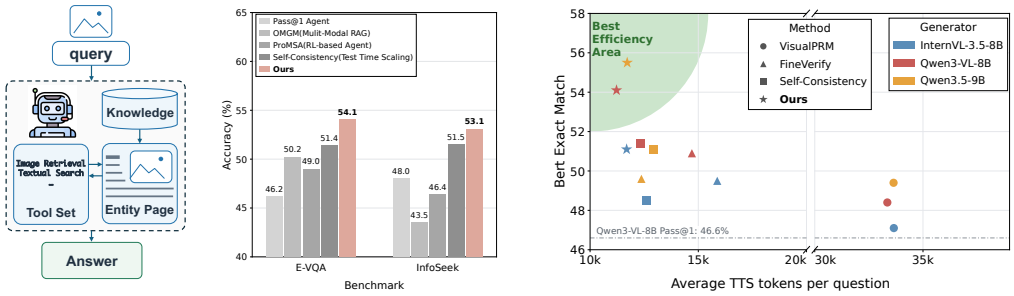
DiVE: Test-Time Scaling of Direct Visual Exploration via Cross-Trajectory Verification

Anonymous authors
Paper under double-blind review



(a) (b) (c)

Figure 1: KB-VQA overview and performance overview. (a) A KB-VQA pipeline. (b) Performance on E-VQA and InfoSeek. (c) Accuracy versus average verifier-stage token usage on E-VQA, excluding rollout generation. **CTV achieves higher accuracy with fewer verifier-stage tokens than the evaluated TTS selectors on shared rollout pools.**

Abstract

Knowledge-based visual question answering (KB-VQA) requires identifying the entity depicted in an image and retrieving facts about that entity. However, visually similar long-tail entities can confound grounding. We observe a failure pattern, termed a *grounding trap*: an agent may retrieve genuine facts about a visually similar but incorrect entity, yielding an internally coherent yet incorrect trajectory. To address this failure, we propose DiVE, a test-time scaling framework that enables direct visual exploration. DiVE combines fine-grained evidence access with direct interaction tools. At inference time, independent rollouts explore alternative entity hypotheses. An off-the-shelf verifier compares the recorded candidate entities and supporting evidence across these trajectories to select an existing answer. Compared with RAG, agentic-search, and alternative test-time scaling baselines, DiVE achieves state-of-the-art performance on E-VQA and InfoSeek. On shared rollout pools, CTV achieves higher accuracy with fewer verifier-stage tokens than the TTS baselines, indicating better verifier-stage efficiency. Our code is available anonymously.[1]

1 Introduction

In entity-centric knowledge-based visual question answering (KB-VQA), answering a question requires linking the depicted entity to facts in external knowledge (Caffagni et al., 2024). Knowing that an image shows a bridge is insufficient to answer its length; the system must identify *which bridge* it is and retrieve the corresponding fact. A reliable answer therefore depends on both identifying the visual entity and checking the retrieved evidence. Multimodal retrieval-augmented generation and agentic search improve access to external knowledge and reasoning ability (Yang et al., 2025; Yan & Xie, 2024; Wu et al., 2026b; Hong et al., 2026a; Wu et al., 2026a), but factual support for an answer does not necessarily establish the match between retrieved entity and the image.

[1] https://anonymous.4open.science/r/DivE-88CC

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

This dependence creates a subtle failure mode (Ma et al., 2026b): **visually similar entities can lead to competing interpretations of the same image.** As Figure 2 illustrates, confusing *Ram Jhula* with *Lakshman Jhula* can lead to a genuine bridge-length statement that nevertheless answers the question about the wrong bridge. We refer to this observed failure pattern as a ***grounding trap***: **the retrieved evidence supports the answer under an incorrect entity hypothesis, while the visual identity remains insufficiently checked.** Further reasoning may validate the fact without revisiting the visual identity. This coupling of ambiguous visual grounding and entity-specific knowledge makes factual support alone insufficient. An exploratory post-hoc audit provides direct evidence of this pattern: among 2,251 incorrect answers from a search agent on E-VQA, a GPT-5.6 Sol auditor flags 556 (24.70%) with explicit wrong-entity commitment and same-entity tool evidence supporting the queried attribute (Appendix B.5).

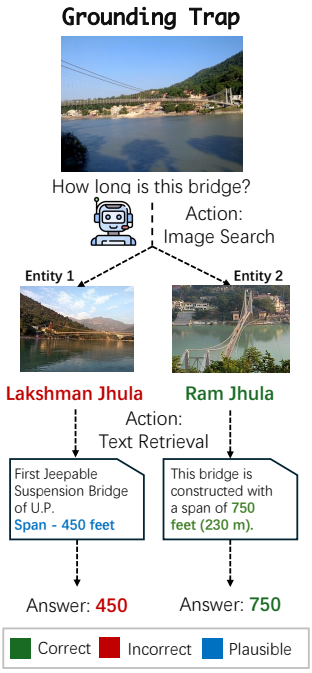


Figure 2: Visually similar bridges lead to pages with genuine length facts; **only the correctly grounded page answers the question.**

Test-time scaling (TTS) allocates additional inference compute to generate and evaluate multiple candidate trajectories (Snell et al., 2024). For KB-VQA, it offers clear coverage headroom: on E-VQA, Oracle@8 (72.61%) exceeds Pass@1 (46.2%). Yet repeated sampling does not guarantee entity diversity. Reflection can improve trajectory-level search (Zhou et al., 2024), but a wrong-entity trajectory may remain locally coherent because its facts are genuine. Selectors based on behavioral scoring, question decomposition, or answer consistency are not designed to explicitly preserve entity–evidence provenance. The challenge is therefore twofold: (1) exploration must reach genuinely different long-tail, visually similar entities rather than repeatedly visit the same plausible error; (2) verification must compare long and heterogeneous trajectories while retaining which evidence supports which entity. Factuality checks and majority voting alone are insufficient, while naively retaining every observation makes context grow with the rollout count. Effective TTS thus requires both *candidate coverage* and *conditional selection*.

In this work, we propose **Di**rect **V**isual **E**xploration (DiVE), a TTS framework for agentic KB-VQA. DiVE broadens candidate coverage by exploring alternative visual entity hypotheses, while CTV organizes their recorded evidence as an *Entity-Evidence Forest* for provenance-aware, token-efficient selection. Together, the two components address the complementary requirements of finding a correct candidate and selecting it reliably.

Experiments on E-VQA and InfoSeek show that DiVE-CTV achieves better performance than the evaluated RAG, agentic-search, and TTS selection baselines. DiVE does not require a fine-tuned retriever or reinforcement learning. As shown in Figure 1c, **CTV uses fewer verifier-stage tokens than other TTS baselines while achieving higher accuracy on the same candidate pool**. Our contributions are threefold:

- We point out a grounding-trap observation in entity-centric KB-VQA: retrieved facts can plausibly support an answer about the wrong entity, motivating comparison of alternative hypotheses.
- We propose DiVE, a test-time scaling framework combining direct visual exploration with CTV. Independent rollouts explore alternative entity hypotheses, while an Entity-Evidence Forest organizes their recorded evidence for joint answer selection.
- Experiments on E-VQA and InfoSeek demonstrate improvements over the evaluated RAG, agentic-search, and TTS baselines. Across four backbones, CTV outperforms the compared TTS selectors on shared E-VQA candidate pools while using fewer verifier-stage tokens.

2 Related Work

From retrieval to agentic KB–VQA. KB–VQA requires identifying the depicted entity before using external facts about it. Multimodal RAG retrieves relevant articles and passages through hierarchical, visual, or coarse-to-fine search (Caffagni et al., 2024; Yan & Xie, 2024; Yang et al., 2025). Retrieval narrows a large knowledge base, but a visually similar result need not depict the same entity; facts from its page can still support a plausible wrong answer. Agentic systems extend retrieval with iterative image and text search, cropping, and tool use (Wu et al., 2026a; Chng et al., 2025; Wu et al., 2026b), while direct corpus interaction exposes context beyond retrieved snippets (Li et al., 2026). DiVE combines indexed candidate discovery, visual inspection of candidate entities, and local access to their pages to keep visual identification distinct from factual lookup.

Test-time scaling and visual exploration. Test-time scaling (TTS) spends additional inference compute on generating or evaluating candidates (Snell et al., 2024). Self-consistency votes over final answers (Wang et al., 2023), whereas VisualPRM scores reasoning steps and FineVerify checks candidate answers against decomposed questions and retrieved evidence (Wang et al., 2025b; Zhao et al., 2026). ReflectiVA and MACT further combine retrieval or collaboration with reflective decision-making (Cocchi et al., 2025; Yu et al., 2025b). In entity-centric KB–VQA, independently sampled trajectories may visit different entities and collect different facts, so agreement between final answers can obscure their evidence provenance. CTV instead groups completed trajectories by entity and compares their recorded entity and evidence paths to select an existing answer. TTSP instead uses entropy-gated perceptual exploration and an Evidence Ledger to guide later visual trajectories (Jiang et al., 2026). Further distinctions and related active-perception work appear in Appendix G.

3 Problem Formulation

Following prior work (Yan & Xie, 2024; Cocchi et al., 2025; Yang et al., 2025), we consider an encyclopedic setting with an external knowledge base of Wikipedia entity articles and their associated images. Let $x = (I, q)$ denote an input image–question pair, and define the knowledge base as

$$\mathcal{K} = \{(e_j, d_j, \mathcal{I}_j)\}_{j=1}^{M}, \tag{1}$$

where M is the number of entries, e_j is an entity identifier, d_j is its textual article, and $\mathcal{I}_j$ is the set of associated reference images. The task is to give an answer $\hat{a} = f_\theta(x; \mathcal{K})$, where f_θ denotes a question-answering system with parameters θ and access to external knowledge $\mathcal{K}$. We instantiate f_θ as an agent with policy π_θ and a tool set $\mathcal{T}$, following the agentic-search paradigm (Yao et al., 2023; Hu et al., 2023). At step t, the policy conditions on the input x and the interaction history h_t to generate intermediate reasoning r_t and an action u_t. The action either invokes a tool $g_t \in \mathcal{T}$ with arguments ξ_t, written as $u_t = (g_t, \xi_t)$, or terminates the interaction with a final answer. Formally,

$$(r_t, u_t) \sim \pi_\theta(\cdot \mid x, h_t, \mathcal{T}), \quad o_t = g_t(\xi_t; I, \mathcal{K}), \quad \text{if } u_t = (g_t, \xi_t). \tag{2}$$

For a tool invocation, the returned observation o_t may contain candidate entities, reference images, or textual evidence. The history is then updated as $h_{t+1} = h_t \oplus (r_t, u_t, o_t)$, where $h_1 = \varnothing$ and $\oplus$ denotes sequence concatenation.

Test-time Scaling. Test-time scaling (TTS) allocates additional inference computation to improve answer prediction (Snell et al., 2024). In our setting, N independent instances of the search agent π_θ for the same input x are launched, each with access to $\mathcal{K}$ and $\mathcal{T}$ under a common per-rollout resource budget B. The instances share π_θ but maintain separate interaction histories and independently sample their reasoning and actions. This produces N trajectories:

$$\begin{aligned} &\tau_i \overset{\text{i.i.d.}}{\sim} p_{\pi_\theta}(\tau \mid x, \mathcal{K}, \mathcal{T}; B), \\ &\tau_i = \big(h_{i,T_i+1}, a_i\big) = \big((r_{i,t}, u_{i,t}, o_{i,t})_{t=1}^{T_i}, a_i\big), \qquad i = 1, \ldots, N, \end{aligned} \tag{3}$$

where p_{π_θ} is the trajectory distribution induced by the agent and its tool interactions, T_i is the number of tool calls in the i-th trajectory, h_{i,T_i+1} is its accumulated interaction history before termination, and a_i is its terminal answer. Given the collection $\mathcal{R}_N = (\tau_1, \ldots, \tau_N)$, a verifier $\mathcal{V}$ jointly examines the trajectories and selects the one it judges most reliable for answering the question. Formally,

$$\hat{\imath} = \mathcal{V}(x, \mathcal{R}_N) \in \{1, \ldots, N\}, \qquad (\hat{\tau}, \hat{a}) = (\tau_{\hat{\imath}}, a_{\hat{\imath}}). \tag{4}$$

The final output consists of the selected trajectory $\hat{\tau}$ and its associated answer $\hat{a}$.

Grounding Traps. Let $e^\star$ be the entity depicted in I and $\tilde{e}$ the entity selected by the agent, with respective knowledge $\mathcal{K}_{e^\star}$ and $\mathcal{K}_{\tilde{e}}$. For a schematic description, we idealize factual reasoning under either hypothesis as accurate, with correct answer $a^\star = f_\theta(x; \mathcal{K}_{e^\star})$. Although different entities may share an answer, the observed pattern concerns trajectories that remain committed to an incorrect entity $\tilde{e}$ and return an evidence-supported but incorrect answer satisfying:

$$\tilde{e} \neq e^\star, \qquad \tilde{a} = f_\theta(x; \mathcal{K}_{\tilde{e}}), \qquad \tilde{a} \neq a^\star. \tag{5}$$

Genuine facts about $\tilde{e}$ can support an internally consistent but incorrect answer without verifying its identity in the image. Further search under the same hypothesis may strengthen factual support without resolving visual ambiguity, motivating comparison of alternative entities. Appendix B gives an illustrative analysis under explicit assumptions. An exploratory GPT-5.6 Sol audit of the query image and full trajectory flags 556 of 2,251 errors (24.70%), requiring explicit wrong-entity commitment and verbatim tool evidence from that entity supporting the queried attribute.

4 DiVE: Test-Time Scaling of Direct Visual Exploration

We propose **Di**rect **V**isual **E**xploration (DiVE), a framework that generates and jointly verifies multiple evidence-seeking trajectories for KB-VQA (Figure 3). Its search agent combines visual candidate discovery with entity-scoped evidence inspection. We equip the agent with direct-interaction tools and use standard SFT on teacher-generated trajectories. At inference time, multiple trajectories start with separate histories, allowing the agent to reconsider the initial entity hypothesis across rollouts under a common per-rollout budget. Cross-Trajectory Verification (CTV) groups trajectories by answer-grounding entity and compares their candidate names and recorded evidence to select one existing answer.

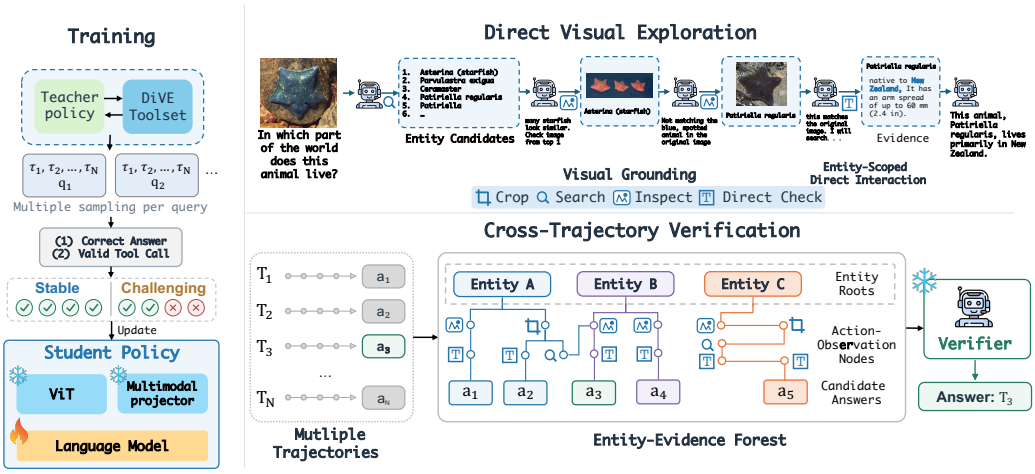


Figure 3: Overview of DiVE framework. **Left**: the agent is trained by SFT on rejection-sampled teacher trajectories. **Top right**: DiVE explores candidate entities through visual tools and inspects their knowledge pages. **Bottom right**: CTV groups the completed trajectories by entity and compares their records to select an existing answer.

