, evidence acquisition, and adaptive focus control, and is optimized with GRPO. a_{tf} denotes the teacher-forced action under the current context state, obtained outside the rollout and used only for reward computation.

CoT introduce staged or visually grounded intermediate reasoning [21, 22, 6, 23, 24]. Furthermore, Visual Sketchpad, VisProg, ViperGPT, DeepEyes, and related works extend VLMs reasoning into a multi-round "reasoning–observing–reasoning" process through tool use or adaptive zooming [9, 25, 26, 10–12]. However, these methods mainly focus on intermediate reasoning or local observation mechanisms, while still lacking dedicated modeling of how observations serve as accumulated visual states for subsequent cross-region dependency progression.

GRPO and process rewards. GRPO/R1-style reinforcement learning shows that rule-based or final-answer rewards can elicit long-chain reasoning capabilities in large models [15, 27], and has been extended to Vision-Language Models by works such as VLM-R1, R1-VL, and OpenVLThinker [28–30]. However, existing GRPO-style VLMs training usually relies on final answers, format constraints, or general rule-based rewards, making it difficult to characterize the process-level contributions of observation, evidence acquisition, and region transition in multi-round visual reasoning. To alleviate reward sparsity in long-trajectory training, VisualPRM, StepSearch, GiGPO, and IGPO explore process rewards, step-level feedback, trajectory comparison, and information-gain signals [13, 31, 32, 14]. However, such process signals usually focus more on the overall contribution of states to final task utility or answer formation, and do not explicitly distinguish different types of evidence progression in multi-round visual reasoning. In contrast, CAVE targets visual evidence progression and structurally decomposes intermediate contributions into three types of process credit: belief update, evidence acquisition, and adaptive focus control.

3 Method

The goal of CAVE is to optimize the model's capability of fragmented visual evidence reasoning during interleaved visual reasoning. Different from measuring intermediate states with a single information gain or final task utility, CAVE focuses on whether intermediate actions promote answer-belief update, key evidence acquisition, and adaptive focus control. To this end, we first define the model's interleaved reasoning trajectory, then introduce three types of structured process credits, and describe how they are integrated into GRPO optimization [15], with details shown in Figure 2.

3.1 Interleaved visual reasoning trajectory

Given an input $x = (I, q)$, where I denotes the image and q denotes the question, CAVE represents the model's solving process as an interleaved visual reasoning trajectory with at most T rounds:

$$\tau = \{(\mathcal{M}_t, a_t, o_t)\}_{t=1}^{T_\tau}, \quad a_t \sim \pi_\theta(\cdot \mid \mathcal{M}_t), \quad \mathcal{M}_{t+1} = \text{Update}(\mathcal{M}_t, a_t, o_t), \quad T_\tau \le T.$$

Here, $\mathcal{M}_t$ denotes the multi-modal context state at the beginning of round t, which contains the original image, the question, historical reasoning texts, and the acquired local visual observations; a_t denotes the assistant action generated by the policy model under the current state, which may

correspond to textual reasoning, a zoom-tool call, or the final answer generation; and o_t denotes the observation returned by the environment according to a_t, such as the local sub-image after zooming and its additional prompt. This trajectory formulation identifies the components of interleaved reasoning and makes explicit the boundary between policy-generated actions and environment-provided observations. We specify the assistant-token optimization and the associated response-mask setting in the policy optimization section.

3.2 Structured credits for fragmented visual evidence reasoning

During reasoning, the value of an effective intermediate step does not always immediately manifest as an increase in answer confidence. Intermediate steps can serve distinct functional roles: acquiring precursor visual evidence, updating intermediate states, or determining the location and scale of subsequent observation. Based on this, CAVE does not adopt a single state-utility gain, but attributes the state transition $\mathcal{M}_t \rightarrow \mathcal{M}_{t+1}$ induced by action a_t from three complementary perspectives: answer belief improvement, visual evidence observation, and region focus matching. These credits constitute process-level constraints on fragmented visual evidence reasoning, rather than collapsing intermediate steps into a single undifferentiated state-utility reward signal.

3.2.1 Belief update credit

Belief update credit characterizes answer-side reasoning progress. If a reasoning step or tool call effectively updates the context state, then under the updated $\mathcal{M}_{t+1}$, the ground-truth answer should have higher conditional log-likelihood. Let $\mathcal{A}$ denote the token interval of the answer body. We use teacher forcing to define the answer-side state value V_{bu} and its adjacent-state gain c_t^{bu} [14]:

$$V_{\mathrm{bu}}(\mathcal{M}_t) = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \log p_\theta \left(y_i^\star \mid \mathcal{M}_t, y_{<i}^\star \right), \qquad c_t^{\mathrm{bu}} = V_{\mathrm{bu}}(\mathcal{M}_{t+1}) - V_{\mathrm{bu}}(\mathcal{M}_t).$$

We only score the answer body to avoid the influence of format templates or irrelevant prefixes and suffixes on the estimation. This credit term rewards intermediate steps that move the model state closer to the correct answer, thereby constraining answer-oriented state updates.

3.2.2 Evidence acquisition credit

Relying solely on answer-belief gain is insufficient for multi-round visual reasoning, since an intermediate action may make critical visual clues available before these clues are fully integrated into an answer-side belief. To capture such precursor progress that may be missed by answer-belief gain, we further introduce evidence acquisition credit. Teacher forcing details are in Appendix E.

For training samples equipped with task-critical evidence annotations, we represent the intermediate visual information required for answer formation as a set of K textualized visual evidence units, $\mathcal{P} = \{p_1, \ldots, p_K\}$, where each p_i denotes an individual evidence unit, as illustrated by the example "There's a triangular building" in Figure 2. These units correspond to necessary visual intermediate information, such as perceptual details, local attributes, cross-region correspondences, path continuity, or structural cues. They are automatically constructed and further verified through human inspection, with detailed construction and quality-control procedures provided in Appendix B.3.

For the j-th evidence unit p_j, let its token interval be $\mathcal{E}_j$. We define its recoverability score s_j under state $\mathcal{M}_t$ and its positive gain as:

$$s_j(\mathcal{M}_t) = \frac{1}{|\mathcal{E}_j|} \sum_{i \in \mathcal{E}_j} \log p_\theta \left(p_{j,i} \mid \mathcal{M}_t,\, p_{j,<i} \right), \qquad c_t^{\mathrm{ea}} = \frac{1}{K} \sum_{j=1}^{K} \left[s_j(\mathcal{M}_{t+1}) - s_j(\mathcal{M}_t) \right]_+ .$$

Here, $[\cdot]_+$ indicates that only positive evidence gains are retained. This term does not require the model to generate standardized explanations during reasoning, but instead transforms whether the current state is closer to the task-critical visual evidence into an optimizable process signal, thereby distinguishing trajectories that blindly guess from those that progressively acquire evidence.

3.2.3 Adaptive focus control credit

Adaptive focus control credit constrains the zoom-based visual exploration in interleaved reasoning. An effective zoom should balance evidence coverage and visual noise, providing informative local perception while preserving necessary contextual cues. Rather than merely rewarding the occurrence of a zoom call [11, 12], CAVE encourages the model to select local observations that both induce state progress and match the current uncertainty.

Specifically, we estimate the uncertainty of the current state using the top-k distribution entropy of answer tokens during answer teacher forcing, denoted as $H_t = \frac{1}{|\mathcal{A}|} \sum_{i \in \mathcal{A}} \mathrm{Entropy}_k(i)$, and

Algorithm 1 Structured credit computation in CAVE

Require: $\tau = \{(\mathcal{M}_t, a_t, o_t)\}_{t=1}^{T_\tau}$, answer span $\mathcal{A}$, evidence units $\mathcal{P}$
Ensure: $R_{\mathrm{CAVE}}(\tau)$
for $t = 1$ to T_τ **do**
  $c_t^{\mathrm{bu}} \leftarrow V_{\mathrm{bu}}(\mathcal{M}_{t+1}) - V_{\mathrm{bu}}(\mathcal{M}_t)$
  $c_t^{\mathrm{ea}} \leftarrow \frac{1}{K}\sum_{j=1}^{K}[s_j(\mathcal{M}_{t+1}) - s_j(\mathcal{M}_t)]_+$
  $\hat{\rho}_t \leftarrow \rho_{\min} + (\rho_{\max} - \rho_{\min})\mathrm{NormEntropy}_k(\mathcal{A} \mid \mathcal{M}_t)$
  $g_t \leftarrow 1 - \exp\big(-\big([c_t^{\mathrm{bu}}]_+ + c_t^{\mathrm{ea}}\big)\big)$
  $c_t^{\mathrm{af}} \leftarrow \mathbb{I}[a_t = \mathrm{Zoom}]\, g_t\, (1 - |\rho_t - \hat{\rho}_t|)$
end for
$C_m(\tau) \leftarrow \sum_{t=1}^{T_\tau} w_t^m \phi_m(c_t^m), \quad m \in \{\mathrm{bu}, \mathrm{ea}, \mathrm{af}\}$
return $R_{\mathrm{CAVE}}(\tau) = \lambda_{\mathrm{bu}} C_{\mathrm{bu}} + \lambda_{\mathrm{ea}} C_{\mathrm{ea}} + \lambda_{\mathrm{af}} C_{\mathrm{af}}$

normalize it to obtain $u_t \in [0, 1]$. For a valid zoom at round t, let the cropped-region area ratio be ρ_t, and define its uncertainty-aware target scale as $\hat{\rho}_t = \rho_{\min} + (\rho_{\max} - \rho_{\min})u_t$.

To avoid rewarding zoom actions with appropriate scale but no actual contribution, we further gate the reward using positive state progress after observation:

$$g_t = 1 - \exp\left(-\left([c_t^{\mathrm{bu}}]_+ + c_t^{\mathrm{ea}}\right)\right), \qquad c_t^{\mathrm{af}} = \mathbb{I}[\mathrm{type}(a_t) = \mathrm{Zoom}]\, g_t\, (1 - |\rho_t - \hat{\rho}_t|)\,.$$

Here, c_t^{bu} and c_t^{ea} denote belief update and evidence acquisition credit, respectively. This term does not rely on region-level ground-truth annotations; instead, it rewards a zoom action only if it yields positive belief-update or evidence-acquisition progress, while keeping its observation scale aligned with the current uncertainty. Therefore, it provides an adaptive zoom-scale constraint for acquiring fragmented visual evidence.

