REMAP: RESTORING THE PERCEPTUAL CYCLE WITH REASONING-TIME LATENT VISUAL MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

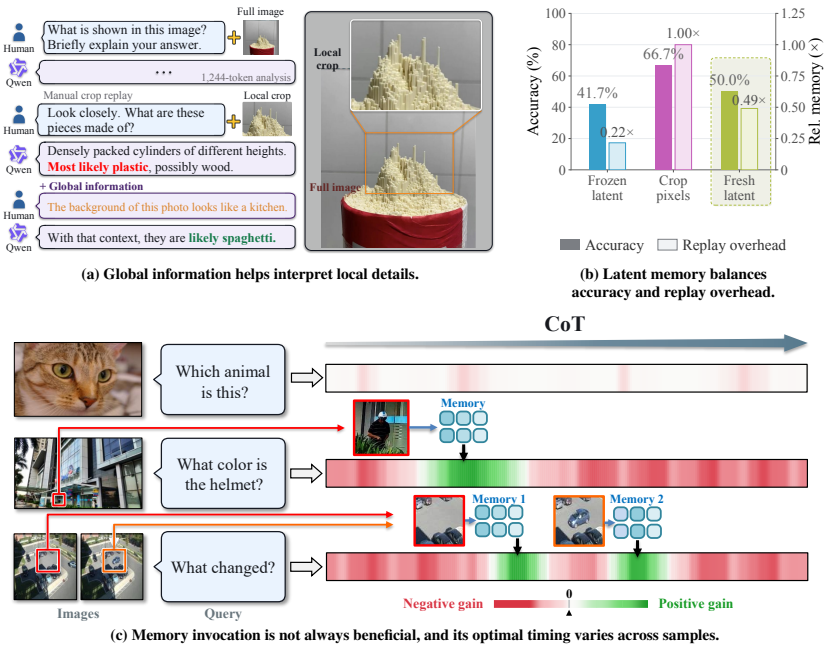
As multimodal large language models (MLLMs) reason for longer, attention to the initial visual input diminishes, weakening visual grounding. Visual memory reintroduces visual evidence during reasoning. We conduct a controlled analysis of visual memory along three axes: curation, organization, and access. We find that local evidence benefits from global context, compact latent representations balance accuracy and visual-context cost, and the utility of memory access depends on the reasoning state. Guided by these findings, we propose **ReMAP** (**Re**asoning-Time **M**emory-**A**ugmented **P**erception), which couples two complementary latent memories: a static, question-conditioned Global memory that preserves scene and cross-image context, and a dynamic Local memory that uses this context as an anchor while selecting and re-encoding region-level evidence according to the current reasoning state and the question. Both memories return compact latent tokens that are inserted into the reasoning sequence, and a reinforcement-learning access policy trained with branched rollouts decides when to continue reasoning or invoke Global or Local memory. On ten benchmark families, ReMAP outperforms prior visual-memory methods on all four multi-image benchmarks, exceeding the strongest prior results on MuirBench and MIMIC by 8.38 and 14.84 percentage points, respectively. Across four backbone families, enabling memory access improves over the same trained model with memory disabled, and on shared V*Bench, CV-Bench-2D, and MuirBench questions ReMAP reduces the visual tokens entering the reasoning sequence by 51.0–76.8% relative to the native-resolution backbone. Further analyses show that Global and Local memory form distinct yet complementary latent representations and play coordinated roles in interleaved vision-language reasoning. Together, these components restore the perceptual cycle by letting the reasoning state trigger targeted visual retrieval, with the retrieved evidence guiding subsequent reasoning.

1 INTRODUCTION

Vision–language models are increasingly applied to visual tasks covering larger visual contexts and longer reasoning trajectories. These tasks require models to draw on visual evidence at different reasoning stages such as inspecting fine-grained details, comparing observations across images, or verifying intermediate conclusions. As visual inputs and generated text accumulate, maintaining access to relevant visual evidence is increasingly challenging. Recent studies report visual forgetting, where attention to visual inputs diminishes during extended generation (Sun et al., 2025; Liu et al., 2025), highlighting the difficulty of sustaining visual grounding throughout reasoning.

Visual memory methods have emerged as a natural solution to retain visual evidence during reasoning. Existing approaches explore several routes with distinct trade-offs: reinspecting source pixels offers fine-grained details but requires extra image processing (Zheng et al., 2026; Su et al., 2025; Jiang et al., 2025), replaying encoded features reuses existing representations but limits available evidence to their initial coverage (Chu et al., 2025; Gao et al., 2025; Zhang et al., 2026a), constructing latent memories provides compact representations at the cost of interpretability (Zhang et al., 2026b; Chen et al., 2026c; Yu et al., 2026). These approaches differ in evidence selection, representation, and delivery timing. We therefore ask: **what design choices of a visual memory system truly improve visual reasoning?**

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.



(a) Global information helps interpret local details.

(b) Latent memory balances accuracy and replay overhead.

(c) Memory invocation is not always beneficial, and its optimal timing varies across samples.

Figure 1: **Investigating memory curation, organization, and access.** **(a)** An interaction with Qwen2.5-VL-7B-Instruct on a MuirBench example illustrates how global context helps resolve ambiguous local evidence. **(b)** A comparison of frozen latents, crop pixels, and fresh latents on a random 25% subset of V*Bench examines the trade-off between accuracy and incremental GPU-memory overhead. **(c)** An easy recognition example and cases from V*Bench and MuirBench show how the need for memory and its invocation timing vary across queries and reasoning states.