4.1 Direct Visual Exploration

Indexed retrieval supports efficient candidate discovery, while direct corpus interaction provides access to exact evidence in the raw corpus. We combine these interfaces to search the entire knowledge base both efficiently and accurately. **Visual grounding** first proposes candidate entities through indexed search, and **entity-scoped direct interaction** inspects their associated pages. This design narrows the scope of text inspection while giving the agent fine-grained control over the evidence it reads. DiVE explores a wider range of entity hypotheses through visual grounding and efficiently obtains direct evidence through entity-scoped interaction. Experiments in Section 5.4 show that, across multiple rollouts, **DiVE encounters 1.59× as many visible entities, proposes 1.65× as**

many image candidates, and inspects 1.79× as many entities as its retrieval-based counterpart.

Visual Grounding. At this stage, the agent proposes entity hypotheses by examining the query and candidate images. The agent uses three complementary tools: (1) *search*, which retrieves candidate entities from the original image or a crop; (2) *crop*, which isolates question-relevant regions; and (3) *compare*, which loads candidate reference images for comparison with the query image and its crops. It can repeat these operations to distinguish visually similar candidates and establish the entity hypotheses used for subsequent text inspection.

Entity-Scoped Direct Interaction. Given the current entity hypotheses, the agent restricts text search in the corresponding entity pages to find evidence answering the query. It issues queries for question-relevant attributes and relations, using regular expressions when appropriate. The agent can reformulate a query, adjust the context window, or inspect the same attribute across several candidate pages. Visual grounding therefore determines *where to look*, while direct interaction determines *what the selected pages support*. By combining visual grounding with direct interaction, we can precisely access local evidence without requiring a corpus-wide text search.

Visual grounding and text inspection share an evolving interaction history and are not restricted to a fixed order. After each tool call, the agent incorporates the returned observation as described in Equation 2. If the available evidence suggests an identity mismatch, it can revisit visual grounding and revise the candidate entity set. The agent continues this interaction until it produces a final answer within the per-rollout budget. Tool arguments, output limits, and the supplementary semantic-retrieval fallback are described in Appendix C.2.

Fine-tuning DiVE. To teach DiVE valid tool calls through supervised fine-tuning, a teacher independently generates four trajectories per training query. We evaluate the final answers and categorize queries as *stable* if all four rollouts are correct, or *challenging* if only one to three are correct. Queries with no correct rollout are excluded. From both groups, rejection sampling retains one answer-correct trajectory with valid tool-call formats and successful executions per selected query. These retained query–trajectory pairs define the empirical training distribution p_{SFT}. We optimize the student policy with the standard SFT objective over the resulting interaction trajectories:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\mathbb{E}_{(x,\tau)\sim p_{\mathrm{SFT}}}\left[\sum_{\ell\in\mathcal{J}_{\mathrm{gen}}(\tau)} \log \pi_\theta(y_\ell \mid x, y_{<\ell}, \mathcal{T})\right], \tag{6}$$

where y_ℓ is the token at position ℓ in the serialized interaction, and $y_{<\ell}$ is its preceding multimodal context, including tool observations. We freeze the visual encoder and multimodal projector during training and update only the language-model parameters. Full data-construction, filtering, and optimization details are provided in Appendices D and F.2.

4.2 Cross-Trajectory Verification

DiVE produces diverse evidence-seeking trajectories, but an individual rollout can remain committed to a visually plausible yet incorrect entity despite accumulating factual support (Appendix B.2–B.3). Visual grounding and entity exploration occur during DiVE rollouts; **Cross-Trajectory Verification** (CTV) selects among completed trajectories using their serialized records. CTV organizes the N trajectories into an entity-evidence forest according to their answer-bearing entity and shared observations, and an off-the-shelf LLM then ranks the existing trajectories. CTV proceeds in three stages, summarized in Algorithm 1: entity assignment, forest construction, and cross-trajectory selection. Experiments in Section 5.4 show that **CTV improves both entity selection and answer selection given the entity**, yielding the highest TTS gain (Figure 6).

Entity assignment. For each trajectory τ_i, CTV extracts normalized entity-title sets from entity-bearing tool calls and collapses consecutive duplicates into the route $\mathcal{E}_i = (E_{i,1}, \ldots, E_{i,K_i})$. Let m_i contain entity titles encountered along τ_i that occur as complete phrases in the frozen answer a_i. Let E_i^{fb} be the last text-search entity state, falling back to the last visual entity state, or $\varnothing$ if neither exists. CTV assigns the terminal entity state as

$$\hat{E}_i = \eta(\tau_i) = \begin{cases} m_i, & m_i \neq \varnothing, \\ E_i^{\mathrm{fb}}, & m_i = \varnothing \ \wedge \ E_i^{\mathrm{fb}} \neq \varnothing, \\ \perp, & m_i = \varnothing \ \wedge \ E_i^{\mathrm{fb}} = \varnothing. \end{cases} \tag{7}$$

This prioritizes answer mentions, then text-search entities, then visual entity states; the full route retains entity switches. CTV entity-assignment rule assigns a root to nearly all trajectories; see Appendix H.5 for the detailed results.

Forest construction. CTV places trajectories with the same terminal entity state under a shared root. Using the trajectory collection $\mathcal{R}_N$, the root-level organization of the forest is

$$\mathcal{F}_N = \left\{ \left(E, \{\tau_i \in \mathcal{R}_N \mid \hat{E}_i = E\} \right) \;\middle|\; E \in \{\hat{E}_i : i = 1, \ldots, N\} \right\}. \tag{8}$$

Within each root, CTV represents the recorded tool calls as *navigation nodes* for image search and cropping, *grounding nodes* for reference-image inspection, or *evidence nodes* for entity-scoped text search. Navigation and evidence observations are shared only when their node type, normalized target, return status, and normalized raw return match. Grounding calls with the same entity and status share one node, while their recorded textual returns are retained as variants. Each trajectory preserves its call order, entity route, return statuses, and unchanged answer a_i. The forest is therefore a compact reorganization of $\mathcal{R}_N$ and performs no tree search, value backup, or rollout expansion. Grouping trajectories by entity makes competing hypotheses explicit, while sharing observations reduces redundant context, aiming to focus the verifier's attention on evidence differences across entities.

Cross-trajectory selection. CTV serializes the entity-evidence forest into the fixed format and presents it to an off-the-shelf LLM together with the question and candidate entity names. The verifier compares each branch's named entity, recorded grounding steps, factual evidence, and answer. It returns the valid non-empty trajectories in preference order; CTV selects the first trajectory as $\hat{\imath}$ in Equation 4 and copies its answer without rewriting it. CTV can therefore recover from a locally coherent wrong-entity trajectory only when another rollout already contains a better-supported answer. Appendix B.4 formalizes the roles of exploration and verification by decomposing final accuracy into candidate coverage and conditional selection success (Equation 17). The complete algorithm and an entity-evidence forest example are provided in Appendix C.3.

5 Experiments

5.1 Datasets

Following prior work, we test DiVE on two challenging KB-VQA datasets, E-VQA (Mensink et al., 2023) and InfoSeek (Chen et al., 2023b). For a fair comparison, we adopt the evaluation setup and knowledge bases used in previous studies (Yan & Xie, 2024; Yang et al., 2025). We use BERT-based Matching (BEM) (Bulian et al., 2022) for E-VQA and both VQA accuracy and relaxed accuracy for InfoSeek. Further details about the benchmarks and their knowledge bases are provided in Appendices E and F.

5.2 Implementation Details

Baselines. We compare with three groups of baselines in Table 1: (i) *multimodal RAG*: EchoSight (Yan & Xie, 2024), OMGM (Yang et al., 2025), VLM-PRF (Hong et al., 2025), REAL (Ye et al., 2026), MaS-VQA (Mao et al., 2026), and CC-VQA (Hong et al., 2026b); (ii) *entity-grounding and agentic search*: IBA-Qwen (Ma et al., 2026b), MMSearch-R1 (Wu et al., 2026a), DeepEyesV2 (Hong et al., 2026a), and ProMSA (Wu et al., 2026b); and (iii) *TTS*: VisualPRM (Wang et al., 2025b), an action-level reward baseline; FineVerify (Zhao et al., 2026), a question-decomposition-based verification baseline; and Self-consistency (Wang et al., 2023), a listwise baseline based on semantic agreement. Baseline details are provided in Appendix F.5.

DiVE Training Recipe. We train DiVE on Qwen3-VL-2B/8B, Qwen3.5-9B and InternVL-3.5-8B using SFT data described in Section 4.1. We fine-tune on 2,000 selected teacher-generated interaction trajectories, evenly divided between the E-VQA and InfoSeek training sets. Teacher agent Qwen3-VL-235B generates four complete rollouts for each training query, after which answers are scored and structurally invalid trajectories are removed. We train for three epochs with updates to the language model while freezing the visual encoder and multimodal projector.

Test-time trajectory scaling. For each evaluation query, we independently sample $N = 8$ complete DiVE trajectories and use the same frozen candidate pool for CTV and all TTS baselines. CTV uses off-the-shelf Qwen3-VL-8B as its listwise verifier.

Table 1: Performance comparison on E-VQA and InfoSeek. All denotes the average of the E-VQA and overall InfoSeek scores. * indicates our implementation using officially released code. ○ = off-the-shelf, ◐ = SFT, and ● = RL. All TTS rows use the same eight frozen DiVE rollouts.

Method	Generator	Ret. PT	Gen. PT	E-VQA	InfoSeek		All
					Un-Q	Un-E	
Multi-Modal RAG							
EchoSight (Yan & Xie, 2024)	Qwen3-VL-8B	◐	○	38.1	32.6	32.2	35.3
OMGM (Yang et al., 2025)	LLaVA-1.5-7B	◐	◐	50.2	43.5	43.5	46.9
VLM-PRF (Hong et al., 2025)	InternVL3-8B	●	○	40.1	43.3	42.1	41.3
REAL (Ye et al., 2026)	Qwen3-VL-8B	◐	◐	45.5	43.1	45.1	44.8
MaS-VQA (Mao et al., 2026)	Qwen3-VL-8B	◐	○	42.2	43.7	43.9	43.0
CC-VQA (Hong et al., 2026b)	Qwen2.5-VL-7B	◐	○	41.4	44.7	46.1	43.3
Agentic Search Frameworks							
IBA-Qwen (Ma et al., 2026b)	Qwen-2.5-VL	○	○	43.6	37.7	37.8	40.7
MMSearch-R1 (Wu et al., 2026a)	Qwen2.5-VL-7B	○	●	40.6	40.3	39.3	40.2
DeepEyesV2 (Hong et al., 2026a)	Qwen2.5-VL-7B	○	●	40.1	41.6	40.8	40.7
ProMSA* (Wu et al., 2026b)	Qwen3-VL-8B	○	●	49.0	47.3	46.0	47.7
Test-time Scaling Agentic Frameworks							
VisualPRM (Wang et al., 2025b)	Qwen3-VL-2B	○	◐	45.1	45.4	43.8	44.7
	InternVL-3.5-8B	○	◐	47.1	44.9	43.4	45.5
	Qwen3-VL-8B	○	◐	48.4	49.8	47.9	48.4
	Qwen3.5-9B	○	◐	49.4	48.5	46.9	48.4
FineVerify (Zhao et al., 2026)	Qwen3-VL-2B	○	◐	45.5	46.4	45.2	45.5
	InternVL-3.5-8B	○	◐	49.5	46.8	45.8	47.8
	Qwen3-VL-8B	○	◐	50.9	51.5	50.5	50.9
	Qwen3.5-9B	○	◐	49.6	51.0	49.4	49.7
Self-consistency (Wang et al., 2023)	Qwen3-VL-2B	○	◐	47.5	47.6	45.7	46.9
	InternVL-3.5-8B	○	◐	48.5	47.3	45.8	47.4
	Qwen3-VL-8B	○	◐	51.4	52.2	51.2	51.5
	Qwen3.5-9B	○	◐	51.1	51.1	49.9	50.7
DiVE-CTV (Ours)	Qwen3-VL-2B	○	◐	50.0	49.8	48.3	49.3
	InternVL-3.5-8B	○	◐	51.1	48.6	47.4	49.4
	Qwen3-VL-8B	○	◐	<u>54.1</u>	**53.8**	**52.9**	<u>53.6</u>
	Qwen3.5-9B	○	◐	**55.5**	<u>52.9</u>	<u>51.9</u>	**53.9**

Training details, sampling settings, and baseline procedures are provided in Appendix F.

5.3 Experimental Results

DiVE achieves state-of-the-art performance. Table 1 reports the main results on two KB-VQA benchmarks. Compared with multimodal RAG methods, DiVE consistently achieves better performance without fine-tuning the encoder or reranker, relying primarily on the agent's reasoning and tool-use capabilities. DiVE also demonstrates substantial improvements over existing agentic-search methods. With only lightweight fine-tuning on 2,000 examples and test-time scaling, it achieves performance comparable to that of agents post-trained with reinforcement learning. On the same frozen rollout pool, CTV outperforms all three types of TTS baselines: process-reward scoring, question decomposition, and consistency-based listwise selection. On E-VQA across all four backbones, CTV's BEM gains over all three TTS baselines are significant within each shared frozen pool (exact McNemar $p < 0.05$ after Holm correction over all 12 comparisons; Appendix H.1).

CTV's advantage becomes more pronounced in more challenging scenarios. Figure 4 groups questions by the number of correct candidate trajectories among eight rollouts; CTV achieves the highest selection accuracy across all difficulty levels. When correct paths are in the minority and easily overwhelmed by incorrect ones, CTV's structured comparison improves verifier performance. The smaller gaps in the 4–8/8-pass subset are consistent with easier selection when at least half of the candidate trajectories already answer correctly.

5.4 Controlled Ablations and Mechanism Analysis

We evaluate all eight combinations of Direct Tool, SFT, and CTV. *Direct Tool off* replaces the direct-access interface with BGE-based textual retrieval, returning the top five sections from the same knowledge base. *SFT off* uses the original Qwen3-VL-8B without fine-tuning. *CTV off* reports single-trajectory accuracy. Table 2 shows complementary benefits: enabling any component improves performance, and every two-component combination outperforms all single-component

Table 2: Ablation study of Toolset, SFT, and TTS on E-VQA. ● and ○ indicate enabled and disabled modules, respectively. ◐ denotes replacing CTV with a flat listwise verifier. Δ is the E-VQA score minus the full-model score.