3.3 Policy optimization with structured credits

As shown in Algorithm 1, for each type of process credit, we first aggregate step-level credits into trajectory-level credits. Let $m \in \{\mathrm{bu}, \mathrm{ea}, \mathrm{af}\}$, and let c_t^m denote the local credit at step t. The CAVE process reward is defined as

$$C_m(\tau) = \sum_{t=1}^{T_\tau} w_t^m\, \phi_m(c_t^m), \qquad R_{\mathrm{CAVE}}(\tau) = \lambda_{\mathrm{bu}} C_{\mathrm{bu}}(\tau) + \lambda_{\mathrm{ea}} C_{\mathrm{ea}}(\tau) + \lambda_{\mathrm{af}} C_{\mathrm{af}}(\tau),$$

where w_t^m denotes a round-dependent weight used to encourage earlier evidence progression, and $\phi_m(\cdot)$ denotes a bounded transformation that clips extreme local credits to stabilize GRPO training. Given an input $x = (I, q)$, we sample a group of rollouts $\{\tau_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\mathrm{old}}}$, and compute the group-relative advantage based on R_{CAVE}:

$$\hat{A}_i = \frac{R_{\mathrm{CAVE}}(\tau_i) - \mathrm{mean}_{j=1}^{G} R_{\mathrm{CAVE}}(\tau_j)}{\mathrm{std}_{j=1}^{G} R_{\mathrm{CAVE}}(\tau_j) + \delta}.$$

We then update the policy using the GRPO objective [15]. Optimization is applied only to the assistant-generated tokens; the observations returned by tools, system prompts, and environment-injected information serve only as contextual conditions, with their response masks set to 0, and do not participate in policy-gradient computation. In practice, we add lightweight task-anchoring terms, such as answer correctness, format validity, and round-number constraints, to maintain training stability. In this way, CAVE extends sparse supervision centered on terminal correctness into an optimization signal for visual evidence progression under group-relative comparison.

4 TRACER-Bench

Scenario coverage and construction. Existing benchmarks with nonlocal visual nature such as Tunnel Vision and BabyVision provide important references for studying the basic visual capabilities of VLMs [7, 8]. However, these benchmarks are difficult to simultaneously satisfy the requirements of scenario coverage, difficulty control, and process-evidence analysis. To this end, we construct TRACER-Bench (Text-Resistant Cross-Region Evidence Reasoning), a compact and controllable fragmented visual reasoning benchmark containing 980 VQA samples. It covers common cross-region dependency patterns with spatially fragmented and semantically weakly separable evidence, including three abstract scenarios and one real-world scenario: Rule-Switching Navigation (VJump), Nonsemantic Tracing (LT), Embedded Matching (Match), and Remote-Sensing Subimage Matching (MME-RS), corresponding to new dimensions of rule-switching dependency chains, non-semantic path tracing, local–global structural matching, and real-world remote-sensing subimage matching.

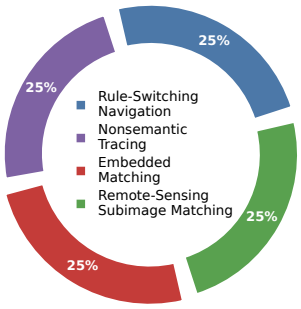


Figure 3: Scenario distribution of TRACER-Bench.

Table 1: Comparison between TRACER-Bench and related vision-centric multimodal benchmarks.

Benchmark	Cross-region dependency	Difficulty control	Explicit process evidence
Tunnel Vision [7]	✓	✓	×
BabyVision [8]	Partial	Partial	×
BLINK [19]	Partial	Partial	×
HallusionBench [20]	Partial	×	×
RealWorldQA [33]	Partial	×	×
MMStar [34]	Partial	Partial	×
TRACER-Bench	✓	✓	✓

In addition, TRACER-Bench annotates task-critical visual evidence in textual form, supporting in-depth offline process analysis of the reasoning process, while not being provided as model input. Scenario distribution and benchmark comparisons are shown in Figure 3 and Table 1. Scenario design, dataset statistics, quality control are provided in Appendix B.

Training corpus and data isolation. We further construct a training corpus for CAVE reinforcement learning and use `perception` to compute the evidence acquisition credit. The training data shares the same four scenario families with TRACER-Bench, but is completely isolated from the test set in terms of random seeds, layouts, image regions, and concrete instances, in order to avoid instance leakage and evaluate generalization to unseen layouts and dependency structures. To improve structural coverage, we additionally construct non-overlapping Tunnel-Vision-style synthetic data [7], and supplement the `perception` field to unify it into the `prompt-answer-perception` format. This data is used only as an auxiliary training corpus, is not part of TRACER-Bench, and does not overlap with the public Tunnel Vision evaluation set. Detailed construction and split protocols are provided in Appendix C.

5 Experiments

5.1 Experimental setup

Baselines. We compare six categories of methods: Human and Random as the reference upper bound and chance-level reference, respectively; GPT-5.2 and Gemini 3.1 Pro as strong closed-source model references [35, 36]; Qwen2.5-VL-7B, Qwen3-VL-8B, InternVL3.5-8B, and MiniCPM-V 4.5 as open-source general-purpose VLMs [4, 37–39]; R1-OneVision-7B, ThinkLite-VL, and VLAA-Thinker as recent reasoning-oriented VLMs [40–42]; DeepEyes as a tool-assisted look-again method [11] and PixelReasoner-RL-v1 as a visual-operation reinforcement learning baseline [43];and Belief-Only as a same-backbone process-reward model for examining whether single answer-side information gain is sufficient. CAVE-3B and CAVE-7B are used to evaluate parameter efficiency and the full method performance, respectively.

Benchmarks. We evaluate model capabilities at two levels: fragmented visual evidence reasoning and general multimodal benchmarks. The former includes Tunnel Vision, BabyVision, and our constructed TRACER-Bench, containing 1,350, 388, and 980 samples, respectively [7, 8]. TRACER-Bench consists of TRACER-Core, which contains three abstract and controllable scenarios, and MME-RS, which targets real-world remote-sensing scenarios. The latter includes MMStar, RealWorldQA, HallusionBench-Image, BLINK, MathVista-mini, and MathVerse-mini, which are used to evaluate out-of-domain transfer and the preservation of general capabilities [34, 33, 20, 19, 44, 45]. Unless otherwise specified, all methods adopt the same input format, answer extraction, and evaluation protocol.

Training details. CAVE uses Qwen2.5-VL as the base model and follows a two-stage training procedure on 8 A800 GPUs: cold-start training on 10k DeepEyes-style samples, followed by process-reward training on 5,000 TRACER-style samples with process evidence annotations [4, 11, 15]. Detailed training and reward configurations are provided in Appendix D.1.

5.2 Main results

We first compare CAVE with different types of baselines on reasoning tasks with fragmented vision, with results shown in Table 2. CAVE achieves the best overall average performance among comparable open-source baselines and significantly outperforms Qwen2.5-VL-7B with the same backbone [4], indicating that its gains mainly come from structured optimization of the intermedi-

Table 2: Main results on fragmented vision reasoning benchmarks. Results are grouped by Tunnel Vision, BabyVision, and TRACER-Bench. Models marked with * are Instruct versions.

Model	Param	Tunnel Vision			BabyVision				TRACER-Bench				Overall Avg.
		ObjReID	VS	Circuit	FD	SP	VPR	VT	LT	Match	VJump	MME-RS	
Human	–	100	100	99.0	94.1	94.1	94.1	94.1	100	97.6	95.2	99.0	98.3
Random	–	50.0	12.5	4.00	10.3	10.3	10.3	10.3	16.2	50.0	12.1	50.0	24.1
GPT-5.2	–	71.8	68.4	52.9	27.3	35.2	54.9	34.9	58.4	68.2	52.2	72.7	61.8
Gemini 3.1 Pro	–	67.3	72.0	46.4	46.2	53.7	53.9	43.4	53.5	64.5	44.1	67.8	56.4
Qwen2.5-VL*	7B	44.4	11.1	17.6	10.4	13.1	13.7	7.79	22.9	53.5	15.9	43.7	25.9
Qwen3-VL*	8B	48.2	13.6	22.4	13.5	14.3	11.7	9.09	**35.9**	55.1	15.1	52.7	30.1
InternVL3.5*	8B	45.1	15.3	20.4	14.1	**21.9**	19.6	10.4	26.1	41.2	**23.7**	58.4	29.2
MiniCPM-V 4.5	8B	50.0	11.8	17.8	12.3	19.8	19.6	9.09	30.2	52.7	16.7	55.1	29.2
R1-OneVision	7B	52.6	8.00	16.7	9.20	16.5	23.5	11.7	22.0	43.7	7.80	51.8	26.0
VLAA-Thinker	7B	46.9	10.9	22.2	12.3	20.9	19.6	11.7	22.0	43.7	14.3	47.8	27.0
ThinkLite-VL	7B	46.9	10.2	22.7	11.0	17.6	21.6	11.7	22.0	45.3	17.6	50.2	27.4
DeepEyes	7B	49.3	13.3	19.1	11.5	18.7	22.1	10.9	23.7	49.4	15.9	54.3	28.5
PixelReasoner-RL-v1	8B	52.0	14.7	22.0	14.1	19.8	23.5	11.7	29.0	57.1	19.6	56.3	31.6
CAVE Belief-Only	7B	49.7	12.2	20.9	12.9	18.7	15.7	9.10	28.8	52.8	16.8	55.2	29.6
CAVE-3B	3B	50.2	12.9	21.1	13.5	17.6	23.5	10.4	27.2	56.0	18.4	57.6	30.5
CAVE-7B	7B	**53.5**	**16.7**	**23.8**	**15.3**	18.7	**25.5**	**14.3**	31.4	**58.4**	20.8	**59.2**	**33.3**
Δ (vs Qwen2.5-VL)	7B	+9.1	+5.6	+6.2	+4.9	+5.6	+11.8	+6.5	+8.5	+4.9	+4.9	+15.5	+7.4

ate visual evidence progression process, rather than a stronger pretrained model. Compared with the tool-assisted DeepEyes [11] and reasoning-oriented VLMs such as ThinkLite-VL and VLAA-Thinker [41, 42], CAVE achieves stronger overall performance, suggesting that neither tool-based re-observation nor generic reasoning-oriented training is sufficient for robust evidence acquisition, state maintenance, and cross-region integration.We additionally compare CAVE with PixelReasoner-RL-v1 [43], a closely related method that encourages pixel-space exploration through curiosity-driven reinforcement learning. CAVE-7B achieves higher overall accuracy (33.3 vs. 31.6), suggesting that structured process-credit optimization provides competitive benefits beyond encouraging visual-operation exploration alone.

Furthermore, Belief-Only outperforms most base and tool-augmented models [37, 38, 40, 11, 41, 42], but remains clearly weaker than the full CAVE, indicating that belief gain is an effective process signal, yet is insufficient by itself to cover precursor evidence acquisition and visual-action focusing. CAVE-3B exceeds multiple 7B/8B open-source baselines in overall average performance [37, 38], showing that structured process-credit training can still induce effective visual evidence progression behaviors with a smaller-parameter backbone. Together, these results demonstrate that the advantage of CAVE does not come from a single information-gain reward, model scale, or a specific tool format, but from structured modeling of the visual evidence progression process.

We further examine the out-of-domain transfer and capability preservation of CAVE on general multimodal evaluations, with results shown in Table 3 [34, 33, 20, 19, 44, 45]. CAVE achieves the best results on four of the six general benchmarks and outperforms PixelReasoner-RL-v1 on five benchmarks, with a higher average score. These results indicate that process optimization centered on visual evidence exploration preserves competitive general multimodal capabilities while providing positive transfer across several vision-language reasoning scenarios.

To assess backbone generalizability, we retrain CAVE on InternVL3.5-8B while keeping the training data, reward design, rollout budget, and evaluation protocol unchanged. As shown in Appendix D.4, CAVE consistently outperforms both direct inference and correctness-only optimization across fragmented-vision tasks. These findings suggest that structured process credits provide transferable optimization signals beyond terminal-answer supervision, with benefits that are not specific to the Qwen2.5-VL backbone.

In summary, CAVE not only brings stable same-backbone improvements on fragmented vision reasoning tasks, but also demonstrates strong transfer capability on general multimodal evaluations, indicating that what it optimizes is not a specific benchmark or data template, but a more transferable capability of visual evidence exploration.

Table 3: Comparison of model performance on general multimodal benchmarks. HallusionBench-Image (avg.) reports the average score over the official HallusionBench metrics.

Model	Generalization Performance					
	MMStar	RealWorldQA	Hallu.-Img (avg.)	BLINK	MathVista (mini)	MathVerse (mini)
Qwen2.5-VL (7B)	61.5	59.4	60.3	58.5	66.7	48.1
Qwen3-VL-8B-Instruct	64.3	<u>69.3</u>	62.5	**63.3**	<u>70.6</u>	**60.9**
InternVL3.5-8B	<u>64.5</u>	61.6	57.9	59.8	69.6	53.2
DeepEyes-7B	63.6	68.4	<u>63.1</u>	56.4	70.3	49.2
PixelReasoner-RL-v1	64.0	67.2	62.7	<u>61.8</u>	69.5	52.3
CAVE-7B	**65.2**	**69.7**	**64.1**	61.5	**71.2**	<u>55.7</u>
Δ (vs. Qwen2.5-VL-7B)	+3.7	+10.3	+3.8	+3.0	+4.5	+7.6

Table 4: Ablation studies on structured credits and training sources, evaluating credit complementarity and the effect of data composition on fragmented vision reasoning.

(a) Structured credits ablation.

Method	Tunnel Vision	Baby Vision	TRACER -Core	MME -RS
Cold-Start	25.2	11.6	30.3	46.1
Correct Only	26.7	11.3	31.2	49.4
w/o Belief	28.9	15.6	34.1	56.7
w/o Evidence	29.6	15.3	34.7	55.9
w/o Focusing	29.3	16.5	35.2	57.6
Full CAVE	31.3	17.3	36.9	59.2

(b) Training-source ablation.

Training Data	Tunnel Vision	Baby Vision	TRACER -Core	MME -RS
Cold-start	25.2	11.6	30.3	46.1
DeepEyes data	29.3	15.1	33.9	53.5
w/o TV	29.9	16.6	34.0	51.8
w/o RS	30.6	16.4	35.5	56.7
w/o TRACER-Core	30.3	15.7	34.8	57.6
Full Source	31.3	17.3	36.9	59.2

5.3 Ablation studies

We conduct ablation analyses from two perspectives: structured credits design and training-source composition. All variants use the same base model, interleaved reasoning structure, and training configuration, changing only the process credit terms or the data sources used in the second-stage training.

Structured credits ablation. Table 4a evaluates the contribution of structured process credits. Cold-start learns the basic multi-round tool-use format and does not undergo CAVE reinforcement learning; Correct Only uses the final-answer correctness reward; and the remaining variants remove belief update, evidence acquisition, or adaptive focus control, respectively. The results show that both Cold-start and Correct Only are clearly weaker than the full CAVE, indicating that the gains do not merely come from interleaved tool use or terminal supervision. Removing any process credit also leads to performance degradation, demonstrating the complementarity of the three signals. Among them, w/o Belief usually causes the most pronounced degradation, suggesting that answer-belief update is the most stable progress signal; evidence acquisition and focusing also bring consistent gains. Overall, the full CAVE achieves the best or most balanced performance, verifying the necessity of structured credit assignment for visual evidence exploration.

Training-source ablation. Table 4b analyzes the role of training sources. We compare training configurations that use DeepEyes general-reasoning data supplemented with perception fields, as well as those that remove TV (Tunnel Vision), TRACER-Core, or MME-RS, respectively. Even when a certain source is removed, the model still improves over cold-start on the corresponding test domain, alleviating the concern that CAVE merely memorizes a specific generator or data-source template. Meanwhile, the full mixed-data setting achieves the best or most balanced performance, indicating that different sources provide complementary coverage: Tunnel-Vision-style data provides non-local reasoning forms [7], TRACER-Core covers abstract and controllable dependencies, and Remote-Sensing introduces scale, rotation, and complex background variations in real images. Together, the two groups of ablations show that the core gains of CAVE come from structured process credits, while multi-source training further improves coverage and robustness.

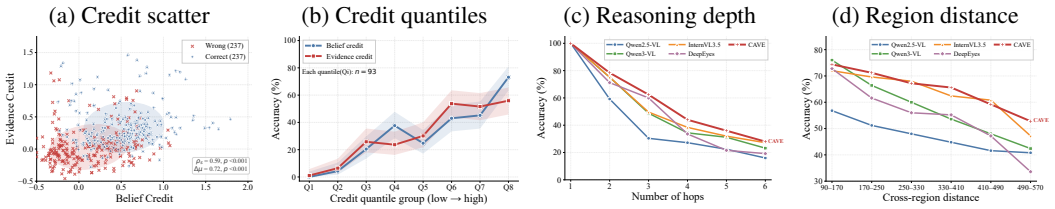


Figure 4: Process-credit validation and difficulty analysis. (a) Distribution of correct and wrong trajectories in the belief-credit and evidence-credit space, with ρ_s and $\Delta\mu$ reporting the credit–correctness correlation and mean credit gap. (b) Accuracy after grouping samples into eight equal-size quantiles from low to high credit values, with shaded Wilson confidence intervals. (c,d) Accuracy on 1,500 fragmented-vision samples grouped by required reasoning hops and cross-region distance, respectively, where CAVE shows more stable performance under increasing difficulty.

Table 5: Credit assignment granularity and training efficiency. (a) Accuracy under different advantage schemes. (b) Performance under different teacher-forcing (TF) scoring budgets. Accuracy is reported in %; TF Time is measured in hours and excludes other training costs.

(a) Advantage granularity.

Scheme	Tunnel Vision	Baby Vision	TRACER -Core	MME -RS
Trajectory-level	**31.3**	**17.3**	**36.9**	59.2
Type-local	29.8	16.5	35.4	57.6
Return-to-Go local	30.7	16.8	36.2	59.6
Hybrid	31.0	16.8	36.5	**60.0**

(b) Scoring cost and sensitivity.

Method	TF Time (h)	TRACER -Bench	Overall
Correct-Only	0.0	35.8	27.8
Full CAVE	6.1	**42.5**	**33.3**
Evidence-50	4.0	42.1	33.0
Step-50	3.4	41.8	32.7

5.4 Process-credit validation and difficulty robustness

Beyond the main results and ablation studies, we further validate the consistency between process credits and successful reasoning, and analyze the robustness of CAVE under different fragmented vision reasoning difficulty conditions.

Validity of process gains. To avoid self-validation bias on the model trained by CAVE, we conduct the analysis using an interleaved reasoning model that has only undergone cold-start training, which has not yet been directly optimized by belief gain or evidence gain [11, 13, 14]. As shown in Figure 4(a)(b), correct trajectories are overall more concentrated in regions with higher belief/evidence gain; after grouping by gain percentiles, the final accuracy also generally increases as the gain becomes larger. Among them, belief gain exhibits a more stable positive correlation, indicating that it can characterize whether the answer formation process progresses in the correct direction; evidence gain shows certain fluctuations, but is also positively correlated with correctness, suggesting that the acquisition of key intermediate evidence contributes to final reasoning. These results indicate that the two types of state-level process signals are not arbitrary heuristics, but are empirically consistent with successful visual evidence reasoning progression.

Robustness under increasing difficulty. We further analyze task difficulty from two dimensions: cross-region distance and reasoning hops, which characterize the spatial span of key evidence and the depth of the dependency chain, respectively [7, 8]. As shown in Figure 4(c)(d), as distance and the number of hops increase, the accuracy of all models generally decreases, indicating that long-range evidence localization, multi-step state maintenance, and region transition substantially increase reasoning difficulty. In contrast, CAVE maintains leading performance in most intervals and exhibits slower performance degradation, suggesting that its benefits are not limited to simple samples, but can still advance the visual evidence chain more stably under longer-range and deeper cross-region dependencies. These results further indicate that the advantages of CAVE mainly come from more effective evidence selection and state maintenance, rather than improvements only on low-difficulty samples.

5.5 Credit Assignment Granularity and Training Efficiency

CAVE measures structured credits at individual state transitions but aggregates them into trajectory-level rewards, assigning a shared GRPO advantage to all assistant-generated tokens within each

Table 6: Budget-controlled comparison of process behavior on fragmented vision reasoning benchmarks. Avg. tokens counts only assistant-generated tokens, excluding tool observations and environment-injected messages. Useful zoom rate is used as a process diagnostic and denotes the proportion of zoom actions that yield positive belief or evidence gain after observation.

Method	Acc. (%)	Avg. Tokens	Avg. Rounds	Avg. Tool Calls	Useful Zoom Rate	Time / Sample (s)
Cold-start	26.5	1935	3.12	2.46	21.4	5.24
DeepEyes	28.5	1643	**2.48**	**1.97**	43.1	4.53
CAVE-7B	**33.3**	**1566**	2.51	2.03	**55.4**	**4.27**

rollout. To examine optimization granularity, we compare four schemes under identical settings: *Trajectory-level* uses aggregated rewards, *Type-local* uses immediate credits, *Return-to-Go local* accumulates future credits, and *Hybrid* combines trajectory-level and return-to-go signals. During rollout, actions and observations are sequentially appended to the context; teacher-forced scoring uses a fixed policy snapshot without gradients, while evidence annotations serve only reward computation. Policy-generated zooms are evaluated using pre-resampling crop areas and post-observation progress. As shown in Table 5(a), trajectory-level optimization achieves the best accuracy on three benchmarks, suggesting that trajectory-wide aggregation better captures delayed action contributions. We further evaluate training efficiency in Table 5(b). Full CAVE improves TRACER-Bench accuracy from 35.8 to 42.5 and overall accuracy from 27.8 to 33.3, requiring 6.1 hours of teacher-forced (TF) scoring. *Evidence-50* and *Step-50* reduce TF overhead to 4.0 and 3.4 hours, respectively, while retaining overall accuracies of 33.0 and 32.7. These results demonstrate the effectiveness of trajectory-level optimization and a favorable accuracy–cost trade-off under reduced scoring budgets.