Direct Tool	SFT	CTV	Acc.	Δ
○	○	○	34.9	-19.2
●	○	○	40.6	-13.5
○	●	○	38.8	-15.3
○	○	●	39.5	-14.6
○	●	●	43.8	-10.3
●	●	○	46.2	-7.9
●	○	●	49.5	-4.6
●	●	◐	50.2	-3.9
●	●	●	54.1	

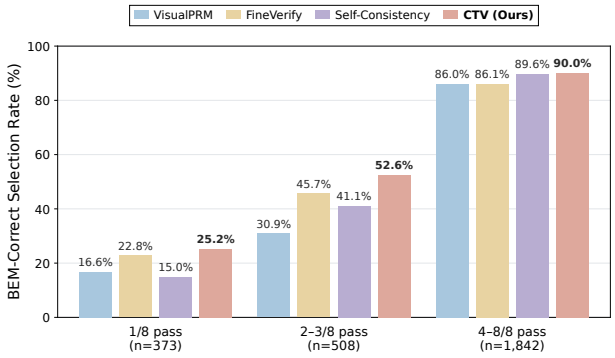


Figure 4: Effect of TTS across subsets. Fewer correct trajectories among eight rollouts indicate more challenging queries. **CTV consistently outperforms the other methods, especially on the challenging subsets.**

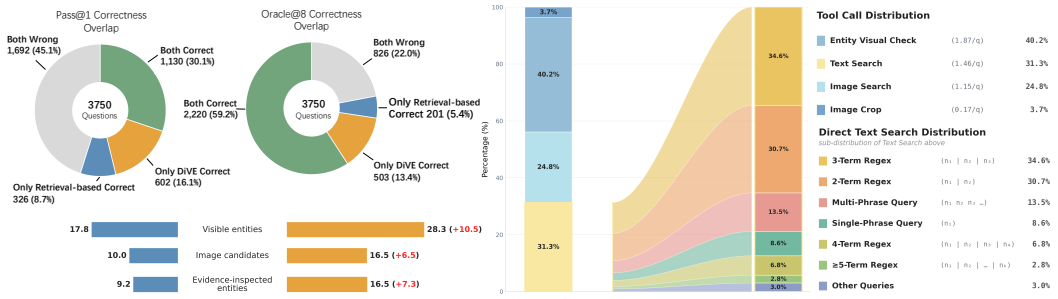


Figure 5: DiVE Tool Analysis: Answer Coverage, Entity Exploration, and Tool Usage. **Left**: correctness overlap and entity exploration statistics for DiVE and its retrieval-based counterpart on E-VQA. **Right**: DiVE's tool-call distribution and the breakdown of direct textual search. **DiVE explores more entities through visual and direct search tools and improves performance.**

variants. Among the two-component configurations, Direct Tool with CTV is closest to the full model: it reaches 49.5% accuracy, compared with 54.1% for full DiVE and 34.9% for the all-disabled baseline. This 14.6-point gain shows that an off-the-shelf MLLM can benefit substantially from the framework. Replacing CTV with a flat listwise verifier yields 50.2% accuracy on E-VQA, versus 54.1% for CTV, a gap of 3.9 percentage points (Table 2). We next examine how Direct Tool broadens entity exploration and how CTV selects among the resulting trajectories.

5.4.1 Why Does DiVE Work?

Direct interaction supports broader entity exploration within a trajectory and greater coverage across multiple rollouts, providing more candidate hypotheses and inspected evidence for cross-trajectory verification. We compare DiVE with its retrieval-based counterpart on E-VQA at Pass@1 and Oracle@8. Figure 5 (left) shows that, at Pass@1, 602 questions are answered correctly only by DiVE, whereas 326 are answered correctly only by the retrieval-based agent. At Oracle@8, the corresponding counts are 503 and 201, respectively. DiVE also encounters more visible entities, image candidates, and evidence-inspected entities across the interaction histories at both sampling budgets. In particular, the number of evidence-inspected entities increases from 2.5 to 4.5 at Pass@1 and from 9.2 to 16.5 at Oracle@8. The gaps in visible entities and image candidates also widen with eight rollouts. More fine-grained tool ablation results are reported in Appendix H.2.

5.4.2 Why Does Cross-Trajectory Verification Work?

CTV improves entity and answer selection. To understand where this advantage comes from, we compare all TTS verifiers on the oracle-hit set, i.e., the questions whose frozen pool contains at least one correct rollout. Figure 6 compares the verifiers across five key axes: token efficiency, entity

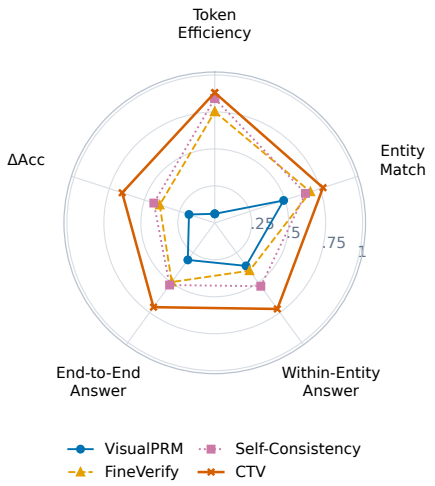


Figure 6: Five-axis diagnostics of TTS methods using Qwen3-VL-8B rollouts on the E-VQA oracle-hit set. The metric definitions and the fixed radial ranges are given in Appendix H. ΔAcc uses the corrected 46.2% Pass@1 baseline.

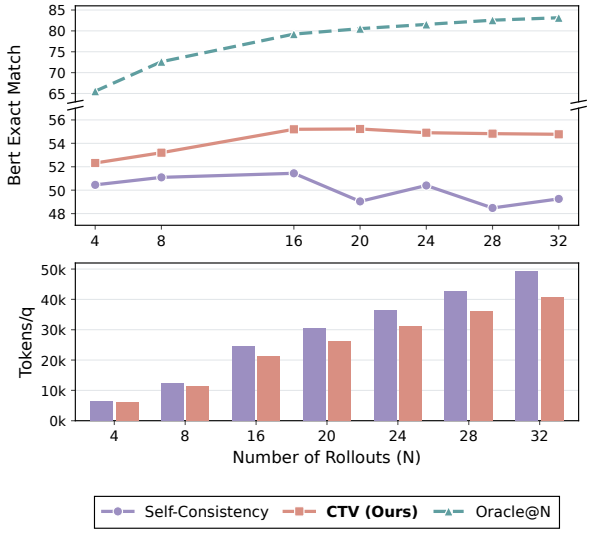


Figure 7: Scaling dynamics of CTV and Self-Consistency. CTV achieves higher accuracy with fewer verifier-stage tokens in the evaluated implementations. The two methods share the same rollout generations.

match (correct entity selection), within-entity answer rate (path accuracy conditioned on correct entity), end-to-end answer rate on the oracle-hit set, and the accuracy gain over the Pass@1 result (ΔAcc). CTV performs best on all five axes, selecting trajectories associated with the gold entity and correct answers within that entity more often, at the lowest reported verifier cost. The forest groups trajectories by terminal entity state and reuses matching observation nodes within each group to reduce duplicated input context. Detailed discussion of these axes, tabular results, and visualizations for other baselines is provided in Appendix H.

Scaling Dynamics. Figure 7 compares E-VQA BEM scores and verifier-stage token consumption for CTV and self-consistency across candidate budgets $N \in \{4, 8, 16, 20, 24, 28, 32\}$. Within each backbone and fixed N, all selectors reuse the same frozen rollout pool, making their generation token cost identical. We report verifier-stage input plus output tokens to isolate the additional token cost of selection (Appendix H). CTV has a better accuracy–verifier-token trade-off than the evaluated self-consistency implementation at the tested budgets; this does not establish an end-to-end compute Pareto frontier. Comparisons across additional base models and test-time scaling (TTS) methods are provided in Figure 1c of the Introduction. Furthermore, the results reveal that although Oracle@N scales steadily up to N=32, the gains of CTV do not fully keep pace with this trajectory.

5.5 Generalization and Supplementary Analyses

Without fine-tuning on task-specific data, we also evaluate DiVE on **ViQuAE** (Lerner et al., 2022), **M3-VQA** (Ma et al., 2026a), and **MC-SEARCH** (Ning et al., 2026b). DiVE-CTV improves over the corresponding Pass@1 agent on all three; Appendix H.4 reports the prompt and output-format adaptations and the remaining Oracle@8 gaps. Qualitative cases appear in Appendix I.

6 Conclusion

We presented DiVE, a test-time scaling framework for entity-centric KB-VQA. DiVE addresses grounding traps, in which genuine facts about an incorrect entity support a coherent but wrong answer, by combining direct visual exploration, independent rollouts, and Cross-Trajectory Verification over recorded entity–evidence paths. On E-VQA and InfoSeek, DiVE outperforms the evaluated retrieval-augmented, agentic-search, and test-time selection baselines. These results show that effective scaling requires both diverse candidates and provenance-aware comparison.

AI USE STATEMENT

We used AI to assist with manuscript drafting and revision, literature search, LaTeX editing, and visualization of experimental results. As part of the research workflow, Qwen3-VL-235B-A22B-Instruct generated candidate training trajectories (Appendix D), and GPT-5.6 Sol performed the exploratory grounding-trap audit using query images, full trajectories, and gold references (Appendix B.5). The model-based generation and verification procedures used in DiVE are described in Section 4. The authors reviewed all AI-assisted work and take full responsibility for the final content, including the reported methods, results, claims, and references.

ETHICS STATEMENT

To the best of our knowledge, this work raises no material ethical concerns. It uses existing KB-VQA benchmarks and does not involve human-subject research, collection of sensitive personal data, or deployment in safety-critical settings. The proposed method is intended for research on factual grounding and verification. As with other knowledge-based VQA systems, its outputs may inherit errors or biases from underlying models, tools, and knowledge sources and should not be treated as authoritative in high-stakes applications.

REPRODUCIBILITY STATEMENT

Our code is available in an anonymous repository.[2] Appendix C provides the agent instructions, tool interfaces, and CTV algorithm. Appendix D describes construction and filtering of the Balanced-2K training set, and Appendix E documents the evaluation benchmarks. Appendix F reports optimization settings, training hardware and runtimes, baseline details, and the shared eight-rollout protocol used for TTS comparisons within each backbone. Paired statistical comparisons under the main-table BEM protocol are reported in Appendix H.1.

REFERENCES

Omar Adjali, Olivier Ferret, Sahar Ghannay, and Hervé Le Borgne. Multi-level information retrieval augmented generation for knowledge-based visual question answering. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 16499–16513, 2024.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection, 2023. URL https://arxiv.org/abs/2310.11511.

Elham J. Barezi and Parisa Kordjamshidi. Disentangling knowledge-based and visual reasoning by question decomposition in KB-VQA, 2024. URL https://arxiv.org/abs/2406.18839.

Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022. doi: 10.18653/v1/2022.emnlp-main.20. URL https://aclanthology.org/2022.emnlp-main.20/.

Davide Caffagni, Federico Cocchi, Nicholas Moratelli, Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Wiki-llava: Hierarchical retrieval-augmented generation for multimodal llms. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1818–1826. IEEE, 2024.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*, 2023a.

[2] https://anonymous.4open.science/r/DivE-88CC/README.md

Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14948–14968. Association for Computational Linguistics, 2023b. doi: 10.18653/v1/2023.emnlp-main.925. URL https://aclanthology.org/2023.emnlp-main.925/.

Yong Xien Chng, Tao Hu, Wenwen Tong, Xueheng Li, Jiandong Chen, Haojia Yu, Jiefan Lu, Hewei Guo, Hanming Deng, Chengjun Xie, et al. Sensenova-mars: Empowering multimodal agentic reasoning and search via reinforcement learning. *arXiv preprint arXiv:2512.24330*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Federico Cocchi, Nicholas Moratelli, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Augmenting multimodal llms with self-reflective tokens for knowledge-based visual question answering. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9199–9209. IEEE, 2025.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. John Wiley & Sons, 2 edition, 2006. doi: 10.1002/047174882X. URL https://doi.org/10.1002/047174882X.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Shaohan Wang, Pengyu Wang, Xiaorui Wang, and Zhendong Mao. A-RAG: Scaling agentic retrieval-augmented generation via hierarchical retrieval interfaces, 2026. URL https://arxiv.org/abs/2602.03442.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models, 2025. URL https://arxiv.org/abs/2407.01449.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Nan Duan, and Weizhu Chen. Critic: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2305.11738.

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. In *International Conference on Learning Representations*, volume 2026, pp. 114851–114872, 2026a.

Yuyang Hong, Jiaqi Gu, Yang Qi, Lubin Fan, Yue Wu, Ying Wang, Kun Ding, Shiming Xiang, and Jieping Ye. Knowledge-based visual question answer with multimodal processing, retrieval and filtering. *Advances in Neural Information Processing Systems*, 38:119251–119282, 2025.

Yuyang Hong, Jiaqi Gu, Yujing Lou, Lubin Fan, Qi Yang, Ying Wang, Kun Ding, Yue Wu, Shiming Xiang, and Jieping Ye. Cc-vqa: Conflict-and correlation-aware method for mitigating knowledge conflict in knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5232–5241, 2026b.

Wenbo Hu, Jia-Chen Gu, Zi-Yi Dou, Mohsen Fayyaz, Pan Lu, Kai-Wei Chang, and Nanyun Peng. MRAG-Bench: Vision-centric evaluation for retrieval-augmented multimodal models, 2025. URL https://arxiv.org/abs/2410.08182.

Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David A. Ross, Cordelia Schmid, and Alireza Fathi. AVIS: Autonomous visual information seeking with large language model agent. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/029df12a9363313c3e41047844ecad94-Abstract-Conference.html.

Furong Jia, Ling Dai, Wenjin Deng, Fan Zhang, Chen Hu, Daxin Jiang, and Yu Liu. SpotAgent: Grounding visual geo-localization in large vision-language models through agentic reasoning, 2026. URL https://arxiv.org/abs/2602.09463.

Pu Jian, Donglei Yu, and Jiajun Zhang. Large language models know what is key visual entity: An LLM-assisted multimodal retrieval for VQA. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 10939–10956. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.613. URL https://aclanthology.org/2024.emnlp-main.613/.

Zheng Jiang, Yiming Chen, Nan He, Jiahui Chen, Chaoyang Li, Houde Qian, and Lifeng Sun. Test-time scaling over perception: Resolving the grounding paradox in thinking with images, 2026. URL https://arxiv.org/abs/2604.11025.

Zhengbao Jiang, Frank F. Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation, 2023. URL https://arxiv.org/abs/2305.06983.

Paul Lerner, Olivier Ferret, Camille Guinaudeau, Hervé Le Borgne, Romaric Besançon, José G Moreno, and Jesús Lovón Melgarejo. Viquae, a dataset for knowledge-based visual question answering about named entities. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, pp. 3108–3120, 2022.

Paul Lerner, Olivier Ferret, and Camille Guinaudeau. Cross-modal retrieval for knowledge-based visual question answering, 2024. URL https://arxiv.org/abs/2401.05736.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. Search-o1: Agentic search-enhanced large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 5420–5438. Association for Computational Linguistics, 2025a. doi: 10.18653/v1/2025.emnlp-main.276. URL https://aclanthology.org/2025.emnlp-main.276/.

Yangning Li, Yinghui Li, Xinyu Wang, Yong Jiang, Zhen Zhang, Xinran Zheng, Hui Wang, Hai-Tao Zheng, Philip S. Yu, Fei Huang, and Jingren Zhou. Benchmarking multimodal retrieval augmented generation with dynamic VQA dataset and self-adaptive planning agent, 2025b. URL https://arxiv.org/abs/2411.02937.

Zhuofeng Li, Haoxiang Zhang, Cong Wei, Pan Lu, Ping Nie, Yi Lu, Yuyang Bai, Shangbin Feng, Hangxiao Zhu, Ming Zhong, Yuyu Zhang, Jianwen Xie, Yejin Choi, James Zou, Jiawei Han, Wenhu Chen, Jimmy Lin, Dongfu Jiang, and Yu Zhang. Beyond semantic similarity: Rethinking retrieval for agentic search via direct corpus interaction, 2026. URL https://arxiv.org/abs/2605.05242.

Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2309.17133.

Weizhe Lin, Jingbiao Mei, Jinghong Chen, and Bill Byrne. PreFLMR: Scaling up fine-grained late-interaction multi-modal retrievers. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 5294–5316. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.289. URL https://aclanthology.org/2024.acl-long.289/.

Yi Lu, Zhuofeng Li, Ping Nie, Haoxiang Zhang, Yuyu Zhang, Kai Zou, Wenhu Chen, Jimmy Lin, Dongfu Jiang, and Yu Zhang. Dr-DCI: Scaling direct corpus interaction via dynamic workspace expansion, 2026. URL https://arxiv.org/abs/2606.14885.

Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 6417–6431. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.emnlp-main.517. URL https://aclanthology.org/2021.emnlp-main.517/.