5.6 Process behavior, efficiency, and training stability

We further conduct diagnostic analyses to verify whether the gains of CAVE come from more reasonable process behavior, rather than longer outputs, more tool calls, or unstable training. First, the reasoning trajectory visualizations in Appendix A show that, compared with the cold-start model and general look-again methods [11, 12], CAVE is more likely to localize task-relevant regions and continuously update intermediate states around the acquired evidence, exhibiting clearer evidence-grounded reasoning behavior. Second, we report the average number of generated tokens, reasoning rounds, zoom calls, useful zoom rate, and inference time in Table 6. The results show that CAVE achieves higher accuracy under a comparable or even lower token/tool budget than other interleaved reasoning or tool-augmented methods [10–12], and obtains a higher useful zoom rate, indicating that its improvement mainly comes from more effective visual focusing and evidence acquisition. Finally, the training dynamics show that the structured rewards and their components improve steadily, as shown in Appendix D.3, while the KL remains within a controllable range, with no obvious policy collapse or abnormal drift observed [15, 46]. Taken together, the improvements of CAVE are reflected not only in final accuracy, but also in more effective evidence localization, higher-quality tool use, and a stable reinforcement learning optimization process.

6 Conclusion

This paper studies Fragmented Visual Reasoning, where VLMs must integrate spatially distributed and semantically weakly separable evidence under low language priors. To systematically evaluate this challenge, we construct TRACER-Bench with diverse cross-region scenarios and verifiable intermediate evidence. To address the difficulty of attributing interleaved reasoning trajectories, we propose CAVE, which assigns process credits for visual evidence exploration from three action-level dimensions: belief update, evidence acquisition, and adaptive focus control. These credits guide the model to discover task-critical evidence, update reasoning states, and focus on informative regions. Experiments show that CAVE consistently improves fragmented visual reasoning performance while maintaining strong general multimodal capabilities. Further analyses show that CAVE advances visual evidence chains more effectively and exhibits stronger robustness under long-range and deep cross-region dependencies.

7 Funding and Competing Interests

This work received no external funding or third-party support. The authors declare no competing financial interests related to this work.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tianmin Han, Zhitao Gong, Sina Samangooei, William Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

[2] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, et al. Qwen2.5-VL technical report, 2025.

[5] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, 2022.

[6] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*, 2024.

[7] Shmuel Berman and Jia Deng. VLMs have tunnel vision: Evaluating nonlocal visual reasoning in leading VLMs. In *Advances in Neural Information Processing Systems*, 2025. URL `https://vlmtunnel.github.io/`. Spotlight presentation.

[8] Liang Chen, Weichu Xie, Yiyan Liang, Hongfeng He, Hans Zhao, Zhibo Yang, Zhiqi Huang, Haoning Wu, Haoyu Lu, Y. Charles, Yiping Bao, Yuantao Fan, Guopeng Li, Haiyang Shen, Xuanzhong Chen, Wendong Xu, Shuzheng Si, Zefan Cai, Wenhao Chai, Ziqi Huang, Fangfu Liu, Tianyu Liu, Baobao Chang, Xiaobo Hu, Kaiyuan Chen, Yixin Ren, Yang Liu, Yuan Gong, and Kuan Li. BabyVision: Visual reasoning beyond language, 2026.

[9] Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *Advances in Neural Information Processing Systems*, 2024.

[10] Zhaochen Su, Linjie Li, Mingyang Song, Yunzhuo Hao, Zhengyuan Yang, Jun Zhang, Guanjie Chen, Jiawei Gu, Juntao Li, Xiaoye Qu, and Yu Cheng. OpenThinkIMG: Learning to think with images via visual tool reinforcement learning. *arXiv preprint arXiv:2505.08617*, 2025.

[11] Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning, 2025.

[12] Xintong Zhang, Zhi Gao, Bofei Zhang, Pengxiang Li, Xiaowen Zhang, Yang Liu, Tao Yuan, Yuwei Wu, Yunde Jia, Song-Chun Zhu, and Qing Li. Chain-of-Focus: Adaptive visual search and zooming for multimodal reasoning via RL, 2025.

[13] Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, Lewei Lu, Haodong Duan, Yu Qiao, Jifeng Dai, and Wenhai Wang. VisualPRM: An effective process reward model for multimodal reasoning, 2025.

[14] Guoqing Wang, Sunhao Dai, Guangze Ye, Zeyu Gan, Wei Yao, Yong Deng, Xiaofeng Wu, and Zhenzhe Ying. Information gain-based policy optimization: A simple and effective approach for multi-turn search agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=qkWP6phrvZ`.

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

[16] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, et al. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[17] Bo Li, Kaichen Zhang, Biao Zhang, et al. LLaVA-OneVision: Easy visual task transfer, 2024.

[18] Yi-Fan Zhang, Huanyu Zhang, Haochen Tian, Chaoyou Fu, Shuangqing Zhang, Junfei Wu, Feng Li, Kun Wang, Qingsong Wen, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. MME-RealWorld: Could your multimodal llm challenge high-resolution real-world scenarios that are difficult for humans? In *The Thirteenth International Conference on Learning Representations*, 2025.

[19] Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, et al. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, 2024.

[20] Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, et al. 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*, 2024.

[21] Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. DDCoT: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. In *Advances in Neural Information Processing Systems*, 2023.

[22] Guowei Xu, Peng Jin, Ziang Wu, Hao Li, Yibing Song, Lichao Sun, and Li Yuan. LLaVA-CoT: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*, 2024. ICCV 2025.

[23] Minheng Ni, Zhengyuan Yang, Linjie Li, Chung-Ching Lin, Kevin Lin, Wangmeng Zuo, and Lijuan Wang. Point-RFT: Improving multimodal reasoning with visually grounded reinforcement finetuning, 2025.

[24] Qiong Wu, Xiangcong Yang, Yiyi Zhou, Chenxin Fang, Baiyang Song, Xiaoshuai Sun, and Rongrong Ji. Grounded chain-of-thought for multimodal large language models. *arXiv preprint arXiv:2503.12799*, 2025.

[25] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[26] Dídac Surís, Sachit Menon, and Carl Vondrick. ViperGPT: Visual inference via python execution for reasoning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

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

[28] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, Ruochen Xu, and Tiancheng Zhao. VLM-R1: A stable and generalizable R1-style large vision-language model, 2025.

[29] Jingyi Zhang, Jiaxing Huang, Huanjin Yao, Shunyu Liu, Xikun Zhang, Shijian Lu, and Dacheng Tao. R1-VL: Learning to reason with multimodal large language models via step-wise group relative policy optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[30] Yihe Deng, Hritik Bansal, Fan Yin, Nanyun Peng, Wei Wang, and Kai-Wei Chang. OpenVLThinker: An early exploration to complex vision-language reasoning via iterative self-improvement. *arXiv preprint arXiv:2503.17352*, 2025.

[31] Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21805–21830, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1106. URL https://aclanthology.org/2025.emnlp-main.1106/.

[32] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. GiGPO: Group-in-group policy optimization for LLM agent training, 2025.

[33] xAI. RealWorldQA: A benchmark for evaluating real-world spatial understanding in multimodal ai models. https://huggingface.co/datasets/xai-org/RealworldQA, 2024.

[34] Lin Chen et al. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, 2024.

[35] OpenAI. Introducing GPT-5.2. https://openai.com/index/introducing-gpt-5-2/, 2025. Accessed: 2026-04-26.

[36] Google DeepMind. Gemini 3.1 pro model card. https://deepmind.google/models/model-cards/gemini-3-1-pro/, 2026. Accessed: 2026-04-26.

[37] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[38] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

[39] Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, et al. MiniCPM-V 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*, 2025.

[40] Yi Yang, Xiaoxuan He, Hongkun Pan, Xiyan Jiang, Yan Deng, Xingtao Yang, et al. R1-onevision: Advancing generalized multimodal reasoning through cross-modal formalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

[41] Xiyao Wang, Zhengyuan Yang, Chao Feng, Hongjin Lu, Linjie Li, Chung-Ching Lin, Kevin Lin, Furong Huang, and Lijuan Wang. Sota with less: Mcts-guided sample selection for data-efficient visual reasoning self-improvement. *arXiv preprint arXiv:2504.07934*, 2025.

[42] Hardy Chen, Haoqin Tu, Fali Wang, Hui Liu, Xianfeng Tang, Xinya Du, Yuyin Zhou, and Cihang Xie. SFT or RL? an early investigation into training R1-like reasoning large vision-language models, 2025.

[43] Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel reasoner: Incentivizing pixel space reasoning via curiosity-driven reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

[44] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024.

[45] Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, et al. MathVerse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, 2024.

[46] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, et al. HybridFlow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, 2025.

A Case study on rollout trajectories

Figure 5 and Figure 6 present a comparison of rollout trajectories before and after CAVE training. The example requires the model to identify which candidate fish is exactly identical to the target fish. The correct decision depends on several confusable local cues. We use this case to visualize how different models acquire visual evidence through zoom-in actions and how such evidence affects subsequent elimination of candidates.