Man Luo, Zhiyuan Fang, Tejas Gokhale, Yezhou Yang, and Chitta Baral. End-to-end knowledge retrieval with multi-modal queries. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 8573–8589. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.478. URL https://aclanthology.org/2023.acl-long.478/.

Jiatong Ma, Longteng Guo, Yuchen Liu, Zijia Zhao, Dongze Hao, Xuanxu Lin, and Jing Liu. M^3-VQA: A benchmark for multimodal, multi-entity, multi-hop visual question answering. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40633–40669, 2026a. doi: 10.18653/v1/2026.acl-long.1888. URL https://aclanthology.org/2026.acl-long.1888/.

Qian Ma, Qiong Wu, Zhengyi Zhou, and Yao Ma. Ground then rank: Revisiting knowledge-based vqa with training-free entity identification. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11617–11635, 2026b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Xianwei Mao, Kai Ye, Sheng Zhou, Nan Zhang, Haikuan Huang, Bin Li, and Jiajun Bu. Mas-vqa: A mask-and-select framework for knowledge-based visual question answering. *arXiv preprint arXiv:2602.15915*, 2026.

Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. Encyclopedic VQA: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. URL https://arxiv.org/abs/2306.09224.

Linyong Nan, Weining Fang, Aylin Rasteh, Pouya Lahabi, Weijin Zou, Yilun Zhao, and Arman Cohan. OMG-QA: Building open-domain multi-modal generative question answering systems. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 1001–1015. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-industry.75. URL https://aclanthology.org/2024.emnlp-industry.75/.

Shan Ning, Longtian Qiu, and Xuming He. Wiki-r1: Incentivizing multimodal reasoning for knowledge-based vqa via data and sampling curriculum. *arXiv preprint arXiv:2603.05256*, 2026a.

Xuying Ning, Dongqi Fu, Tianxin Wei, Mengting Ai, Jiaru Zou, Ting-Wei Li, Hanghang Tong, Yada Zhu, Hendrik Hamann, and Jingrui He. MC-Search: Evaluating and enhancing multimodal agentic search with structured long reasoning chains, 2026b. URL https://arxiv.org/abs/2603.00873.

Brandon Ong, Tej Deep Pala, Vernon Toh, William Chandra Tjhi, and Soujanya Poria. Training vision-language process reward models for test-time scaling in multimodal reasoning: Key insights and lessons learned. *arXiv preprint arXiv:2509.23250*, 2025.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition, July 2026. URL https://openai.com/index/gpt-5-6/. Published July 9, 2026. Accessed September 25, 2026.

Timothy Ossowski and Junjie Hu. Retrieving multimodal prompts for generative visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 2518–2535. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.158. URL https://aclanthology.org/2023.findings-acl.158/.

Alin-Ionut Popa. Question-guided evidence acquisition for multimodal visual question answering, 2026. URL https://arxiv.org/abs/2608.19739.

Haozhan Shen, Kangjia Zhao, Tiancheng Zhao, Ruochen Xu, Zilun Zhang, Mingwei Zhu, and Jianwei Yin. ZoomEye: Enhancing multimodal LLMs with human-like zooming capabilities through tree-based image exploration. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6602–6618. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.335. URL https://aclanthology.org/2025.emnlp-main.335/.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Xin Su, Man Luo, Kris W Pan, Tien Pei Chou, Vasudev Lal, and Phillip Howard. SK-VQA: Synthetic knowledge generation at scale for training context-augmented multimodal LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57008–57029. PMLR, 2025. URL https://proceedings.mlr.press/v267/su25a.html.

Manan Suri, Puneet Mathur, Franck Dernoncourt, Kanika Goswami, Ryan A. Rossi, and Dinesh Manocha. Visdom: Multi-document QA with visually rich elements using multimodal retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 6088–6109. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.310. URL https://aclanthology.org/2025.naacl-long.310/.

Ryota Tanaka, Taichi Iki, Taku Hasegawa, Kyosuke Nishida, Kuniko Saito, and Jun Suzuki. Vdocrag: Retrieval-augmented generation over visually-rich documents, 2025. URL https://arxiv.org/abs/2504.09795.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions, 2023. URL https://arxiv.org/abs/2212.10509.

Liang Wang, Haonan Chen, Nan Yang, Xiaolong Huang, Zhicheng Dou, and Furu Wei. Chain-of-retrieval augmented generation, 2025a. URL https://arxiv.org/abs/2501.14342.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, et al. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*, 2025b.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/11718.

Navve Wasserman, Roi Pony, Oshri Naparstek, Adi Raz Goldfarb, Eli Schwartz, Udi Barzelay, and Leonid Karlinsky. REAL-MM-RAG: A real-world multi-modal retrieval benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, pp. 31660–31683. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.1528. URL https://aclanthology.org/2025.acl-long.1528/.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022. URL https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Jinming Wu, Zihao Deng, Wei Li, Yiding Liu, Bo You, Bo Li, Zejun Ma, and Ziwei Liu. MMSearch-r1: Incentivizing LMMs to search. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2456–2487, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.114. URL https://aclanthology.org/2026.acl-long.114/.

ZhengXian Wu, Hangrui Xu, Kai Shi, Zhuohong Chen, Yunyao Yu, Chuanrui Zhang, Zirui Liao, Jun Yang, Zhenyu Yang, Haonan Lu, et al. Promsa: Progressive multimodal search agents for knowledge-based visual question answering. *arXiv preprint arXiv:2606.27974*, 2026b.

Alexandros Xenos, Themos Stafylakis, Ioannis Patras, and Georgios Tzimiropoulos. A simple baseline for knowledge-based visual question answering. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 14871–14877, 2023. doi: 10.18653/v1/2023.emnlp-main.919. URL https://aclanthology.org/2023.emnlp-main.919/.

Yibin Yan and Weidi Xie. Echosight: Advancing visual-language models with wiki knowledge, 2024. URL https://arxiv.org/abs/2407.12735.

Wei Yang, Jingjing Fu, Rui Wang, Jinyu Wang, Lei Song, and Jiang Bian. Omgm: Orchestrate multiple granularities and modalities for efficient multimodal retrieval. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24545–24563, 2025.

Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. MM-REACT: Prompting ChatGPT for multimodal reasoning and action, 2023. URL https://arxiv.org/abs/2303.11381.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

Kai Ye, Xianwei Mao, Sheng Zhou, Zirui Shao, Ye Mo, Liangliang Liu, Haikuan Huang, Bin Li, and Jiajun Bu. Real: Resolving knowledge conflicts in knowledge-intensive visual question answering via reasoning-pivot alignment. *arXiv preprint arXiv:2602.14065*, 2026.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. Visrag: Vision-based retrieval-augmented generation on multi-modality documents, 2025a. URL https://arxiv.org/abs/2410.10594.

Xinlei Yu, Chengming Xu, Zhangquan Chen, Yudong Zhang, Shilin Lu, Cheng Yang, Jiangning Zhang, Shuicheng Yan, and Xiaobin Hu. Visual document understanding and reasoning: A multi-agent collaboration framework with agent-wise adaptive test-time scaling, 2025b. URL https://arxiv.org/abs/2508.03404.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Seyed Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/214308a2d5e3f83ef9ad2739e1cbc46d-Abstract-Conference.html.

James Xu Zhao, Hui Chen, Bryan Hooi, and See-Kiong Ng. Fineverify: Scaling test-time compute with fine-grained self-verification for agentic search. *arXiv preprint arXiv:2606.00660*, 2026.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, Xiaojie Cai, Lyumanshan Ye, Pengrui Lu, and Pengfei Liu. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.22. URL https://aclanthology.org/2025.emnlp-main.22/.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62138–62160. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhou24r.html.

APPENDIX

A Limitations . . . 17

B Grounding-Trap Observations and Conditional Analysis . . . 17
- B.1 Entity-Conditioned Evidence and Grounding Ambiguity . . . 17
- B.2 Why Additional Facts May Not Resolve the Grounding . . . 17
- B.3 When a Trajectory Remains Committed . . . 17
- B.4 Independent Exploration and Candidate Selection . . . 18
- B.5 Exploratory Grounding-Trap Analysis . . . 18

C DiVE Agent Details . . . 19
- C.1 Agent Instruction . . . 19
- C.2 Agent Tool Details . . . 19
- C.3 CTV Agent Construction . . . 20

D Training Details of DiVE . . . 21

E KB-VQA Benchmarks . . . 23

F Implementation Details . . . 24
- F.1 Knowledge Base Setting . . . 24
- F.2 Student Optimization . . . 24
- F.3 Training Details . . . 24
- F.4 Shared TTS Protocol . . . 24
- F.5 KB-VQA Baselines . . . 24

G Detailed Related Work . . . 26

H Additional Experimental Results and Analysis . . . 27
- H.1 Paired Statistical Comparisons on E-VQA . . . 27
- H.2 Detailed Tool Ablation Experiments . . . 27
- H.3 Fine-Grained Verifier Diagnostics . . . 28
- H.4 Cross-Benchmark Generalization . . . 29
- H.5 Coverage of Terminal Entity Assignment . . . 32

I Case Study . . . 32

A LIMITATIONS

We discuss two limitations of DiVE. First, scaling gains diminish as the rollout count increases. As shown in Section 5.4, Oracle@N continues to increase, but its growth flattens as the number of samples (N) rises; the verifier's path-selection gains also taper. Thus, simply sampling more trajectories with the current agent and verifier yields progressively smaller returns, especially on difficult questions. Second, independently sampled rollouts do not transfer memory from earlier sampling stages to later ones and may therefore repeat previously observed errors. Memory-guided exploration could reduce this redundancy and is a promising direction for future work.

B GROUNDING-TRAP OBSERVATIONS AND CONDITIONAL ANALYSIS

We give a conditional interpretation of the grounding-trap observation in Section 3 through information availability, hypothesis persistence, and candidate selection. The derivations illustrate consequences of explicit assumptions.

B.1 ENTITY-CONDITIONED EVIDENCE AND GROUNDING AMBIGUITY

Let $e^\star$ be the entity depicted in I, $\tilde{e} \neq e^\star$ a confusable alternative, and $\mathcal{K}_e$ the facts associated with entity e. For a question with a unique answer up to paraphrase, write $a(e, q)$ for its answer about e and $a^\star = a(e^\star, q)$. We consider grounding errors that change the answer: $a(\tilde{e}, q) \not\equiv a^\star$. Evidence $\mathcal{E}_T$ collected about $\tilde{e}$ may satisfy

$$\mathcal{E}_T \subseteq \mathcal{K}_{\tilde{e}}, \qquad \mathcal{E}_T \models [a(\tilde{e}, q) \equiv \tilde{a}], \qquad \tilde{a} \not\equiv a^\star. \tag{9}$$

Such evidence supports a fact about the selected entity without establishing that this entity appears in the image.

B.2 WHY ADDITIONAL FACTS MAY NOT RESOLVE THE GROUNDING

Set $t = 0$ when the agent adopts $\tilde{e}$. Let v_0 be the extracted visual information, h_0 the preceding history, and $C_0 = (q, v_0, h_0, \tilde{e})$. Let $Z \in \{e^\star, \tilde{e}\}$ represent the unknown identity, with $Z = e^\star$ in the failure case. An analytical distribution with $\mathrm{H}(Z \mid C_0) > 0$ describes ambiguity relative to this information state, not the MLLM's internal beliefs. We use conditional entropy H and mutual information I with base-two logarithms (Cover & Thomas, 2006, Chapter 2).

Observation model. The analyzed continuation generates (r_t, u_t) only from (C_0, h_t) and randomness independent of Z given these inputs. Access to visual information outside this state is treated as an observation. Accordingly,

$$\mathrm{I}\big(Z; (r_t, u_t) \mid C_0, h_t\big) = 0, \qquad \Delta_t = \mathrm{I}\big(Z; o_t \mid C_0, h_t, r_t, u_t\big). \tag{10}$$

An observation adds no grounding information when $o_t \perp Z \mid C_0, h_t, r_t, u_t$, equivalently $\Delta_t = 0$. This is a condition on the observation distribution, not on its modality: entity-scoped facts may satisfy it, whereas text linking an observed visual feature to an entity may not.

Conditional information identity. For a fixed horizon $T \geq 1$, with early termination padded by null transitions,

$$\mathrm{I}(Z; h_T \mid C_0) = \mathrm{H}(Z \mid C_0) - \mathrm{H}(Z \mid C_0, h_T) = \sum_{t=0}^{T-1} \Delta_t. \tag{11}$$

Derivation. Since h_0 is included in C_0, the chain rule expands the appended transitions (r_t, u_t, o_t). Equation 10 removes the action terms; the entropy identity follows from the definition of conditional mutual information.

Thus, if every $\Delta_t = 0$, the continuation adds no identity information. For any verification decision D_T based only on (C_0, h_T) and independent randomness, conditional data processing gives $\mathrm{I}(Z; D_T \mid C_0) \leq \sum_t \Delta_t$. This limits new information, not a computationally limited agent's ability to use information already available; it does not establish an accuracy bound for such an agent.

B.3 WHEN A TRAJECTORY REMAINS COMMITTED

Information limitation alone does not prevent hypothesis revision. Let S_t be the event that the policy has not revised $\tilde{e}$ in its first t continuation steps, so S_0 is certain and $S_{t+1} \subseteq S_t$. Termination without revision preserves this event. As a separate behavioral assumption, suppose that, whenever $P(S_t \mid C_0) > 0$,

$$P(S_{t+1}^c \mid S_t, C_0) \leq \varepsilon, \qquad 0 \leq \varepsilon \leq 1, \quad t < T. \tag{12}$$

Conditional persistence bound. Under this assumption,

$$P(S_T \mid C_0) \geq (1 - \varepsilon)^T, \qquad P(S_T^c \mid C_0) \leq 1 - (1 - \varepsilon)^T \leq T\varepsilon. \tag{13}$$

Derivation. Nestedness and Equation 12 give $P(S_{t+1} \mid C_0) \geq (1 - \varepsilon) P(S_t \mid C_0)$, also trivially when $P(S_t \mid C_0) = 0$. Iterate from S_0 and apply Bernoulli's inequality. No independence across steps is required.

For $T\varepsilon \ll 1$, revision is unlikely; $\varepsilon = 0$ rules it out within the horizon. This assumption does not estimate actual revision rates, and revision need not identify $e^\star$.

B.4 Independent Exploration and Candidate Selection

Independent rollouts restart from $x = (I, q)$ with separate histories, allowing alternative hypotheses without guaranteeing entity diversity. For a fixed policy, knowledge base, toolset, and per-rollout budget B, define

$$p_B(x) = P(a_i \equiv a^\star \mid x; B), \qquad \mathcal{C}_N = \{\exists i \le N : a_i \equiv a^\star\}. \tag{14}$$

This coverage analysis allows reasoning errors and shared answers across entities; $p_B(x)$ measures answer correctness. For $N \ge 1$ i.i.d. rollouts, all candidates fail with probability $(1 - p_B(x))^N$, so

$$P(\mathcal{C}_N \mid x; B) = 1 - (1 - p_B(x))^N. \tag{15}$$

This is the Oracle@N coverage probability. It increases strictly with N when $0 < p_B(x) < 1$, but remains zero when $p_B(x) = 0$. Benchmark coverage averages this expression over inputs.

For a verifier that selects an existing candidate answer, define, whenever $P(\mathcal{C}_N \mid x; B) > 0$,

$$\alpha_N(x) = P(\hat{a} \equiv a^\star \mid \mathcal{C}_N, x; B), \tag{16}$$

and set $\alpha_N(x) = 0$ otherwise. Since a correct selection requires a correct candidate, the law of total probability gives

$$P(\hat{a} \equiv a^\star \mid x; B) = \alpha_N(x)\big[1 - (1 - p_B(x))^N\big]. \tag{17}$$

CTV aims to improve $\alpha_N(x)$ by jointly comparing recorded evidence. Selection depends on the grounding cues represented in the forest; improvements over other selectors require empirical validation. Coverage alone does not ensure increasing accuracy, since $\alpha_N(x)$ may fall with N. The fixed-B analysis does not compare alternative allocations of the same total compute.

B.5 Exploratory Grounding-Trap Analysis

To document the grounding-trap, we conduct an exploratory post-hoc audit of a vanilla search agent on the 3,750-question E-VQA Single test set. The agent uses EVA-CLIP for image retrieval and BGE for text retrieval (Wu et al., 2026b). Official BEM marks 2,251 answers incorrect. We choose GPT-5.6 Sol as the auditor based on its reported multimodal reasoning and long-context retrieval performance: 83.0% on MMMU-Pro (no tools) and 91.5% on MRCR v2 (8-needle, 256K–512K) under OpenAI's evaluation settings (OpenAI, 2026). For each error, GPT-5.6 Sol receives the query image and question, the full recorded interaction trajectory, the gold entity label, and the reference answer. The audit first extracts the final answer, the entity to which the agent commits, and text-retrieval evidence tagged by entity, then checks whether the answer and entity actually differ from their gold references.

Grounding-trap labeling requires strict within-trajectory evidence: the agent's own words must explicitly commit to the wrong entity, and a tool result for that same entity must contain a verbatim passage supporting the predicted answer for the same attribute asked in the question. Both gold mismatches and both evidence checks must hold. Gold entity labels and reference answers are used only for this post-hoc audit, not agent inference. Table 3 reports the resulting automatic categories.

Table 3: Automatic categories for the 2,251 BEM-incorrect E-VQA Single answers. The four categories are mutually exclusive.

Category	**Count**	**Ratio**
Grounding trap: Wrong entity, trace-supported answer	556	24.70%
Wrong entity, no qualifying trace support	1,376	61.13%
Correct entity, wrong answer	233	10.35%
Others	86	3.82%
Total BEM errors	2,251	

"Others" covers responses without a usable evidence-grounded answer, such as "not mentioned" or a direct answer that does not use retrieved facts. Among the 2,251 errors, 1,932 (85.83%) involve a wrong entity, whereas 233 (10.35%) follow identification of the correct entity. In 556 cases (24.70%), the auditor judged that the wrong-entity commitment and same-entity evidence met the above criteria. This audit provides observational evidence of wrong-entity commitment accompanied by answer-supporting tool evidence in this agent and dataset. The 556/2,251 (24.70%) figure is a descriptive proportion flagged by this GPT-5.6 Sol auditor, not a calibrated estimate of the true prevalence of grounding traps. Entity commitment, attribute matching, and evidence entailment may still be misjudged without human or cross-judge validation, so we use the audit to characterize a possible failure pattern rather than establish its general importance or a causal mechanism.

DiVE Prompt

Role
You are a step-by-step reasoning assistant. Given a question and image, solve the problem one substep at a time.

Exact entity identification
One depicted entity may have multiple extremely similar entities, and several candidate pages may contain information that appears capable of answering the question. You should prioritize identifying and verifying the exact depicted entity before attempting to answer the question. Treat every retrieved entity as a candidate, not an identity confirmation. Use the available crop, visual knowledge-base search, and scoped text-search tools as needed, compare the evidence, and make every reasonable effort to determine the exact depicted entity before answering. When image_search_tool returns multiple visually plausible candidates, inspect the necessary reference images with image_source_tool by copying the exact entity name from a candidate line. image_source_tool resolves these representative images internally; use entity names, never image paths, in text_search_tool entities.

Guiding Principles
At each turn, you must either issue one specific tool call enclosed in <tool_call> </tool_call> tags, or provide the final answer enclosed in <answer> </answer> tags.

All outputs must begin with a thought enclosed in <thinking> </thinking> tags, explaining your current reasoning and what to do next.

Output Format (strict)
Always start with <thinking>. Do not output the previous reasoning chain. Then output exactly one of:

<thinking> Your current reasoning and next plan </thinking>
<tool_call> One precise tool call to assist your reasoning </tool_call>

or:

<thinking> Summarize the decisive evidence and derive the answer </thinking>
<answer> Final answer </answer>

Query
<image>[image]</image>
What do the Pashto people call this bird?

Figure 8: DiVE prompt format.

C DiVE Agent Details

C.1 Agent Instruction

E-VQA and InfoSeek use the same system instruction and the same four tool schemas; no benchmark-specific instruction is added to either prompt. The user input consists of the query image followed by the raw benchmark question. The complete DiVE agent instruction is shown in Figure 8.

C.2 Agent Tool Details

`image_crop_tool`. This tool takes a bounding box `bbox_2d` and an optional `image_index`. Coordinates are expressed relative to a 0–1000 grid, and `image_index=0` denotes the original query image. The selected region is cropped, resized when necessary, appended to the multimodal conversation, and assigned a new image index. The returned crop can be inspected directly or passed to `image_search_tool`; the agent may also crop a previously generated image when a finer region is required.

`image_search_tool`. Given the index of the original image or a crop, this tool queries three EVA-CLIP indexes constructed from entity titles, representative images, and entity summaries. Each route retrieves its own ranked candidates. We independently min–max normalize the route scores, merge candidates referring to the same entity, and rank the union by the sum of their normalized scores. Although the backend retains a deeper

Table 4: Agent-facing tool interface. An image index of 0 always denotes the query image. Returned images are appended to the trajectory and assigned the next available index.

Tool	Arguments	Returned observation
image search tool	Optional integer `image_index` (default 0).	A ranked list of at most ten entity titles. The titles are candidates, not identity decisions.
image source tool	Exact string entity copied from a candidate line.	The query-relevant representative image, its entity title, and its new image index.
image crop tool	Required `bbox_2d` $= [x_1, y_1, x_2, y_2]$ on a 0–1000 grid; optional `image_index`.	The cropped image and its new image index.
text search tool	Required short `query` and self-contained `bge_query`; optional entities, regex, context-window sizes, and output cap.	Up to three exact/regex windows per supplied entity. If fewer than five entities are supplied, three BGE-retrieved, question-conditioned section summaries are appended as a recall fallback.

candidate pool for fusion, the tool exposes at most ten entity names to the agent. The returned ordering is used only for candidate discovery; neither a high rank nor agreement across indexes is treated as identity verification.

`image_source_tool`. Given an exact entity name from `image_search_tool`, this tool resolves a representative image for that entity, resizes it to the supported visual-token range, appends it to the trajectory, and returns its new image index. When multiple reference images are available, the implementation selects a query-relevant representative using the original image and falls back to the stored entity–image mapping if online selection is unavailable. The tool itself produces no entity decision: the agent compares the returned reference image with the original image and available crops, and may inspect several candidates before continuing.

`text_search_tool`. The agent supplies short answer-bearing terms through `query`, a list of candidate titles through `entities`, and optionally enables regular-expression matching through `regex`. The tool searches only the specified entity pages; global corpus search is disabled. At most five entity pages are inspected per call, with up to three matched windows per page. By default, each match retains 220 characters on both sides, while the agent may request asymmetric windows of up to 500 characters per side; the combined CLI output is capped at 4096 characters. Queries containing alternatives such as `height|length|wingspan` automatically activate regular-expression matching. As an implementation-level recall fallback, when fewer than five entity titles are supplied, the tool additionally uses `bge_query` to retrieve the top three semantically relevant sections and compresses each section with a question-conditioned summarizer. The resulting summaries are appended as candidate observations and do not replace the entity-scoped CLI evidence or constitute the core interaction mechanism.

The tool schemas are appended to this instruction inside `<tools>` tags. Table 4 records the exposed arguments and the observation returned to the policy. Tool observations are wrapped in `<tool_response>` tags and supplied as a user turn.

Tool-return examples. Examples of real observations from the Balanced-2K trajectories are shown in Figure 9. The `<image>` token below marks image outputs, and the long text-search return is abridged only where explicitly marked.

These examples also illustrate the division of labor in the interface: visual retrieval proposes entity names, source-image and crop returns expose visual evidence, and text search returns evidence only from a bounded image-grounded workspace plus its explicitly marked recall fallback. Each trajectory allows up to 15 interaction steps.

C.3 CTV Agent Construction

The search agent observes the query image, reference images, and crops during its rollouts. CTV receives the question, candidate entity names, and a serialized forest of tool-call records, textual observations, and candidate answers. Based on this example, we analyze the DiVE agent's behavior across multiple sampling runs and observe how the "grounding trap" manifests in a real query. As shown in Figure 11, the DiVE agent generally follows the logic of visual grounding and corpus interaction during each run. Across eight independent inference attempts, the forest contains eight labeled answer leaves, including non-empty abstentions stating that the requested information is unavailable. Only one run identified the actual entity and returned the correct answer in this example. Notably, even in that successful case, multiple visual verification steps were performed during

Tool-Observation Examples

image_search_tool:
{"image_index": 0}
RETURN:
1. Demoiselle crane
2. Common crane
3. Crane (bird)
4. White-naped crane
5. Ohrada Zoo
6. Tsolwana Nature Reserve
7. Panmun Field
8. List of Gruiformes by population
9. List of birds of South Africa
10. List of birds of Namibia

image_source_tool:
{"entity": "Demoiselle crane"}
RETURN:
Here is the representative image for Demoiselle crane (image_index=1):
<image>

image_crop_tool:
{"bbox_2d": [450, 393, 550, 507], "image_index": 0}
RETURN:
Here is the cropped image (image_index=2):
<image>

text_search_tool:
{"query": "diet|food|prey",
"bge_query": "What do White-tailed eagles commonly eat? Include details about their primary prey and feeding habits based on visual clues like their size and habitat.",
"entities": ["White-tailed eagle"]}
RETURN:
Local KB text search results (mode=entity):
entity: White-tailed eagle
matches: 3
1. ... an abundant food supply of fish and birds (largely water birds) amongst nearly any other available prey. Both a powerful apex predator and an opportunistic scavenger ...
[two further bounded context windows omitted]

BGE fallback candidate sections:
Candidate entity: White-tailed eagle
Evidence: White-tailed eagles have a varied diet that includes fish,
birds, and mammals, with fish and water birds as primary food sources ...

Figure 9: DiVE tool-return examples.

the visual grounding phase. This example illustrates how reasoning under a visually similar but incorrect entity hypothesis can yield a coherent yet incorrect answer.

Algorithm 1 summarizes the complete inference procedure of CTV. Given the frozen trajectory collection $\mathcal{R}_N$, CTV first assigns each trajectory a terminal entity state, then groups trajectories with the same state into an entity-evidence forest while consolidating repeated observations. The forest is serialized together with the question and candidate entity names for a single verifier call. The verifier selects one valid trajectory, whose answer is returned unchanged. CTV does not invoke additional tools, continue existing trajectories, or synthesize a new answer during verification.

D Training Details of DiVE

Candidate construction. We construct disjoint candidate pools from the training splits of E-VQA and InfoSeek. E-VQA candidates are deduplicated and filtered for title leakage and invalid images. InfoSeek images

Algorithm 1 Cross-Trajectory Verification (CTV).

Require: question q; candidate entity names; frozen trajectories $\mathcal{R}_N$; verifier $\mathcal{V}$
Ensure: selected trajectory $\tau_{\hat{\imath}}$ and its unchanged answer $a_{\hat{\imath}}$

Entity assignment
for each trajectory $\tau_i \in \mathcal{R}_N$ **do**
  extract the entity states from its entity-bearing tool calls
  collapse consecutive duplicate states to obtain $\mathcal{E}_i$
  assign $\hat{E}_i$ using Equation 7
end for
Forest construction
initialize $\mathcal{F}_N$ with one root for each distinct $\hat{E}_i$
for each trajectory $\tau_i \in \mathcal{R}_N$ **do**
  create a branch under root $\hat{E}_i$ and retain its route $\mathcal{E}_i$
  for each recorded tool call in τ_i, in order **do**
    map the call to a navigation, grounding, or evidence node
    reuse a matching node under $\hat{E}_i$, or create one if none exists
    append the node and its return status to the branch
  end for
  append the unchanged answer a_i as the leaf
end for
Cross-trajectory selection
present q, candidate entity names, and the serialized forest to $\mathcal{V}$
ask $\mathcal{V}$ to return the valid non-empty trajectories in preference order
$\hat{\imath} \leftarrow$ the first returned trajectory label
return $(\tau_{\hat{\imath}}, a_{\hat{\imath}})$

Cross-Trajectory Verification Prompt

Role
You are a verifier for frozen KB-VQA trajectories. Judge only the original entity, raw tool returns, supplied compare entities, required evidence hops, and final entailment. Retrieval rank, fluent prose, and repeated answers are not evidence. Reject wrong entity identity, reversed relations, unsupported bridge hops, and answers not entailed by the observations.

Rank every listed valid answer leaf from strongest to weakest. A valid leaf means only that its frozen trajectory produced a non-empty answer; it is not evidence of correctness. Judge raw observation support, entity grounding, and complete answer-slot entailment. Return every supplied valid label exactly once. Repetition and tree size are not votes.

Input
Question: {benchmark question}

All supported tool actions and process states are retained.
Raw observations are exactly deduplicated inside each tree; no semantic fact filtering is applied.
Branch A-O-I sequences reference the deduplicated observation node IDs.
INFERENCE is sampled-path interpretation, not evidence. No original or comparison image is supplied.
Tree visits are correlated Monte Carlo samples, not correctness votes.

[entity-evidence forest]

VALID ANSWER LEAVES TO RANK: A, B, C, D, E, F, G, H

Figure 10: Cross-Trajectory Verification prompt.

are aligned with M-BEIR, and candidates without valid answers, reference entity images, or title-leakage-free questions are removed. The teacher receives only the query image and question; answers, entities, and reference evidence are stored in a separate gold sidecar used exclusively for post-hoc evaluation.

Teacher rollout collection. We use Qwen3-VL-235B-A22B-Instruct as the teacher policy π_{T} and sample $N_{\mathrm{T}} = 4$ trajectories for each question. This offline teacher rollout count is independent of the test-time rollout count N. The agent interacts with the query image and external knowledge through `image_search_tool`, `image_source_tool`, `text_search_tool`, and `image_crop_tool`. We collect 2,000 questions from each benchmark, producing approximately 16,000 teacher trajectories in total. For each training instance x_n, the rollouts are sampled independently from the trajectory distribution induced by the teacher and its tool interactions:

$$\tau_{n,i} \overset{\text{i.i.d.}}{\sim} p_{\pi_{\mathrm{T}}}(\tau \mid x_n, \mathcal{K}, \mathcal{T}), \qquad i = 1, \ldots, N_{\mathrm{T}}. \tag{18}$$

Post-hoc scoring and structural filtering. Teacher generation is gold-blind. After rollout, E-VQA answers are evaluated using the official BEM score, with a threshold of 0.5, while InfoSeek answers are evaluated using Modified Relaxed Accuracy. Correct trajectories must additionally satisfy the prescribed message and tool-call schema, invoke the required image and text search tools, contain no failed tool response, remain within the interaction budget, and resolve every image marker to an existing file. We do not require retrieval of the gold entity, strict gold-evidence closure, or exact replay of tool responses; the retained samples are therefore answer-correct and structurally valid trajectories rather than manually verified perfect evidence chains.

SFT data selection. After generation, each final answer receives a correctness indicator $c_{n,i} \in \{0, 1\}$ under the scoring protocol above. The empirical teacher success rate for question x_n is

$$\rho_n = \frac{1}{N_{\mathrm{T}}} \sum_{i=1}^{N_{\mathrm{T}}} c_{n,i}. \tag{19}$$

We refer to questions with $\rho_n = 1$ as *stable* and those with $0 < \rho_n < 1$ as *challenging*; these names describe observed teacher outcomes, not student difficulty. Questions with $\rho_n = 0$ provide no answer-correct demonstration and are discarded. Among the remaining questions, we retain only those with at least one trajectory that also passes structural filtering.