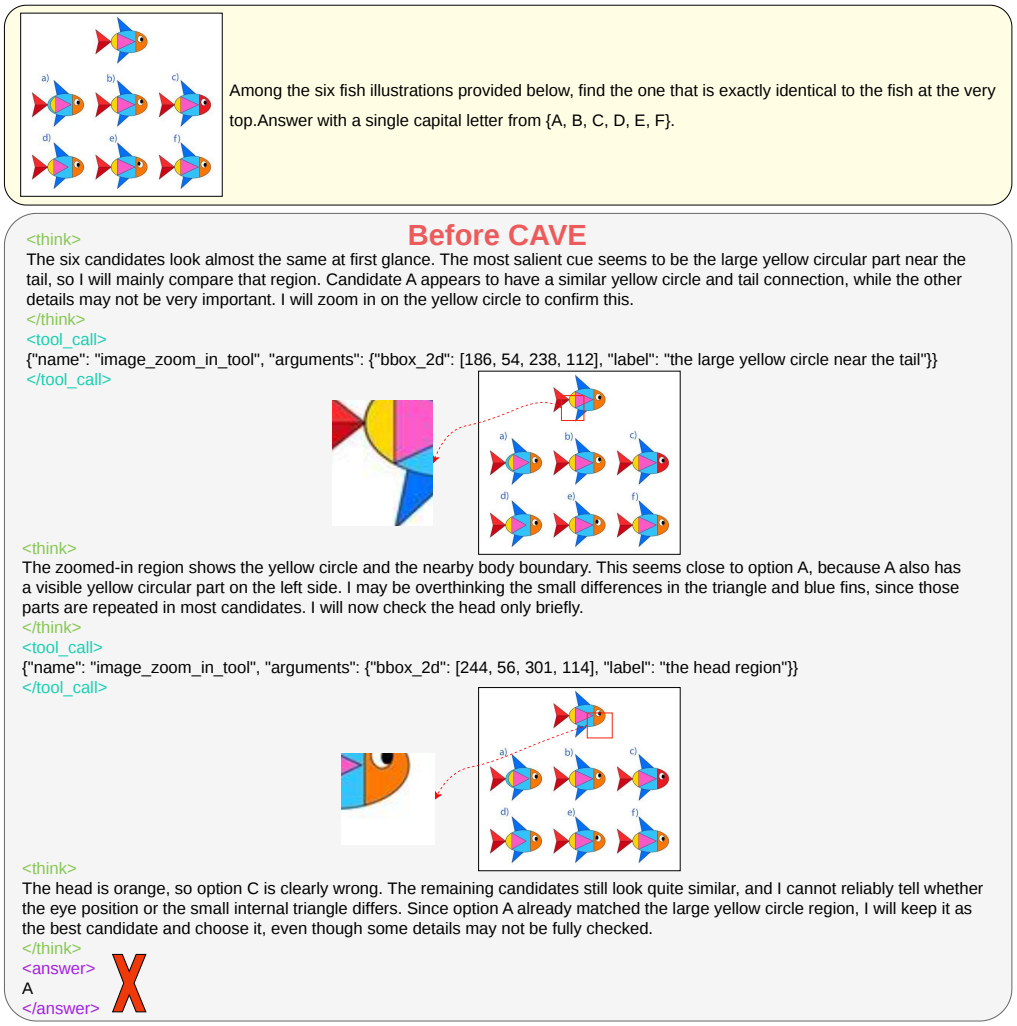
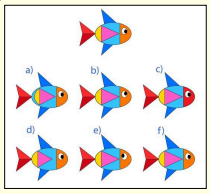


Figure 5: Case study of rollout trajectories before CAVE training.

B TRACER-Bench

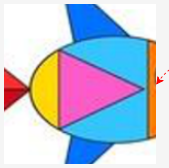
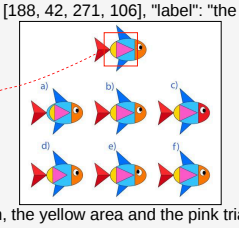
This section provides additional details of TRACER-Bench, including scenario design, dataset statistics and quality control. TRACER-Bench is designed to evaluate cross-region dependency reasoning under controlled yet diverse visual structures. Each sample follows a unified `prompt-answer-perception` format, where the `perception` field records key intermediate visual evidence required to solve the task. This design allows the benchmark to support both final-answer evaluation and process-level analysis.



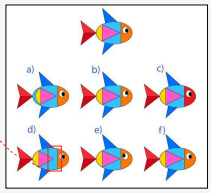
Among the six fish illustrations provided below, find the one that is exactly identical to the fish at the very top.Answer with a single capital letter from {A, B, C, D, E, F}.

After CAVE

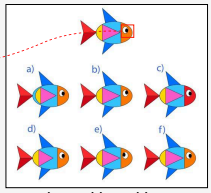
<think>
I need to identify which candidate fish is exactly identical to the target fish at the top. From the full image, all six candidates share the same overall fish shape, with a red tail, blue fins, a colored body, and an orange or red head. A coarse comparison is not sufficient to distinguish them reliably. The most informative differences are likely to lie in the internal body structure and the head region. I should first zoom in on a larger body region to inspect the relative positions of the yellow area and the pink triangular region.
</think>
<tool_call>
{"name": "image_zoom_in_tool", "arguments": {"bbox_2d": [188, 42, 271, 106], "label": "the large body region including the yellow area and the pink triangle"}}
</tool_call>

<think>
Now I observe the zoomed-in body region. In the target fish, the yellow area and the pink triangle have a specific relative layout. Comparing this with the candidates, option A shows a mismatch in the relative position of the yellow area, so A cannot be identical to the target fish. Among the remaining candidates, option D is still visually confusable because its overall body colors look similar. However, the front tip of the pink triangle in option D appears to have a different alignment from the target fish. I should zoom in on this local triangular tip to verify whether D can be eliminated.
</think>
<tool_call>
{"name": "image_zoom_in_tool", "arguments": {"bbox_2d": [54, 308, 72, 354], "label": "the front tip of the pink triangle in option D"}}
</tool_call>

<think>
Now I observe the zoomed-in triangular tip in option D. In the target fish, the front tip of the pink triangle points toward a specific position within the blue body region, while in option D this triangular tip is shifted and does not align with the target structure. Therefore, D can be eliminated. After eliminating A and D, I should inspect the head region, since the eye position and head color may help distinguish the remaining candidates.
</think>
<tool_call>
{"name": "image_zoom_in_tool", "arguments": {"bbox_2d": [267, 54, 291, 79], "label": "the head region including the eye and head color"}}
</tool_call>

<think>
Now I observe the zoomed-in head region. In the target fish, the eye is positioned in a specific relative location within the orange head. Comparing this local feature across the remaining candidates, options B and E show mismatched eye positions, so they cannot be identical to the target fish.Next, I compare the head color. In the target fish, the head is orange. Option C has a red head instead of an orange head, so C is also eliminated.After eliminating B, E, and C, only option F remains. Therefore, the fish that is exactly identical to the target fish is F.
</think>
<answer>
F
</answer>

Figure 6: Case study of rollout trajectories after CAVE training.

B.1 Scenario design

Table 7 summarizes four scenario families in TRACER-Bench. These scenarios are designed to probe basic yet challenging visual reasoning abilities of VLMs under cross-region dependency, where successful prediction requires not only recognizing local visual patterns but also maintaining and propagating visual evidence across spatially separated regions.

Table 7: Scenario definitions of TRACER-Bench. "RS Subimage Matching" denotes Remote-Sensing Subimage Matching.

Scenario	Required evidence	Answer type
Rule-Switching Navigation	Node attributes, directed edges, and rule switches	Final target node
Nonsemantic Tracing	Path identity under distractors	Endpoint label
Embedded Matching	Fine-grained template-grid correspondence	Binary decision
RS Subimage Matching	Local image patch, scale, and rotation consistency	Binary decision

Figure 7 shows a sample of Rule-Switching Navigation. The model must maintain the current node as an intermediate state and update it according to the rule specified at each step. Because arrow-following and color-matching rules appear in an interleaved manner, the next target region may depend on either local directional cues or nonlocal attribute matching. This makes the evidence chain both spatially distributed and easily confusable: the model has to distinguish valid arrows from nearby distractors, compare colors across separated nodes, and propagate the current state through multiple transitions. Thus, the task tests whether VLMs can perform rule-conditioned state tracking and nonlocal evidence integration, rather than relying on single-region recognition.

Figure 8 shows a sample of Nonsemantic Tracing. The model must follow a continuous curve from a specified letter endpoint to its corresponding numbered endpoint, without relying on semantic cues such as components, ports, or object identities. The task is inherently nonlocal because the correct endpoint is often spatially distant from the starting letter, and the model has to maintain path identity across a long curved trajectory. At the same time, the curve is easily confusable: multiple colored paths may overlap, run in parallel, or pass through nearby regions, while background noise introduces additional visual distractions. The curves are procedurally generated with randomized Bezier control points, which produces smooth but variable direction changes and prevents the model from relying on simple straight-line extrapolation. Thus, the task tests whether VLMs can perform continuous visual tracking under distractors, rather than solving the problem through local endpoint recognition or semantic priors.

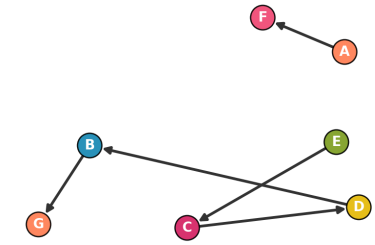
Figure 9 shows a sample of Embedded Matching. The model must determine whether the local template appears exactly within the larger structure, requiring simultaneous comparison of multiple shapes, their relative arrangement, and their rotations. This setting is easily confusable because rotated shapes may appear visually similar while still differing in orientation or local configuration, and such differences are difficult to describe completely within a bounded text representation. Moreover, matching multiple elements requires the model to retain previous comparison states while checking subsequent positions in the larger image. Thus, the task tests confusable local-to-global structural correspondence beyond single-shape recognition.

Figure 10 shows a sample of Remote-Sensing Subimage Matching. We derive the original remote-sensing images from MME-RealWorld [18], while constructing new subimage-matching instances rather than reusing its original QA annotations. Compared with abstract scenes, these images contain richer visual content, including dense buildings, roads, shadows, and background clutter. The model must determine whether the candidate subimage corresponds to a region in the original image, even when rotation or scale changes are applied. These transformations make the evidence more confusable, since visually similar urban structures may appear at different orientations or resolutions. Thus, the task tests whether VLMs can maintain robust local-to-global correspondence in realistic scenes with complex background variation.

B.2 Dataset statistics

Table 8 reports the dataset statistics of TRACER-Bench. All scenarios are organized with the same annotation interface. For binary matching tasks, we balance positive and negative samples. For

Rule-Switching Navigation



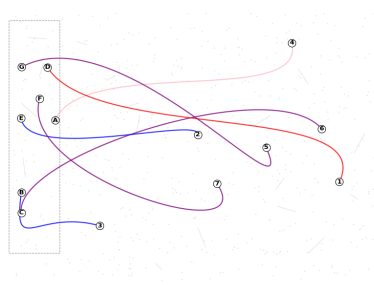
prompt:
Start from point E. First, follow the direction of the arrow. Second, follow the direction of the arrow. Third, follow the direction of the arrow. Fourth, follow the direction of the arrow. Fifth, jump to the other point that has the same color as the current point. Sixth, follow the direction of the arrow. Which point do you finally reach? Answer with the single letter only, e.g {C}.

answer:
F

perception:
The first jump starts at point E and ends at point C, following an arrow.

Figure 7: Representative example of Rule-Switching Navigation.