Within each benchmark, we retain 600 stable questions with $\rho_n = 1$, for which all four rollouts are correct, and 400 challenging questions with $0 < \rho_n < 1$, for which one to three rollouts are correct. Each question contributes one correct and structurally valid trajectory τ_n^{sel}. When multiple candidates remain, we select the trajectory with the fewest assistant turns, followed by the fewest tool calls, source-run order, and rollout index. This yields 1,000 trajectories per benchmark and 2,000 SFT trajectories in total, corresponding to a stable-stratum weight $\lambda = 0.6$ in the empirical SFT distribution of Equation 20. For completeness, this data composition can be written as

$$p_{\mathrm{SFT}}(x, \tau) = \lambda p_{\mathrm{stable}}(x, \tau) + (1 - \lambda) p_{\mathrm{challenging}}(x, \tau). \tag{20}$$

Here p_{stable} and $p_{\mathrm{challenging}}$ are the empirical distributions over the retained pairs $(x_n, \tau_n^{\mathrm{sel}})$ in each group, and $\lambda \in [0, 1]$ is the proportion of stable questions. All selected trajectories are trained with the same standard SFT objective in Equation 6.

E KB-VQA BENCHMARKS

Encyclopedic-VQA. Encyclopedic-VQA (E-VQA) evaluates whether a model can answer questions about detailed properties of fine-grained visual categories and instances (Mensink et al., 2023). The complete benchmark contains 221K unique question–answer pairs, each paired with up to five images, yielding approximately one million image–question–answer triplets. It also provides a controlled Wikipedia-derived knowledge base and the supporting evidence for each answer. Following prior KB-VQA agents such as ProMSA (Wu et al., 2026b), we evaluate on the official *Single* test subset, which contains 3,750 questions: 2,750 automatically generated questions and 1,000 templated questions. We use the official BERT-based Matching (BEM) metric, which judges semantic agreement between a generated answer and the reference answer.

InfoSeek. InfoSeek targets visual information-seeking questions whose answers require fine-grained knowledge about the entity depicted in an image rather than visual perception or common-sense knowledge alone Chen et al. (2023b). The benchmark contains a human-written component with 8.9K examples and a large automatically constructed Wikidata component with approximately 1.35M examples spanning 11,481 entities and 2,739 entity types. Its evaluation protocol separates two forms of generalization: *unseen entity*, in which the queried entity does not occur during training, and *unseen question*, in which the entity is seen but the corresponding question–answer pair is held out. We evaluate on all 71,335 validation examples from these two splits, comprising 53,202 unseen-entity and 18,133 unseen-question examples. Following prior work (Yan & Xie, 2024), we report VQA accuracy for string and time questions and relaxed accuracy for numerical questions.

F Implementation Details

F.1 Knowledge Base Setting

To maintain consistency with prior work (Yang et al., 2025; Yan & Xie, 2024) and ensure that performance gains stem from tool use rather than a different multimodal encoder, we use the same EVA-CLIP-8B encoder. We use the original two million Wikipedia articles with images Mensink et al. (2023) for E-VQA and the 100K-entity InfoSeek subset released by EchoSight Yan & Xie (2024). For indexed search, we use the off-the-shelf index files released by prior work (Yang et al., 2025; Cocchi et al., 2025; Yan & Xie, 2024).

F.2 Student Optimization

The student agent is trained for three epochs. We freeze the visual encoder and multimodal projector while fully updating the language-model parameters. We use FSDP over eight GPUs, an effective global batch size of 32, a peak learning rate of 1×10^{-5}, cosine decay, a 5% warm-up ratio, and weight decay of 0.01. The maximum sequence length is 8,192 tokens and the maximum image budget is 262,144 pixels. The loss is applied only to assistant-generated tokens; system messages, user inputs, and tool observations are masked.

F.3 Training Details

All four SFT runs use a single node with $8\times$ NVIDIA H100 80GB HBM3 GPUs and FSDP, training on 2,000 trajectories for three epochs. Table 5 reports the elapsed time of each training process, rather than the requested Slurm job time limit.

Qwen3.5-9B uses 24 linear-attention layers and eight full-attention layers. In our shared training environment, the runs did not use `fla`, `causal_conv1d`, or `kernels`; the Transformers implementation provides a PyTorch fallback for the linear-attention operations. This interaction between the hybrid architecture and available operator implementations is a plausible contributor to the longer training time in Table 5.

Figure 12 reports the logged assistant-token training loss for all four policies trained on the same 2,000 trajectories. Each run uses three epochs and 189 optimizer updates. The logs record step 1, every fifth update, and the final update; the figure plots these raw values without smoothing or interpolation. Loss decreases substantially for every model, from 1.013 to 0.367 for Qwen3-VL-2B, 0.731 to 0.228 for InternVL3.5-8B, 0.719 to 0.208 for Qwen3-VL-8B, and 0.613 to 0.119 for Qwen3.5-9B. Because architectures and tokenizers differ, the absolute loss levels should be read as within-run optimization diagnostics rather than as a cross-model quality ranking.

Table 5: Training-process elapsed time for the four SFT runs. Parenthetical durations are rounded to the nearest second.

Model	Training time
Qwen3-VL-8B	781.7 s (13 min 02 s)
Qwen3-VL-2B	403.2 s (6 min 43 s)
Qwen3.5-9B	3,849.0 s (1 h 04 min 09 s)
InternVL3.5-8B	850.9 s (14 min 11 s)

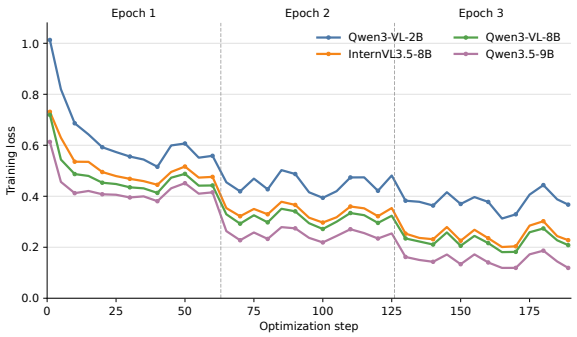


Figure 12: Training loss on the identical Balanced-2K trajectory set. Dashed lines mark epoch boundaries at optimizer steps 63 and 126.

F.4 Shared TTS Protocol

For each evaluation query, we independently sample $N = 8$ complete trajectories from DiVE using different random seeds. We set the temperature to 1.0 and top-p to 1.0. Once generation finishes, all test-time scaling methods select among the eight candidates. All generative TTS selectors use greedy decoding. We use Qwen3-VL-8B as the CTV listwise verifier. The verifier is decoded greedily with temperature 0 and is constrained by a JSON schema. We use the same frozen candidate pool for all test-time scaling baselines. Thus, the compared TTS methods differ in selection procedure, while the source rollouts remain fixed.

F.5 KB-VQA Baselines

We summarize the comparison methods in Table 1, covering multimodal retrieval and knowledge processing, entity-grounding and search frameworks, and test-time scaling.

EchoSight. EchoSight (Yan & Xie, 2024) first retrieves Wikipedia articles through image-to-image similarity. A trained Q-Former reranker then combines the query image and question to rank article sections, and the selected knowledge is passed to an answer generator.

OMGM. OMGM (Yang et al., 2025) adopts coarse-to-fine retrieval across modalities and knowledge granularities. It first matches the query image against entity summaries, then applies multimodal fusion reranking and question-conditioned section selection to obtain fine-grained evidence for answer generation.

VLM-PRF. VLM-PRF (Hong et al., 2025) organizes knowledge acquisition into processing, retrieval, and filtering stages. Tool-assisted visual processing improves the retrieval query, while a reinforcement-trained VLM filters retrieved content into task-relevant knowledge using answer-accuracy and format-consistency rewards.

REAL. REAL (Ye et al., 2026) detects knowledge conflicts at reasoning pivots, namely entities or relations essential to answering the question. Pivot-aware supervised fine-tuning trains a conflict discriminator, whose predictions guide a contrastive decoding strategy that reduces interference from conflicting evidence.

MaS-VQA. MaS-VQA (Mao et al., 2026) uses retrieved knowledge to guide visual masking and question-conditioned phrase selection, suppressing irrelevant image regions and textual fragments. The filtered evidence then conditions a frozen MLLM to generate complementary knowledge clues, which are combined with the explicit evidence for final prediction.

CC-VQA. CC-VQA (Hong et al., 2026b) externalizes the model's parametric knowledge and compares it with retrieved contexts through vision-centric conflict reasoning. It further uses correlation-aware positional compression and adaptive decoding to reduce the influence of weakly relevant or conflicting information during answer generation.

IBA-Qwen. IBA-Qwen uses the identify-before-answer framework (Ma et al., 2026b) to separate entity identification from evidence ranking. Given the query image and retrieved candidate names, a Qwen-based MLLM selects high-confidence entities; an off-the-shelf text reranker then selects supporting sections within their articles for answer generation.

MMSearch-R1. MMSearch-R1 (Wu et al., 2026a) integrates image and text search into multi-turn multimodal reasoning. It uses GRPO with an answer-based reward and a search-call penalty to learn when to retrieve external information, how to formulate queries, and when to answer.

DeepEyesV2. DeepEyesV2 (Hong et al., 2026a) interleaves web search and executable code within a multimodal reasoning loop, supporting both knowledge acquisition and visual or numerical analysis. Cold-start supervised fine-tuning establishes tool-use patterns, followed by reinforcement learning to refine context-dependent tool selection and combinations.

ProMSA. ProMSA (Wu et al., 2026b) learns a budgeted KB-VQA search policy that chooses among image retrieval, text retrieval, and termination, using deduplication and question-conditioned summaries to manage retrieved evidence. Rejection-sampling SFT provides a cold start, followed by Tool-Normalized GSPO, which accounts for both generation length and tool-interaction depth. For our reproduction, ProMSA's official training and inference code was available, but its knowledge base, training data, and checkpoints were not. We therefore report our implementation of ProMSA using our tool set and the officially released reward and algorithm. Following the official recipe, we run three rounds of reinforcement learning using TN-GSPO and the proposed reward.

Self-consistency. Self-consistency aggregates independently sampled reasoning paths through agreement among their final answers (Wang et al., 2023). In our evaluation, it groups semantically equivalent terminal answers and selects from the largest group.

VisualPRM. VisualPRM is a multimodal process reward model trained to assess intermediate reasoning steps (Wang et al., 2025b). We use VisualPRM-8B-v1.1 to score candidates independently and perform Best-of-N selection, rather than voting over terminal answers.

FineVerify. FineVerify decomposes a question into checkable sub-questions, evaluates each candidate against these shared criteria, and selects the candidate with the highest aggregated score (Zhao et al., 2026). Our KB-VQA adaptation verifies candidates independently using fresh entity-scoped evidence for the atomic checks.

TTSP. TTSP uses entropy-gated perceptual exploration and an Evidence Ledger to guide later visual trajectories (Jiang et al., 2026). We adapted TTSP to an eight-trajectory E-VQA budget using two rounds of four fresh trajectories and one Ledger update while otherwise retaining the official parameters; it obtained 13.63% raw official BEM, versus 53.81% for historical CTV. The frozen CTV pool lacks per-token entropy and cannot reproduce TTSP's feedback-guided sampling. This end-to-end comparison therefore matches only trajectory count—not pools, tokens, models, or tools—and neither reproduces TTSP's official 32-trajectory setting nor supports causal or broader claims. We did not tune the prompt, parameters, seeds, or tools after observing the result, and we did not extend the experiment to InfoSeek. Matched-total-token comparisons remain future work.

G Detailed Related Work

KB–VQA retrieval, decomposition, and training data. The Visual Retriever-Reader pipeline learns retrieval and answer prediction with weak supervision (Luo et al., 2021). Barezi & Kordjamshidi (2024) decompose complex questions into visual and knowledge-based sub-questions, assigning them to a captioner and an LLM, respectively. SK-VQA supplies large-scale synthetic question–answer pairs with external knowledge for context-augmented multimodal training (Su et al., 2025). These works address evidence retrieval, modality-specific decomposition, and training data; CTV studies inference-time selection among completed entity-grounded trajectories.

Fine-grained multimodal retrieval. ReViz learns retrieval from multimodal queries without first reducing the query to a text-only description (Luo et al., 2023). FLMR addresses fine-grained query–passage relevance through late interaction, and PreFLMR scales this approach through multimodal retrieval pretraining (Lin et al., 2023; 2024). Cross-modal and mono-modal retrieval provide complementary signals for named-entity KB-VQA (Lerner et al., 2024). LLM-assisted retrieval identifies question-relevant visual entities (Jian et al., 2024), while multi-level retrieval jointly considers entity and passage evidence (Adjali et al., 2024). These approaches motivate preserving entity-specific evidence, but retrieval relevance alone does not certify that the selected entity matches the query image.

Prompting and adaptive knowledge acquisition. Retrieval-augmented generation combines parametric generation with externally retrieved knowledge (Lewis et al., 2020), while chain-of-thought prompting supplies intermediate reasoning exemplars (Wei et al., 2022). These are complementary ingredients rather than substitutes for verifying the visual identity of a queried entity. Training-free in-context KB-VQA uses question-informative captions (Xenos et al., 2023); multimodal prompt retrieval also supports generative VQA in zero- and few-shot transfer settings (Ossowski & Hu, 2023). These alternatives are important controls for evaluating the necessity of training and tool use. In text-based QA, IRCoT interleaves retrieval with reasoning (Trivedi et al., 2023), FLARE triggers retrieval using predicted upcoming content and uncertainty (Jiang et al., 2023), and Self-RAG learns retrieval and critique with reflection tokens (Asai et al., 2023). CoRAG trains intermediate retrieval chains and scales their length or number at inference (Wang et al., 2025a); Search-o1 integrates agentic search with a separate document-reasoning module (Li et al., 2025a). These methods distinguish acquiring missing facts from reasoning over already retrieved facts, a distinction that remains important when the initial entity hypothesis is uncertain.

Tool use and evidence access. MM-REACT connects language reasoning with vision experts through textualized tool interfaces (Yang et al., 2023). OmniSearch adapts multimodal retrieval plans to dynamic and multi-hop questions (Li et al., 2025b), while DeepResearcher trains agents through reinforcement learning in real-world web environments (Zheng et al., 2025). At the interface level, DR-DCI combines retrieval-based workspace construction with direct corpus operations (Lu et al., 2026), and A-RAG exposes keyword search, semantic search, and chunk reading (Du et al., 2026). DiVE similarly combines indexed discovery with local evidence inspection, but focuses on visually confusable entity hypotheses and post-hoc comparison of completed trajectories.

Active perception and grounded search. ZoomEye searches a hierarchy of image regions for task-relevant visual details (Shen et al., 2025). Q-Guide chooses targeted reading, zooming, and grounding tools from the question and missing evidence (Popa, 2026). SpotAgent couples visual geo-localization with tool-assisted verification, including web search and maps (Jia et al., 2026). These approaches concern where and how to acquire evidence during inference. CTV compares the evidence already recorded by completed rollouts; its evaluation here does not establish performance on their high-resolution, document, or geo-localization settings.

Closed-loop perceptual scaling versus fixed-pool selection. TTSP samples perceptual trajectories in rounds, filters uncertain traces using critical-token entropy, and records supported observations and unresolved conflicts in a correctable Evidence Ledger (Jiang et al., 2026). The ledger guides a subset of later trajectories toward new visual inspections, while another subset remains unguided; the final answer is a reliability-weighted vote over retained traces. Its hypotheses concern image regions, whereas DiVE explores candidate entities linked to external knowledge. DiVE samples complete entity-search trajectories independently, then CTV groups their recorded candidate names, tool calls, textual observations, and answers into an entity-evidence forest for selecting an existing answer. Consequently, applying TTSP's final aggregation to DiVE's frozen pool would omit TTSP's feedback-guided exploration.

Reflection, consensus, and learned verification. Self-Refine iteratively revises outputs using model-generated feedback (Madaan et al., 2023), whereas CRITIC obtains external tool feedback before revision (Gou et al., 2024). Universal Self-Consistency uses an LLM to identify agreement among free-form candidates (Chen et al., 2023a). Learned verifiers can instead rank sampled solutions, as demonstrated for math word problems (Cobbe et al., 2021); generative reward models formulate verification through next-token prediction (Zhang et al., 2025). For multimodal reasoning, perception-focused supervision can improve the detection

of grounding errors in vision-language process reward models (Ong et al., 2025). CTV explicitly aligns competing entity hypotheses and their recorded evidence in a joint selection step. ReflectiVA's retrieval/reflection tokens (Cocchi et al., 2025) and MACT's collaborative judgment loop (Yu et al., 2025b) are relevant complementary mechanisms; matched-budget comparisons remain future work.

Visual evidence and document-grounded QA. MRAG-Bench tests settings where retrieved visual information is particularly useful (Hu et al., 2025). ColPali retrieves page images with multi-vector representations (Faysse et al., 2025), while VisRAG and VDocRAG retain visual document information throughout retrieval and generation (Yu et al., 2025a; Tanaka et al., 2025). VisDoM studies multi-document QA with visually rich elements and cross-modal consistency (Suri et al., 2025); REAL-MM-RAG probes realistic retrieval challenges, including table-heavy documents and query rephrasing (Wasserman et al., 2025). OMG-QA further studies open-domain multimodal generative QA with evidence distributed across documents (Nan et al., 2024). These settings motivate broader tests of cross-trajectory verification, but differ from the entity-centric encyclopedic benchmarks evaluated here.

H Additional Experimental Results and Analysis

H.1 Paired Statistical Comparisons on E-VQA

We compare CTV with the three TTS baselines in Table 1 on all 3,750 E-VQA questions for each of the four backbones. Within each backbone, all selectors share the same frozen pool of eight rollouts per question. We use each selector's re-evaluated final-output BEM correctness, matching the main-table scoring protocol, rather than inheriting the selected trajectory's frozen raw BEM label.

We report CTV-minus-baseline accuracy differences, paired bootstrap 95% confidence intervals, and two-sided exact McNemar tests. For Qwen3-VL-8B, the percentile bootstrap uses 50,000 resamples of question pairs, with NumPy seeds 20260925, 20260926, and 20260927 for Self-Consistency, VisualPRM, and FineVerify, respectively. Differences are computed from the original correct counts before rounding: for example, Qwen3.5-9B versus Self-Consistency yields $164/3{,}750 \times 100 = 4.3733\ldots$ percentage points, reported as $+4.37$.

In Tables 6–9, all counts have denominator 3,750; Δ and confidence intervals are in percentage points. Intervals and tabulated p-values are unadjusted for multiple comparisons.

Table 6: Qwen3-VL-8B: CTV BEM is 54.13% (2,030/3,750).

Method	BEM (%) (correct)	Δ	95% CI	Exact p
Self-Consistency	51.36 (1,926)	$+2.77$	$[1.60, 3.95]$	5.17×10^{-6}
VisualPRM	48.37 (1,814)	$+5.76$	$[4.48, 7.01]$	9.25×10^{-19}
FineVerify	50.88 (1,908)	$+3.25$	$[1.97, 4.53]$	7.51×10^{-7}

Table 7: Qwen3-VL-2B: CTV BEM is 49.95% (1,873/3,750).

Method	BEM (%) (correct)	Δ	95% CI	Exact p
Self-Consistency	47.47 (1,780)	$+2.48$	$[1.23, 3.73]$	1.24×10^{-4}
VisualPRM	45.12 (1,692)	$+4.83$	$[3.52, 6.13]$	8.98×10^{-13}
FineVerify	45.49 (1,706)	$+4.45$	$[3.20, 5.73]$	8.72×10^{-12}

Table 8: InternVL3.5-8B: CTV BEM is 51.07% (1,915/3,750).

Method	BEM (%) (correct)	Δ	95% CI	Exact p
Self-Consistency	48.48 (1,818)	$+2.59$	$[1.36, 3.81]$	4.41×10^{-5}
VisualPRM	47.12 (1,767)	$+3.95$	$[2.64, 5.28]$	5.35×10^{-9}
FineVerify	49.52 (1,857)	$+1.55$	$[0.21, 2.85]$	0.0233

Table 9: Qwen3.5-9B: CTV BEM is 55.47% (2,080/3,750).

Method	BEM (%) (correct)	Δ	95% CI	Exact p
Self-Consistency	51.09 (1,916)	$+4.37$	$[3.12, 5.63]$	1.56×10^{-11}
VisualPRM	49.39 (1,852)	$+6.08$	$[4.77, 7.39]$	6.84×10^{-20}
FineVerify	49.55 (1,858)	$+5.92$	$[4.59, 7.23]$	4.81×10^{-18}

Across all four backbones, the 12 paired intervals lie above zero and all exact McNemar tests remain significant after Holm correction over the full 12-comparison family ($p_{\mathrm{adj}} \leq 0.0233$). These results support CTV's overall advantage over all three baselines under the current main-table BEM protocol and fixed candidate pools.

H.2 Detailed Tool Ablation Experiments

We further ablate the visual-grounding and direct entity-page interaction tools. All three configurations use the same Balanced-2K policy checkpoint and the 3,750 E-VQA questions, with eight rollouts per question. The full

toolset provides image search, cropping, reference-image inspection, and direct entity-page text access. The no-crop/source variant removes cropping and reference-image inspection but retains image search and direct entity-page access. The no-direct-access variant retains the visual tools but replaces direct entity-page access with BGE-based text retrieval. The query image remains available in all three configurations. Each rollout is scored with official BEM. Mean rollout BEM is computed over all 30,000 ($3{,}750 \times 8$) trajectories; its displayed sample standard deviation is computed across the eight rollout-index accuracies. Oracle@8 is the fraction of questions with at least one BEM-correct answer among their eight trajectories. Table 10 shows that removing either tool group lowers both metrics relative to the full toolset, with a larger drop when direct entity-page access is replaced.

Table 10: Fine-grained tool ablations on E-VQA. Mean rollout BEM is reported with the sample standard deviation across eight rollout-index accuracies (in percentage points); Oracle@8 is question-level BEM.

Configuration	**Mean rollout BEM (%)**	**Oracle@8 (%)**
Full DiVE	46.58 ± 0.42	72.61
No crop / no image source	44.55 ± 1.46	71.92
No direct entity-page access	41.05 ± 0.52	64.56

H.3 Fine-Grained Verifier Diagnostics

This appendix discusses the five quantities reported in Table 11 and plotted in Figure 6. All diagnostics are computed from the frozen rollout pools and the saved selector decisions; no additional model calls are made. Figure 13 shows the full visualization of all TTS verifiers across all backbones.

The benchmark contains 3,750 questions. The *oracle-hit set* contains the N questions whose frozen pool of eight rollouts includes at least one BEM-correct rollout ($\mathrm{Oracle@8} \geq 1$). N equals 2,631, 2,677, 2,723, and 2,767 for Qwen3-VL-2B, InternVL3.5-8B, Qwen3-VL-8B, and Qwen3.5-9B, respectively. The entity- and path-level quantities below are computed on this set, because a selector cannot succeed on a question whose pool contains no correct candidate; accuracy and ΔAcc are reported on all 3,750 questions.

Token efficiency. Token cost is verifier-stage usage per question, i.e., prompt plus completion tokens averaged over all 3,750 questions, and excludes the rollout-generation cost shared by all methods. For each backbone and sampling budget N, the same generated pool is reused by every selector. Writing total generation-plus-selection token usage for method m as $C^{(m)}_{\mathrm{all}}(N) = C_{\mathrm{gen}}(N) + C^{(m)}_{\mathrm{ver}}(N)$, the common generation term does not change the ranking by token count, but is needed to quantify the percentage of end-to-end token savings. $C^{(m)}_{\mathrm{ver}}$ sums input and output tokens over selection-stage model calls. This comparison is conditional on the shared pool: it neither holds generation cost fixed across different N nor equates token counts with FLOPs or latency. Standalone retrieval, image processing, and tool execution costs are outside this token metric. Generating multiple rollouts is a shared cost of the multi-trajectory TTS methods evaluated here, not an overhead specific to CTV. Comparing against a single-rollout baseline changes both candidate generation and selection, so it does not isolate verifier-stage efficiency.

VisualPRM scores each path separately and emits no text, so its completion tokens are zero and its prompt tokens are reconstructed with its official conversation template, including image tokens. Within each base-model block, token efficiency is the relative inverse cost $\mathrm{Eff}_i = (\min_j (\mathrm{in}_j + \mathrm{out}_j))/(\mathrm{in}_i + \mathrm{out}_i)$, so the cheapest verifier equals 1.00. The main text reports the absolute cost (11.2k–11.7k tokens per question for CTV).

Diagnostic dimensions. For each oracle-hit question we follow the selector's chosen rollout and extract its terminal answer-bearing entity. *Right path*: the selected rollout's final answer is correct. *Right entity*: the trajectory terminates at the gold entity. We denote the size of the oracle-hit question set as N_o. Then we define three simple diagnostic metrics:

- Entity match: $(E{=}1) = \frac{\text{Right entity}}{N_o}$, representing the ability to locate the correct entity.
- End-to-end answer rate: $(A{=}1) = \frac{\text{Right path}}{N_o}$, representing whether the verifier can find the answer.
- Within-entity answer: $(A{=}1 \mid E{=}1)$, representing the ability to arrive at the correct answer given the correct entity hypothesis.

Accuracy is the official BEM accuracy on all 3,750 questions, computed from the selector's final answer, and ΔAcc subtracts the base model's Pass@1 shown in the block header of Table 11.

Normalization of the radar. Each axis of Figure 6 is oriented so that outward is better and uses a fixed display range shared by all four panels, so the polygons show absolute levels rather than per-panel ranks and no verifier

is pinned to the outer ring by construction. The ranges are: Token Efficiency 8,000–35,000 tokens per question (inverted), Entity Match 65–85%, Within-Entity Answer 85–95%, End-to-End Answer 60–80%, and ΔAcc 0–12 points; values outside a range are clipped to its ends. Because the ranges are shared across panels, the four panels are directly comparable, and the radial tick labels denote positions within these fixed ranges.



Figure 13: Radar visualization across all backbones, synchronized with Table 11. Lines connect normalized point estimates as visual guides.

H.4 Cross-Benchmark Generalization

We evaluate whether the test-time scaling behavior of DIVE transfers beyond the source benchmarks used to train the agent. We retain the same source-trained Qwen3-VL-8B policy and the same eight-rollout inference budget, and do not update the weights of either the policy or the verifier on target-task supervision. We consider three substantially different knowledge-intensive VQA benchmarks: ViQuAE, which emphasizes fine-grained named-entity recognition and evaluates short free-form answers with exact match (EM) and token F1, and M3-VQA, which requires parallel or sequential reasoning over multiple entities and evaluates sets of answers with its official multi-answer score. We additionally evaluate on MC-SEARCH, which requires agentic multimodal retrieval over five structured long-chain reasoning topologies and uses normalized token F1. Because these metrics are not commensurate, we report the three datasets separately.

ViQuAE generalization results. On ViQuAE, DIVE already reaches 53.38 EM and 61.15 F1 with a single rollout. CTV raises these scores to 59.82 EM and 68.85 F1, corresponding to gains of 6.44 and 7.70 points over the same frozen agent. The resulting scores are also 9.52 EM and 13.25 F1 points above Wiki-R1 7B, the

Table 11: Fine-grained comparison of TTS verifiers across four base models. VisualPRM input tokens are reconstructed using its conversation template. Token efficiency is the relative inverse verifier cost within each base-model block (cheapest verifier = 1.00). The three metrics under the Oracle-hit Set header are the positive counterparts of the error categories: Entity Match ($E{=}1$) is the share of oracle-hit questions whose selected trajectory terminates at the gold entity; Within-Entity Answer ($A{=}1 \mid E{=}1$) is the share of $E{=}1$ selections whose final answer is also correct; End-to-End Answer ($A{=}1$) is the share of oracle-hit questions answered correctly. A selected answer may be correct even when its entity does not match the gold entity.

Method	Input tokens/q ↓	Output tokens/q ↓	Token eff. ↑	Oracle-hit Set (Oracle@8 ≥ 1)			Acc. ↑	Δ Acc. ↑
				Entity Match ($E{=}1$) ↑	Within-Entity Answer ($A{=}1 \mid E{=}1$) ↑	End-to-End Answer ($A{=}1$) ↑		
Qwen3-VL-2B DiVE Pass@1							40.3	–
VisualPRM	33131.1	0.0	0.34	72.7	88.4	64.3	45.1	+4.8
FineVerify	13143.0	3335.0	0.68	74.0	87.7	64.8	45.5	+5.2
Self-Consistency	11941.4	163.5	0.92	75.8	89.2	67.7	47.5	+7.2
CTV (Ours)	10899.6	273.8	**1.00**	**78.4**	**90.8**	**71.2**	**50.0**	**+9.7**
InternVL3.5-8B DiVE Pass@1							43.4	–
VisualPRM	33653.0	0.0	0.35	74.0	89.2	66.0	47.1	+3.7
FineVerify	12750.1	3125.3	0.74	77.0	90.1	69.4	49.5	+6.1
Self-Consistency	12434.3	164.7	0.93	75.3	90.1	67.9	48.5	+5.1
CTV (Ours)	11408.1	280.9	**1.00**	**77.6**	**92.2**	**71.5**	**51.1**	**+7.7**
Qwen3-VL-8B DiVE Pass@1							46.2	–
VisualPRM	33352.6	0.0	0.34	74.8	88.6	66.2	48.4	+2.2
FineVerify	11852.8	2851.6	0.76	78.6	89.0	69.9	50.9	+4.7
Self-Consistency	12145.2	162.4	0.91	77.9	90.3	70.4	51.4	+5.2
CTV (Ours)	10940.4	281.1	**1.00**	**80.4**	**92.2**	**74.1**	**54.1**	**+7.9**
Qwen3.5-9B DiVE Pass@1							47.8	–
VisualPRM	33640.8	0.0	0.35	75.0	89.2	66.9	49.4	+1.6
FineVerify	10047.1	2324.5	0.95	76.4	87.8	67.1	49.6	+1.8
Self-Consistency	12692.1	223.1	0.91	76.1	90.9	69.2	51.1	+3.3
CTV (Ours)	11361.9	353.6	**1.00**	**80.9**	**92.9**	**75.2**	**55.5**	**+7.7**

Table 12: Generalization to the official 1,257-question ViQuAE test split. Published baselines are taken from the corresponding papers (Lerner et al., 2022; 2024; Adjali et al., 2024; Ning et al., 2026a). Wiki-R1 is trained with reinforcement learning on Encyclopedic-VQA and InfoSeek.

Method	**EM ↑**	**F1 ↑**
ViQuAE few-shot RC (Lerner et al., 2022)	22.07	25.43
DPR_{V+T} (Lerner et al., 2024)	30.90	34.30
MiRAG (Adjali et al., 2024)	36.60	41.20
Wiki-R1 7B (Ning et al., 2026a)	50.30	55.60
DiVE Qwen3-VL 8B (Pass@1)	53.38	61.15
DiVE-CTV Qwen3-VL 8B	**59.82**	**68.85**
DiVE-CTV Qwen3-VL 8B Oracle@8	*73.51*	*80.68*

strongest target-training-free result included in Table 12. This comparison should be interpreted as cross-system context rather than a controlled ablation because the methods use different source training data and retrieval stacks. The controlled comparison is between DiVE Pass@1 and DiVE-CTV, which share the checkpoint, tools, knowledge base, and decoding configuration.