Nonsemantic Tracing



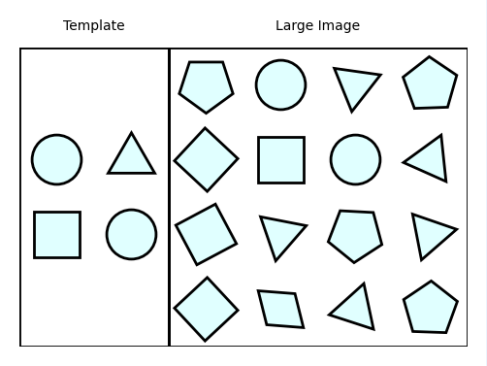
prompt:
Starting from the letter C, which number does the line end at? Each line in the picture starts with a letter and ends with a number. The color of each line does not change. Only answer the number, e.g {3}.

answer:
6

perception:
The letter C is connected to the number 6.

Figure 8: Representative example of Nonsemantic Tracing.

Embedded Matching



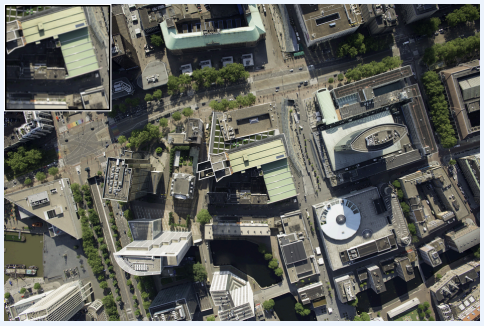
prompt:
The left side shows a 2×2 template inside a black box, consisting of basic shapes (triangle, square, circle, parallelogram, pentagon) each with a specific rotation angle and uniform visual size. The right side shows a 4×4 large image inside a black box (sharing a common edge with the template box) filled with the same types of shapes (uniform visual size, random rotation except for template area). Please check if the exact shape pattern of the template (including the identical rotation angle for each corresponding shape) exists in the large image. Respond with only {yes} or {no}.

answer:
no

perception:
Equilateral triangle in template points upward.

Figure 9: Representative example of Embedded Matching.

Remote-Sensing Subimage Matching



prompt:
The original image is the remote sensing image, while the sub-image exists in the upper left or upper right corner of the image. After a possible certain degree of rotation and scaling, does the sub-image (the entire image as a whole) exactly match a certain area of the original image? Please answer {yes} or {no}. As long as the sub-image matches a certain area of the original image, the answer will be {yes}.

answer:
yes

perception:
Sub-image contains light-roofed rectangular buildings.

Figure 10: Representative example of Remote-Sensing Subimage Matching.

chain-based and path-based tasks, we stratify the samples according to the corresponding difficulty factors, such as the number of nodes, path complexity, or distractor density.

Table 8: Statistics of TRACER-Bench.

Scenario	#Samples	Answer type	Main difficulty factors
Rule-Switching Navigation	245	Multi-class	Nodes / transitions / distractors
Nonsemantic Tracing	245	Multi-class	Number of lines / path ambiguity
Embedded Matching	245	Binary	Candidate size / structural distractors
RS Subimage Matching	245	Binary	Scale / rotation / background complexity
Total	980	–	–

Note: "RS Subimage Matching" denotes Remote-Sensing Subimage Matching.

Data format Each TRACER-Bench sample is stored as one JSONL entry. Table 9 describes the unified field format. The `perception` field is not intended to be a free-form explanation. Instead, it records task-critical intermediate visual evidence derived from the generation process or annotation procedure, and is used only for process-level analysis or supervision. A representative JSONL example is shown as listing 1.

Listing 1: Example JSONL entry from the Embedded Matching scenario.

```
{
  "id": "match_0",
  "image": "./images/exp_0000.png",
  "prompt": "The left side shows a 2x2 template inside a black box, consisting of basic
      shapes (triangle, square, circle, parallelogram, pentagon) each with a specific
      rotation angle and uniform visual size. The right side shows a 4x4 large image
      inside a black box (sharing a common edge with the template box) filled with the
      same types of shapes (uniform visual size, random rotation except for template
      area). Please check if the exact shape pattern of the template (including the
      identical rotation angle for each corresponding shape) exists in the large image.
      Respond with only {yes} or {no}.",
  "answer": "yes",
  "perception": "Equilateral triangle in template points upward."
}
```

Difficulty factors TRACER-Bench controls difficulty through task-specific factors that correspond to different forms of cross-region dependency. Table 10 summarizes the main factors used in each scenario. These factors are not intended to define a single universal difficulty score. Instead, they provide interpretable axes for analyzing how model performance changes under longer dependency chains, denser visual distractors, or more complex local-to-global correspondence.

B.3 Quality control

For synthetic scenarios, final answers and perception fields are derived from the underlying generation states, such as sampled paths, template configurations, endpoint mappings, or patch corre-

Table 9: Unified sample format of TRACER-Bench.

Field	Description
`id`	Unique sample identifier.
`image`	Relative path to the input image.
`prompt`	Natural-language instruction describing the visual reasoning task.
`answer`	Final target label or binary decision.
`perception`	Key intermediate visual evidence required to solve the task.

Table 10: Difficulty factors used in TRACER-Bench.

Difficulty factor	Definition	Applied scenarios
Dependency length	Number of required reasoning hops or state updates.	VJump
Structural distractors	Number of irrelevant shapes or candidate substructures.	Match
Cross-region distance	Spatial span between key evidence regions.	VJump, LT
Scale/rotation variation	Scale or rotation change between the candidate and source patch.	Match, MME-RS

spondences. To ensure annotation reliability, the generated samples were manually inspected by the authors to check the consistency among visual content, prompts, final answers, and perception fields. For binary matching tasks, we maintain a balanced positive/negative distribution. For three abstract scenarios, we perform stratified sampling across difficulty levels, such as dependency length, path complexity, candidate size, and visual distractor density. This prevents the benchmark from being dominated by overly simple or overly difficult cases, and yields a more balanced evaluation of model performance across different reasoning conditions. For remote-sensing subimage matching, candidate patches are selected to preserve sufficient visual information and avoid overly confusable or low-texture crops. Samples with unclear correspondences, severe visual artifacts, or inconsistent annotations are removed during inspection.

The perception field is designed as a compact description of task-critical visual evidence rather than a free-form rationale. Because the target tasks are precisely those where current VLMs are brittle, using AI-generated rationales to annotate evidence would make the evidence reliability questionable and substantially increase the burden of manual verification. Instead, we derive perception units from the latent generation state of each scripted scenario, which allows us to control the evidence type, difficulty, and dependency structure during data construction. We keep only evidence necessary for solving the instance, such as local attributes, path transitions, rule-dependent correspondences, endpoint relations, or patch-level matches, and avoid redundant explanations or direct restatements of the final answer. During inspection, we check whether each perception unit is visually grounded, necessary for the answer, and consistent with the prompt and generated image. The perception field is used only for computing process rewards and offline analysis, and is never provided to the model during evaluation.

C Training data construction and split protocol

This section provides additional details on the training data used for CAVE reinforcement learning. The training corpus shares the same scenario families with TRACER-Bench, but is generated independently from the benchmark. In addition, we construct non-overlapping Tunnel-Vision-style synthetic data as auxiliary training data and convert it into the unified **prompt-answer-perception** format.

C.1 Training corpus composition

Figure 11 summarizes the composition of the training corpus, which combines four TRACER-style scenario families with three Tunnel-Vision-style auxiliary scenario types. The auxiliary Tunnel-Vision-style data is used only for training, is not part of TRACER-Bench, and does not overlap with the public Tunnel Vision evaluation set.

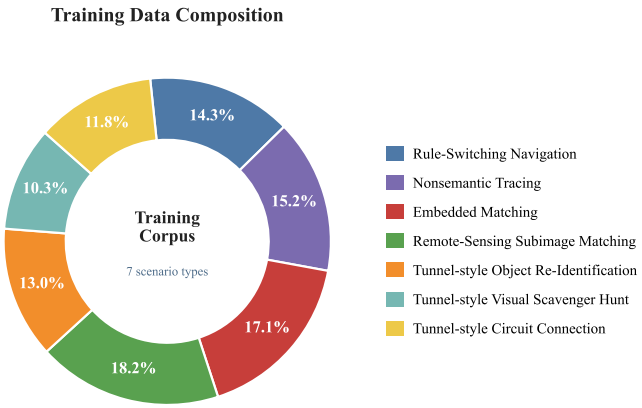


Figure 11: Composition of the training corpus used for CAVE reinforcement learning. The corpus consists of four TRACER-style scenario families and three independently generated Tunnel-Vision-style auxiliary scenario types.

Table 11: Split and isolation protocol between the training corpus and TRACER-Bench.

Isolation factor	**Protocol**
Random seeds	Training and benchmark samples are generated with disjoint random seeds.
Generated layouts	Exact generated layouts are not reused between the training corpus and TRACER-Bench.
Rule configurations	Training and benchmark data may share abstract rule types, while concrete ordered rule sequences are independently sampled rather than copied across splits.
Image regions	For real-image scenarios, training and benchmark patches are sampled from disjoint images or non-overlapping regions.
Concrete instances	No identical image–question–answer tuple appears in both the training corpus and TRACER-Bench.
Shared difficulty axes	Training and benchmark data may share difficulty axes, such as dependency length, cross-region distance, and distractor level, but not concrete generated configurations.
Tunnel-Vision-style data	Auxiliary Tunnel-Vision-style training data is generated independently and does not overlap with the public Tunnel Vision evaluation set [7].

C.2 Split and isolation protocol

To prevent instance leakage, we separate the training corpus and TRACER-Bench at both the sample and generator-configuration levels, as summarized in Table 11. This protocol reduces the risk of instance memorization and encourages evaluation on unseen layouts, image regions, and dependency configurations.

C.3 Tunnel-Vision-style example and perception

Figure 12 shows a representative Tunnel-Vision-style jumping example used in the training corpus. The task requires the model to follow a sequence of visual transitions across spatially separated regions. To unify such auxiliary data with the main Tracer-style training scenarios, we supplement each sample with a `perception` field.

The `perception` field is constructed from the generation process rather than written as a free-form rationale. For Tunnel-Vision-style jumping scenes, it records the minimal visual evidence needed to support the region transitions, such as the start region, the visual cue used for each jump, and the reached target region. We deliberately keep this field concise: it should describe only the key visual evidence required for supervision, rather than provide a long chain-of-thought explanation. This design allows the auxiliary synthetic data to share the same training interface as the main scenarios while providing a verifiable intermediate signal for process-level optimization.

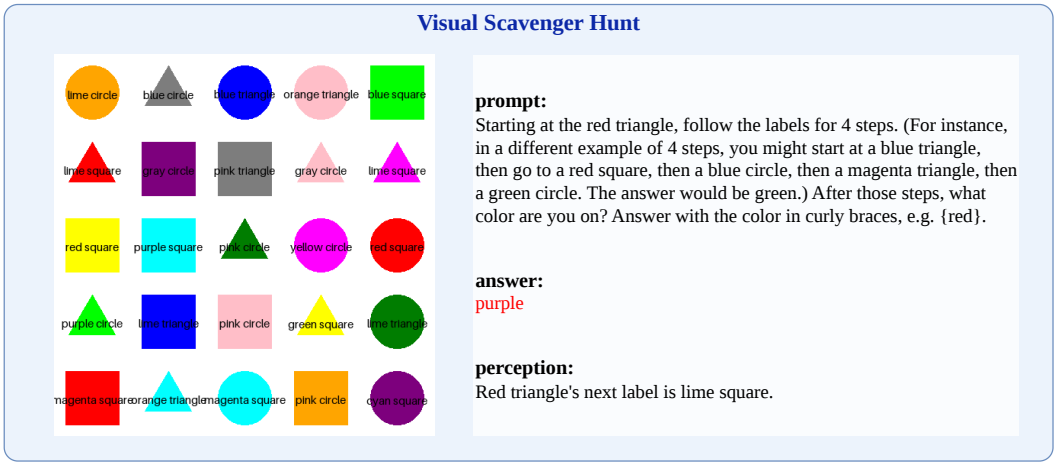


Figure 12: Representative Tunnel-Vision-style jumping example in the training corpus. The sample is converted into the unified prompt-answer-perception format.

D Train progress

D.1 Training and reward hyperparameters

Training details CAVE uses Qwen2.5-VL as the base model and adopts a two-stage training procedure on 8 A800 GPUs [4, 46, 15]. First, we use 10k DeepEyes-style samples for cold-start training, enabling the model to acquire basic interleaved reasoning and tool-use capabilities [11]. Then, we perform process-reward training on 5,000 TRACER-style samples with process evidence annotations [15, 7], and incorporate constraints on answer correctness, format validity, and the number of rounds to stabilize optimization. CAVE-3B adopts the same training procedure but replaces the backbone with a 3B model; Belief-Only keeps the same backbone, data, and optimization settings, while replacing the structured process rewards with answer-belief gain only. Tool observations are used only as contextual feedback, with their response masks set to 0, and do not participate in policy-gradient computation. The weights of the three structured credits are set to a ratio of $\lambda_{\mathrm{bu}} : \lambda_{\mathrm{ea}} : \lambda_{\mathrm{af}} = 4 : 3 : 3$, selected based on validation performance after a small hyperparameter sweep.

Reward aggregation. CAVE combines three structured process credits: belief update, evidence acquisition, and adaptive focus control. For each rollout trajectory τ, the final structured process reward is computed as

$$R_{\mathrm{CAVE}}(\tau) = \lambda_{\mathrm{bu}} C_{\mathrm{bu}}(\tau) + \lambda_{\mathrm{ea}} C_{\mathrm{ea}}(\tau) + \lambda_{\mathrm{af}} C_{\mathrm{af}}(\tau),$$

where C_{bu}, C_{ea}, and C_{af} denote the aggregated credits for answer-belief update, task-critical evidence acquisition, and uncertainty-aware adaptive focus control, respectively. In all main experiments, we use

$$\lambda_{\mathrm{bu}} : \lambda_{\mathrm{ea}} : \lambda_{\mathrm{af}} = 4 : 3 : 3.$$

Equivalently, the normalized weights are 0.4, 0.3, and 0.3.

Selection protocol. We select the reward weights using a small validation-based sweep on the validation split constructed from the training sources, without using any reported test benchmark for hyperparameter selection. We consider several nearby configurations around balanced weighting, including equal weighting and belief-dominant variants. The final ratio $\lambda_{\mathrm{bu}} : \lambda_{\mathrm{ea}} : \lambda_{\mathrm{af}} = 4 : 3 : 3$ gives slightly higher weight to belief update, which provides the most stable answer-side progress signal, while retaining substantial supervision from evidence acquisition and adaptive focus control. This avoids reducing the reward to a single answer-side information-gain objective and improves early-stage RL stability. In all experiments, we set the entropy top-k to $k = 500$, fix the crop-scale range to $\rho_{\min} = 0.02$ and $\rho_{\max} = 0.30$, use a round-decay weight $w_t^m = 0.8^{t-1}$, apply $\phi_m(c) = \mathrm{clip}(c, -1, 2)$ to bound credit magnitudes, and limit each rollout to at most $T = 5$ reasoning rounds.

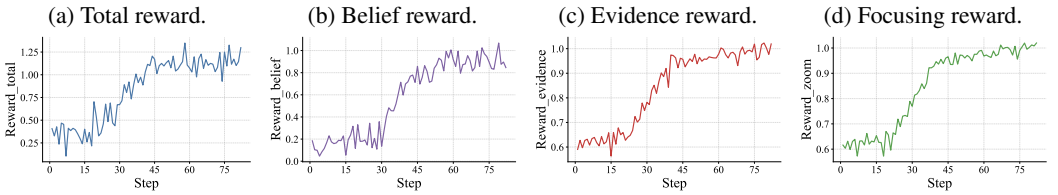


Figure 13: Training dynamics of CAVE process rewards. The total structured reward and its three components—belief update, evidence acquisition, and adaptive focus control—improve steadily, indicating that CAVE effectively optimizes the intended visual evidence progression signals.

Use across experiments. We use the same reward and rollout settings for the final CAVE model and the smaller-backbone CAVE-3B variant. For reward ablations, the corresponding credit term is removed while the remaining terms keep their relative proportions. The Belief-Only baseline keeps the same backbone, cold-start initialization, training data, rollout setting, and optimization recipe, but replaces the structured reward with the belief-update credit alone. Thus, differences between CAVE and its controlled variants mainly reflect the effect of structured credit composition rather than changes in training setup.

Stability considerations. During training, we monitor validation reward, answer accuracy, and KL divergence to avoid reward-scale imbalance or policy drift. The selected weighting yields stable reward improvement without observed policy collapse. Since GRPO uses group-relative advantages, the absolute reward scale is less important than the relative ranking of trajectories within each rollout group; nevertheless, we keep the same reward normalization and clipping strategy for all compared variants to ensure fair comparison.

D.2 Experimental resource consumption

All training experiments are conducted on a single node with 8 NVIDIA A800 80GB GPUs. We implement the reinforcement learning stage with the verl framework and use mixed-precision training and gradient checkpointing to reduce memory consumption. For CAVE-7B, the cold-start supervised fine-tuning stage on 10K interleaved reasoning samples takes approximately 10 hours. The subsequent CAVE process-reward training stage on 5K samples takes approximately 24 hours, including policy rollout, tool-based visual re-observation, structured credit computation, and GRPO optimization.

D.3 Process behavior and efficiency

To verify that the improvement of CAVE does not simply come from longer responses, more reasoning rounds, or more frequent tool use, we compare the inference budget and process behavior of DeepEyes, the cold-start model, and CAVE under the same evaluation protocol. Specifically, we report the average accuracy, number of assistant-generated tokens, reasoning rounds, zoom-tool calls, and inference time per sample. Tool-returned observations are treated as environment feedback and are not counted as policy-generated tokens.

As shown in Table 6, CAVE achieves higher accuracy without requiring a substantially larger inference budget. Compared with the cold-start model, CAVE improves the effectiveness of intermediate reasoning and visual evidence acquisition rather than merely increasing the number of rounds or tool calls. Compared with DeepEyes, CAVE maintains a comparable tool-use budget while achieving stronger performance, suggesting that structured process credits help the model select and exploit task-relevant visual evidence more efficiently.

D.4 Cross-backbone generalization

To further evaluate the transferability of CAVE across different VLM architectures, we conduct an additional experiment using InternVL3.5-8B as the backbone. We retain the same training data, structured reward design, rollout budget, and evaluation protocol as in the Qwen2.5-VL experiments. We compare the original model (Direct), correctness-only reinforcement learning (Correct-Only), and the full CAVE method.

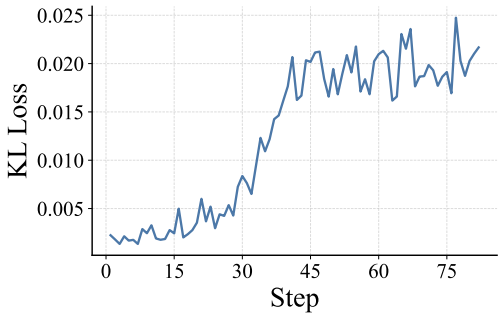


Figure 14: KL dynamics during CAVE training. The KL loss increases in the early stage and then remains within a controlled range, suggesting stable optimization without obvious policy collapse or abnormal drift.

Table 12: Cross-backbone generalization results on InternVL3.5-8B. Aggregate denotes the overall accuracy on fragmented visual reasoning benchmarks, while General Avg. denotes the average performance across six general multimodal benchmarks.

Method	Tunnel Vision	BabyVision	TRACER-Core	MME-RS	Aggregate	General Avg.
Direct	26.9	16.5	30.3	58.4	29.2	61.1
Correct-Only	29.7	17.3	33.5	60.0	31.7	61.3
CAVE	**32.4**	**19.1**	**36.7**	**62.4**	**34.4**	**63.7**

As shown in Table 12, CAVE consistently outperforms both Direct and Correct-Only across all four fragmented visual reasoning task groups. In particular, it improves the aggregate accuracy by 5.2 percentage points over Direct and 2.7 percentage points over Correct-Only. Moreover, the average score across six general multimodal benchmarks increases by 2.6 points over Direct and 2.4 points over Correct-Only. These results suggest that structured credit assignment can improve visual evidence reasoning across different VLM backbone families, rather than relying exclusively on the Qwen2.5-VL architecture.

E Teacher forcing details

Figure 15 illustrates how teacher forcing is used to compute answer-belief gain and evidence gain. After a rollout is generated, we separately score fixed target sequences under each context state. For belief update, the target is the ground-truth answer body; for evidence acquisition, the targets are textualized evidence units. The adjacent-state log-likelihood gain from $\mathcal{M}_t$ to $\mathcal{M}_{t+1}$ is used as the corresponding process credit. This scoring pass is only used for reward computation and does not affect rollout generation or later context states.

F Statistical significance

We conduct a statistical significance analysis for the main same-backbone comparison between CAVE-7B and Qwen2.5-VL*. For each benchmark group, we model accuracy as a binomial proportion over test instances and report 95% confidence intervals. Since the two models are evaluated on the same test instances, we further apply McNemar's test to the paired correct/incorrect outcomes. As shown in Table 13, the observed gains remain statistically meaningful at the aggregate benchmark level, supporting the robustness of the same-backbone improvements.

Paired comparison with Pixel Reasoner. We additionally compare CAVE-7B with the officially released PixelReasoner-RL-v1 checkpoint, without additional fine-tuning, data adaptation, or test-set-specific adjustment. Both models are evaluated on exactly the same test instances using the same image preprocessing, visual-tool interface, maximum number of interaction rounds, decoding configuration, stopping criteria, and answer-parsing procedure. The overall accuracy in this paired comparison is weighted over 2,718 test instances from Tunnel Vision, BabyVision, and TRACER-Bench.