ViQuAE transfer accommodation. ViQuAE rewards a single short entity string, whereas the source agent often produces explanatory answers. We therefore add only a task-format instruction that requests the concise answer; the reasoning prompt, source-trained policy, retrieval policy, and CTV algorithm are not trained or fine-tuned on ViQuAE.

M3-VQA generalization results. On the full M3-VQA benchmark, CTV improves the same frozen agent from 25.74 to 30.67, a gain of 4.93 points. This places DiVE-CTV within 0.53 points of the published Qwen2.5-VL-7B Agentic Retrieval result and above the published InternVL2.5-8B, LLaVA-OneVision-7B, and MiniCPM-V-2.6 Agentic Retrieval results. It does not reach the 38.90 absolute best result, which uses a 72B solver. Moreover, the published Agentic Retrieval system additionally uses GPT-4o as a planner, Qwen2.5-VL-7B for

Table 13: Generalization to the complete M3-VQA (Ma et al., 2026a). We report the official M3-VQA-specific Planner–Executor–Solver baselines. DIVE-CTV uses the same frozen eight-rollout pool as DIVE Pass@1.

Method	M3-VQA score ↑
GPT4o Planner + InternVL2.5-78B	**36.40**
GPT4o Planner + InternVL2.5-8B	29.40
GPT4o Planner + MiniCPM-V-2.6-8B	28.00
DIVE QWEN3-VL-8B (Pass@1)	25.74
DIVE-CTV Qwen3-VL-8B	**30.67**
DiVE-CTV Qwen3-VL-8B Oracle@8	*52.06*

Table 14: Generalization to 3,333-question MC-SEARCH benchmark. Published overall scores are reconstructed from Ning et al. (Ning et al., 2026b) and weighted by the corresponding topology sizes in their Table 2. They are therefore derived totals rather than values quoted directly from the paper.

Method	Overall F1 ↑
GPT-4o-Mini	30.77
Gemini-2.5-Flash	37.81
Gemini-2.5-Pro	40.65
Claude-3.7-Sonnet	31.88
InternVL3.5-8B	35.52
InternVL3.5-8B + SEARCH-ALIGN	38.36
Qwen2.5-VL-7B	25.08
Qwen2.5-VL-7B + SEARCH-ALIGN	40.17
DIVE Qwen3-VL 8B (Pass@1)	38.76
DIVE-CTV Qwen3-VL 8B	**41.21**
DiVE-CTV Qwen3-VL 8B Oracle@8	*49.60*

object detection, BGE-Large-en-v1.5 for text retrieval, and CLIP-ViT-Large for image retrieval; the absolute values are therefore not a like-for-like systems comparison.

M3-VQA transfer accommodation. M3-VQA differs from the source tasks not only in reasoning depth but also in its output contract: an answer is a set of unique atomic values covering all requested entities. We preserve the agent checkpoint, tool policy, eight rollouts, and CTV forest construction, but use a concise answer-set instruction and change the natural-language CTV review rubric to check multi-entity coverage, deduplication, and answer-set format. We also retain our BGE-M3/EVA retrieval stack instead of reproducing the benchmark's GPT-4o planner and dedicated detector stack, which keeps the transfer experiment close to the original DIVE system but limits direct leaderboard comparability.

MC-Search generalization results. We compute the overall score as

$$\mathrm{F1}_{\mathrm{mc-search}} = \frac{1306\,\mathrm{F1}_{\mathrm{IIC}} + 233\,\mathrm{F1}_{\mathrm{MIF}} + 169\,\mathrm{F1}_{\mathrm{TIC}} + 680\,\mathrm{F1}_{\mathrm{PITF}} + 945\,\mathrm{F1}_{\mathrm{TC}}}{3333},$$

where IIC, MIF, TIC, PITF, and TC denote *Image-Initiated Chain*, *Multi-Images Fork*, *Text-Initiated Chain*, *Parallel Image-Text Fork*, and *Text-Only Chain*, respectively. This sample-count-weighted aggregation differs from an unweighted mean over the five topologies. On MC-SEARCH, CTV improves the same frozen agent from 38.76 to 41.21 F1, a gain of 2.46 points, and is 0.56 points above the strongest reconstructed published total, Gemini-2.5-Pro at 40.65. The reviewer ranks branches using their recorded entity and evidence paths. The prompt also preserves the dependency chain stated in the question, treats retrieval misses and unavailable images as unknown rather than false, checks every requested answer slot and its relation owner, and favors complete natural-language multi-part answers. The comparison with published systems is not like-for-like because their backbones, retrieval stacks, and top-1 retrieval protocol differ.

Generalization takeaway. Across all three target tasks, CTV improves the frozen Pass@1 agent despite changes in entity granularity, reasoning structure, retrieval corpus, answer format, and evaluation metric. The gain is 6.44 EM and 7.70 F1 points on ViQuAE and 4.93 points on M3-VQA, while MC-SEARCH improves by 2.46 F1 points. The learned agent and number of rollouts remain unchanged. ViQuAE and M3-VQA require only knowledge-index and output-contract accommodations, whereas MC-SEARCH v2 uses a disclosed task-specific ranking prompt without consensus pre-filtering or an additional model call. These results support the

Table 15: Source of the entity root for every frozen trajectory. Percentages use the trajectory count in the second column as denominator. "Assigned" means a nonempty entity root.

Benchmark	Trajectories	From Answer	From Tool Call	Assigned	Ungrounded
E-VQA Single	30,000	16,527 (55.09%)	13,194 (43.98%)	29,721 (**99.07%**)	279 (0.93%)
InfoSeek	570,680	242,307 (42.46%)	316,822 (55.52%)	559,129 (**97.98%**)	11,551 (2.02%)

claim that test-time selection gains are not tied to a single dataset or scorer, while the disclosed adaptations avoid overstating the experiments as uniformly zero-shot.

H.5 Coverage of Terminal Entity Assignment

CTV assigns each frozen trajectory an entity root by first checking whether its final answer explicitly names an entity encountered along that trajectory. Otherwise, it uses the last text-search entity, or the last visual entity state if no text-search entity is available. A trajectory with neither source is marked ungrounded. We apply this unchanged production rule to all eight trajectories per question on E-VQA Single and InfoSeek; the unit of analysis is a trajectory rather than a question.

CTV obtains an entity root for 99.07% of E-VQA trajectories and 97.98% of InfoSeek trajectories (approximately 98.0%). This coverage shows that the entity-based forest can organize nearly all sampled paths under an explicit hypothesis, with only a small ungrounded remainder. Assignment coverage measures the availability of an entity label, not the correctness of visual identification.

I Case Study

Figure 14 illustrates the complete inference process of a DiVE agent. The query is: "In which part of the world does this animal live?" DiVE first retrieves ten candidate entities using an image retrieval tool. During its reasoning phase, it notes that *"many starfish look alike"*; consequently, it compares the entities' reference images with the query image one by one, starting with the top candidate. After examining the candidate images, the agent identifies discrepancies and invokes the same tool four times. Ultimately, it identifies the entity as ***Patiriella regularis*** and obtains the answer by directly searching for query-related terms within that entity's knowledge base. This example demonstrates the respective roles of visual grounding and direct interaction. Without visual comparison, the agent could have selected another starfish entity page containing plausible habitat information, producing a typical "grounding trap."

Entity-Evidence Forest example

Question: {When is the mating period?}

===== ENTITY_ROOT 1=**Smooth earth snake** =====
PATHS=A,C VISITS=2
N1 target=image[0]; ACTION: image_search_tool {"image_index": 0}
G3 target=Smooth earth → ACTON: image_search_tool → representative image
ACTION: text_search_tool → OBSERVATION: entity: Smooth earth snake: ···
--- R1.B1 PATH=A ---
ANSWER_LEAF A: The mating period for the depicted snake···
--- R1.B2 PATH=C ---
ANSWER_LEAF C: The mating period for the Smooth earth snake is not explicitly documented in the available data···

===== ENTITY_ROOT 2=**Rhinophis goweri** =====
PATHS=B VISITS=1
N1 target=image[0]; ACTION: image_search_tool {"image_index": 0}
G1 target=Rhinophis goweri → ACTON: image_search_tool → representative image
--- R2.B1 PATH=B ---
ANSWER_LEAF B: The mating period for Rhinophis goweri (the Gower's earth snake) is not known···

===== ENTITY_ROOT 3=**Short-tailed snake** =====
PATHS=D VISITS=1
N1 target=image[0]; ACTION: image_search_tool {"image_index": 0}
E3 target=Short-tailed snake → ACTION: text_search_tool → OBSERVATION: related species and unrelated entries ···
--- R3.B1 PATH=D ---
ANSWER_LEAF D: The mating period for the depicted snake species is likely during late spring···

===== ENTITY_ROOT 4=Carphophis amoenus =====
PATHS=E VISITS=1
N1 target=image[0] status=OBSERVED; ACTION: image_search_tool {"image_index": 0}
N2 target=image[0] → ACTION: image_crop_tool {"bbox_2d": [300, 450, 700, 850]} → cropped image (image_index=4)
N3 target=image[4] → ACTION: image_search_tool {"image_index": 4} → OBSERVATION: ··· 4. Carphophis amoenus ···
G4 target=Carphophis amoenus → ACTON: image_search_tool → representative image
ACTION: text_search_tool → OBSERVATION: ···Reproduction: Courtship and mating probably occur in the spring; the sexes are most often found together between late April and June···
--- MCT4.B1 PATH=E ---
ANSWER_LEAF E: The mating period for the Leatherback snake (Carphophis amoenus) occurs in the spring, typically between late April and June.

===== ENTITY_ROOT 5=Pseudorabdion mcnamarae =====
PATHS=F VISITS=1
N1 target=image[0]; ACTION: image_search_tool {"image_index": 0}
G7 target=Pseudorabdion mcnamarae → ACTON: image_search_tool → representative image
ACTION: text_search_tool → OBSERVATION: null
--- R5.B1 PATH=F ---
ANSWER_LEAF F: The mating period of Pseudorabdion mcnamarae is not known; there is no available scientific documentation specifying its breeding season.

===== ENTITY_ROOT 6=Prong-snouted blind snake =====
PATHS=G VISITS=1
N1 target=image[0]; ACTION: image_search_tool {"image_index": 0}
G1 target=Prong-snouted blind snake → ACTON: image_search_tool → representative image
E1 target=Prong-snouted blind snake → ACTION: text_search_tool → OBSERVATION: null
E2 target={Prong-snouted blind snake | Typhlopidae | Anilloidae} → ACTION: text_search_tool → OBSERVATION: null
--- R6.B1 PATH=G ---
ANSWER_LEAF G: The mating period of the Prong-snouted blind snake (Reptiloechis pictus) is not documented···

===== ENTITY_ROOT 7=Pink-tailed worm-lizard =====
PATHS=H VISITS=1
N1 target=image[0] status=OBSERVED; ACTION: image_search_tool {"image_index": 0}
G1 target=Pink-tailed worm-lizard→ representative image check
E1 target={Pink-tailed worm-lizard | Prong-snouted blind snake | Rufous beaked snake} → matches: 0
E2 target={Rhinophis goweri | Smooth earth snake} → matches: 0
--- R7.B1 PATH=H ---
ANSWER_LEAF H: The mating period for the Pink-tailed worm-lizard (a closely related species) is not documented···

VALID ANSWER LEAVES TO RANK: A, B, C, D, E, F, G, H

Verifier OUTPUT:
"ranked_labels": ["E", "A", "D", "C", "F", "G", "H", "B"]

Figure 11: Entity-evidence forest example.

Agent-Tool Interaction Example

Query
In which part of the world does this animal live?



ASSISTANT
<thinking> I need to identify the exact species of this blue starfish-like animal first, because its geographic distribution will depend on its identity. The image shows **a five-armed, blue, spotted echinoderm** on a rock in shallow water — likely a starfish or sea star. I'll start with an image search to find matching entities. </thinking>
<tool_call>
{"name": "image_search_tool", "arguments": {"image_index": 0}}
</tool_call>

TOOL OBSERVATION
<tool_response>1. Asterina (starfish) 2. Parvulastra exigua 3. Ceramaster
4. Patiriella regularis 5. Patiriella 6. Astropecten articulatus···</tool_response>

ASSISTANT
<thinking> The top candidate is "Asterina (starfish)" — this is a genus of small, often colorful starfish commonly called "blazing starfish" or "blue starfish," sometimes with yellow spots, matching the image. But I must verify: **many starfish look similar**. I will visually confirm by loading the representative image of "Asterina (starfish)" to compare with the depicted animal. </thinking>
<tool_call>
{"name": "image_source_tool", "arguments": {"entity": "Asterina (starfish)"}}
</tool_call>

TOOL OBSERVATION
<tool_response>
Here is the representative image for Asterina (starfish):
</tool_response>



ASSISTANT
<thinking> The representative image for "Asterina (starfish)" shows small, **reddish-orange, five-armed starfish — not matching the blue, spotted animal in the original image**. This suggests "Asterina (starfish)" is not the correct entity. I will now inspect the next candidate, "Parvulastra exigua," which also appears visually plausible from the list. </thinking>
<tool_call>
{"name": "image_source_tool", "arguments": {"entity": "Parvulastra exigua"}}
</tool_call>

TOOL OBSERVATION
<tool_response>
Here is the representative image for Parvulastra exigua:
</tool_response>



ASSISTANT
<thinking> The representative image of "Parvulastra exigua" shows **a small, greenish-brown starfish on a rock — not matching the blue, spotted animal in the original image**. I will now inspect "Ceramaster," which is another candidate from the image search and may better match the color and pattern.
</thinking>
<tool_call>
{"name": "image_source_tool", "arguments": {"entity": "Ceramaster"}}
</tool_call>

TOOL OBSERVATION
<tool_response>
Here is the representative image for Ceramaster:
</tool_response>



ASSISTANT
<thinking> The representative image of "Ceramaster" shows **a pinkish-orange starfish with a textured surface — still not matching the blue, spotted starfish in the original image.** I will now inspect "Patiriella regularis," which is known for its vibrant colors and may be a better match. </thinking>
<tool_call>
{"name": "image_source_tool", "arguments": {"entity": "Patiriella regularis"}}
</tool_call>

TOOL OBSERVATION
<tool_response>
Here is the representative image for Patiriella regularis:
</tool_response>



ASSISTANT
<thinking> The representative image of "Patiriella regularis" **shows a dark blue, five-armed starfish with lighter spots — this closely matches the color and pattern of the animal in the original image**. This is a strong candidate. I will now search for geographic distribution information about "Patiriella regularis" to answer the question. </thinking>
<tool_call>
{"name": "text_search_tool", "arguments": {"query": "distribution|range|habitat", "entities": ["Patiriella regularis"]}}
</tool_call>

TOOL OBSERVATION
<tool_response>
Local KB text search results (mode=entity):
entity: Patiriella regularis matches: 1
native to **New Zealand**. It has an arm spread of up to 60 mm (2.4 in). It is one of New Zealand's most common rocky shore starfish. It is abundant in shallow water around both mainland islands. They come in a large range of colours and have 4–6 arms.

ASSISTANT
<thinking> The text search confirms that Patiriella regularis is native to New Zealand, specifically abundant in shallow water around both main islands. It is one of New Zealand's most common rocky shore starfish and occupies intertidal zones. While broader regional context (like southern Australia or Chile) was mentioned under the genus Patiriella, the species-level evidence points decisively to New Zealand. Thus, I can now confidently answer the question. </thinking>
<answer> **This animal, Patiriella regularis, lives primarily in New Zealand**, where it is commonly found on rocky shores in shallow, intertidal waters around both the North and South Islands. </answer>

Figure 14: A case study on an E-VQA question.