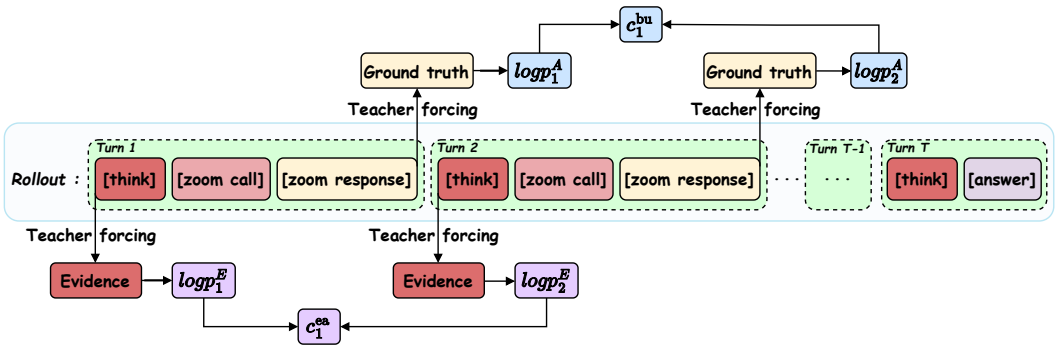


Figure 15: Teacher-forced answer and evidence gains.

Table 13: Statistical significance analysis for the main same-backbone comparison between CAVE-7B and Qwen2.5-VL*. Bracketed values report 95% normal-approximation confidence intervals for aggregate accuracies and accuracy gaps, and p-values are obtained using McNemar's test on paired correct/incorrect outcomes.

Benchmark	N	CAVE-7B	Qwen2.5-VL*	Δ	p-value
Tunnel Vision	1350	31.3 [28.9, 33.8]	24.4 [22.1, 26.7]	+7.0 [3.6, 10.3]	5.4×10^{-5}
BabyVision	388	17.3 [13.5, 21.1]	11.0 [7.8, 14.1]	+6.3 [1.4, 11.2]	1.23×10^{-2}
TRACER-Bench	980	42.5 [39.4, 45.5]	34.0 [31.0, 37.0]	+8.5 [4.2, 12.7]	1.18×10^{-4}

Table 14: Fixed-checkpoint comparison between PixelReasoner-RL-v1 and CAVE-7B under the unified evaluation protocol.

Method	Overall Acc. (%)	Avg. Tokens
PixelReasoner-RL-v1	31.6	1,534
CAVE-7B	**33.3**	1,566

To quantify the performance difference between the two fixed checkpoints, we compute a paired-bootstrap 95% confidence interval based on per-instance correctness and apply an exact two-sided McNemar test to the paired predictions.

Table 15: Paired statistical comparison between CAVE-7B and PixelReasoner-RL-v1. The accuracy difference and confidence interval are in percentage points.

Comparison	Δ Acc.	Paired 95% CI	McNemar p-value
CAVE vs. Pixel Reasoner	$+1.8$	$[+0.5, +3.2]$	0.007

Both checkpoints use deterministic decoding. Accordingly, this paired analysis characterizes their performance difference on the same test instances, rather than variability across training seeds. The confidence interval for the accuracy difference lies entirely above zero, and the exact McNemar test is significant ($p = 0.007$), providing statistical support for CAVE's improvement under this evaluation protocol. Multi-seed variability remains untested.

G Limitations

Our empirical evaluation focuses on open-source VLM backbones at moderate model scales, allowing controlled same-backbone comparisons and detailed analysis of intermediate reasoning behaviors. While CAVE is not architecture-specific and only assumes an autoregressive interleaved reasoning interface, its scaling behavior on substantially larger VLMs remains an interesting direction for future investigation. Moreover, the current implementation uses localized zoom-based re-observation as the main visual action space. Future work may extend structured credit assignment to richer multimodal actions, such as explicit region comparison, object-level grounding, and external visual tool invocation.

Table 16: Existing assets used in this work and their licenses or terms of use.

Asset	Use in this work	License / terms
Qwen2.5-VL	Base model and baseline	Apache-2.0
Qwen3-VL	Baseline evaluation	Apache-2.0
InternVL3.5	Baseline evaluation and cross-backbone CAVE training	Apache-2.0
MiniCPM-V 4.5	Baseline evaluation	Apache-2.0
R1-OneVision	Baseline evaluation	Apache-2.0
ThinkLite-VL	Baseline evaluation	MIT
DeepEyes	Baseline/cold-start data reference	Apache-2.0
PixelReasoner-RL-v1	Baseline evaluation	Apache-2.0
Tunnel Vision	Benchmark and data reference	BSD-3-Clause
BabyVision	Public benchmark evaluation	MIT
MME-RealWorld	Remote-sensing source and evaluation	Apache-2.0
MMStar	General multimodal benchmark evaluation	Apache License 2.0
RealWorldQA	General multimodal benchmark evaluation	CC-BY-ND-4.0
HallusionBench	General multimodal benchmark evaluation	BSD-3-Clause
BLINK	General multimodal benchmark evaluation	Apache-2.0
MathVista	General multimodal benchmark evaluation	CC-BY-SA-4.0
MathVerse	General multimodal benchmark evaluation	MIT

H Broader impacts

CAVE is intended to improve the reliability and interpretability of VLMs by encouraging models to acquire and integrate task-relevant visual evidence during interleaved reasoning. Together with TRACER-Bench, it provides a controlled framework for diagnosing whether models rely on grounded visual evidence rather than superficial language priors. These properties may benefit the development of more robust multimodal systems.

However, stronger visual evidence exploration may also be misused in sensitive visual analysis scenarios, such as surveillance or fine-grained monitoring. Moreover, performance gains on controlled benchmarks should not be taken as sufficient evidence for safe deployment in high-stakes applications. We therefore position this work as a research contribution to evidence-grounded visual reasoning, and encourage task-specific validation, human oversight, and appropriate safeguards before any real-world use.

I Existing assets and licenses

This work uses existing assets only for model initialization, baseline comparison, evaluation, and controlled data construction. We properly credit the original creators through citations to the corresponding papers, model cards, datasets, or repositories. We use all third-party assets under their released licenses or terms of use, and we do not claim ownership of any existing model, dataset, benchmark, or codebase.

Table 16 summarizes the main existing assets used in this work. Public model checkpoints, including Qwen2.5-VL, Qwen3-VL, InternVL3.5, MiniCPM-V 4.5, R1-OneVision, ThinkLite-VL, and DeepEyes, are used only for initialization or baseline evaluation under their respective licenses. Existing benchmarks and datasets, including Tunnel Vision, BabyVision, MME-RealWorld, MMStar, RealWorldQA, HallusionBench, BLINK, MathVista, and MathVerse, are used only for research evaluation or controlled data construction. We do not currently redistribute third-party model weights or restricted benchmark assets. Any future release of derived samples and metadata will comply with the applicable source licenses and data-access restrictions.

J Additional characterization of fragmented visual evidence reasoning

Given an image I and a question q, we view a fragmented visual evidence reasoning instance as requiring a set of task-critical visual observations distributed across multiple regions. Let $G_q = (V_q, E_q)$ denote a query-induced dependency graph, where each node $v_i \in V_q$ corresponds to an

image region $R_i \subset I$ and each edge $e_{ij} \in E_q$ indicates that evidence from region R_i is needed to determine or verify evidence in region R_j. We denote the corresponding task-critical visual evidence units as

$$\mathcal{P}_q = \{p_1, \ldots, p_K\},$$

where each evidence unit p_k is associated with a node, an edge, or a small subgraph in G_q, such as a local attribute, a path transition, or a cross-region correspondence. The answer can then be written abstractly as

$$y = g(\mathcal{P}_q, q),$$

where $\mathcal{P}_q$ denotes the visual evidence required by the question.

Such tasks are characterized by two properties. First, the required evidence is often nonlocal: solving the instance requires integrating information from spatially separated regions rather than recognizing a single local object. Second, the evidence is difficult to fully preserve through short textual descriptions. Fine-grained attributes, path continuity, local–global correspondences, and visually confusable differences may be lost or become confusable once compressed into language.

These properties make fragmented visual evidence reasoning naturally sequential. At each step, the model must decide which region to inspect, which evidence to retain, and how to update its current hypothesis based on previous observations. Therefore, the value of an intermediate action may not be immediately reflected in final answer correctness. A useful step may instead improve the model's answer belief, acquire a missing piece of task-critical evidence, or select a more appropriate region for the next observation. This motivates the structured credit assignment in CAVE, which decomposes intermediate progress into belief update, evidence acquisition, and adaptive focus control.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The claims in the Abstract and Section 1 are limited to the problem setting studied in this paper, the proposed CAVE framework, the construction of TRACER-Bench, and the empirical improvements observed in our experiments. These claims are supported by the method description in Section 3, the benchmark construction in Section 4, and the experimental results, ablations, and diagnostic analyses in Section 5.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We provide limitation discussions in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not present formal theoretical results such as theorems or lemmas.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides the method definition in Section 3, dataset construction and split protocols in Section 4 and Appendix C.2, and the evaluation setup and training details in Section 5 and Appendix D.1.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The complete code and reproduction package are not yet publicly available. We plan to release the implementation and supporting materials following publication, subject to applicable licensing and data-access restrictions.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper provides the experimental setup, benchmarks, evaluation protocol, and training details in Section 5, with additional data split and isolation details in Appendix C.2 and training/reward hyperparameters in Appendix D.1. The method and GRPO-based optimization objective are described in Section 3.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Appendix F reports an aggregate statistical significance analysis.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: We provide the information about compute resources in the Appendix D.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses synthetic, public, and benchmark-derived visual reasoning data. We have reviewed the NeurIPS Code of Ethics and are not aware of any violation.

   Guidelines:

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

10. **Broader impacts**

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

Answer: [Yes]

Justification: We provide discussions of broader impacts in Appendix H.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [N/A]

Justification: No high-risk models or datasets are currently released as part of this work. Therefore, safeguards for their public release are not applicable.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Existing assets are credited in the paper and summarized in Appendix I.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: This paper introduces TRACER-Bench and documents its construction, statistics, quality control, and data isolation protocols in the paper and appendices. The benchmark data and implementation are not yet publicly released, with release planned following publication.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing experiments or research with human subjects. Human performance is used only as a reference upper-bound baseline in Section 5, rather than as a newly conducted human-subject study.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects. Therefore, IRB approval or equivalent review is not applicable.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: The core method does not use an external LLM as an important, original, or non-standard component for data generation, annotation, evaluation, or analysis.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.