We characterize visual memory along three design axes: **curation** (what evidence enters memory), **organization** (how evidence is represented), and **access** (when to access memory and which memory to use). Under this framework, we conduct preliminary experiments combining qualitative analysis with controlled empirical evaluations, as illustrated in Figure 1. These analyses yield three key findings: ❶ Memory **curated** from local visual evidence alone can be insufficient for contextual reasoning, as illustrated in Figure 1(a). ❷ Memory **organization** affects both reasoning accuracy and computational overhead, as illustrated by the trade-off between image-crop replay and compact latent representations in Figure 1(b). ❸ Reintroducing visual evidence is not always beneficial, and its utility depends on when it is **access**ed during reasoning (Figure 1(c)).

Guided by these findings, we propose **ReMAP** (*Reasoning-Time Memory-Augmented Perception*). For **memory curation**, ReMAP combines Global memory, which preserves scene and cross-image context from the input images and question, with Local memory, which selects and re-encodes image regions according to the current reasoning state. For **memory organization**, Global and Local memory return compact latent tokens that are inserted into the reasoning sequence to balance reasoning accuracy and computational overhead. For **memory access**, we treat memory invocation as an agentic tool call so that the model can obtain relevant visual evidence when needed. We leverage branched rollouts before each invocation to learn whether, when, and which memory to invoke. These rollouts compare the outcomes of requesting memory versus continuing from the same reasoning state and teach when memory helps. Inference follows a single trajectory.

We evaluate ReMAP against existing visual memory methods on ten benchmark families spanning single-image perception, multi-image reasoning, and broader visual understanding. ReMAP achieves state-of-the-art overall performance across the benchmark suite. Notably, ReMAP significantly improves multi-image reasoning, outperforming existing visual-memory methods on all four benchmarks and exceeding the strongest prior results on MuirBench and MIMIC by 8.38 and 14.84 percentage points, respectively. Beyond these aggregate gains, our analyses characterize reasoning-time visual memory: ReMAP's memory access improves performance across four back-

bone families, Global and Local evidence provide complementary benefits, and compact latent returns reduce visual-context cost while maintaining leading accuracy. Further analyses make Global and Local memory interpretable through their distinct yet complementary latent representations and coordinated roles in interleaved vision-language reasoning. Our contributions are threefold:

- ❶ **We identify three principles for reasoning-time visual memory:** contextualize local evidence with global context, compress representations, and adapt access to the evolving reasoning state.
- ❷ **We introduce ReMAP, a framework for reasoning-time visual memory.** ReMAP combines query-conditioned Global context with state-guided Local evidence in compact latent returns and learns when and which memory to invoke through branched rollouts.
- ❸ **We demonstrate strong multi-image reasoning with interpretable memory roles.** ReMAP outperforms prior visual-memory methods on all four multi-image benchmarks, while representation and trajectory analyses reveal the complementary functions of Global and Local memory.

2 Related Work

Memory curation spans pixel reacquisition, feature replay, and learned memory formation, with distinct trade-offs between evidence coverage and processing overhead (Zheng et al., 2026; Su et al., 2025; Zhang et al., 2026b). Memory organization represents evidence through images, text, encoded features, or latent tokens, with different trade-offs in interpretability and context cost (Hu et al., 2024; Zhang et al., 2026d; Huang et al., 2026; Yu et al., 2026). Memory access uses heuristic triggers, uncertainty criteria, or learned policies (Sun et al., 2025; Liu et al., 2026b; Peng et al., 2026), and related policy optimization methods improve tool use and visual reasoning through training feedback (Dong et al., 2026a; Li et al., 2026c). ReMAP bridges these dimensions by combining pixel reacquisition with compact latent memory and preserving scene and cross-image context to interpret newly acquired local evidence. Its access training branches before memory invocation to compare requesting memory with continuing from the same reasoning state, providing feedback on the utility of memory access. A detailed review and comparison are provided in Appendix A.1.

3 Reasoning-Time Memory-Augmented Perception

In this section, we introduce ReMAP, which interleaves autoregressive reasoning with access to latent visual memory to restore the perceptual cycle. Figure 2 provides an overview of the framework. Given images $\mathcal{I} = \{I_i\}_{i=1}^{N}$ and a question Q, the model begins reasoning from a coarse visual prefix $V^c = E_c(\mathcal{I})$. During reasoning, the model can continue generating text or invoke Global or Local latent memory. After each memory return, the model resumes reasoning with the retrieved evidence. Specifically, Section 3.1 describes how Global and Local memories are formed and integrated into the reasoning sequence. Section 3.2 describes memory access and its training policy.

3.1 Memory Curation and Organization

Global memory. Global memory provides a compact representation of image content and, for multi-image inputs, relationships across images. We construct it in two stages: the Image Weaver summarizes each image, and the Relation Weaver integrates these summaries into relational context. Both stages are conditioned on the input question.

Let V_i denote the encoded visual features of image I_i. We use two sets of learnable query tokens, Q_G and Q_R, to aggregate per-image content and relational context, respectively. The corresponding modules, W_{img} and W_{rel}, apply cross-attention followed by feed-forward layers. The Image Weaver attends to each image's features separately, while the Relation Weaver attends to the concatenated image summaries. Both include an encoded representation of the question in their attention context, denoted by the conditioning on Q below:

$$Z_i^{\text{img}} = W_{\text{img}}(Q_G, V_i; Q), \qquad \mathcal{Z}^{\text{rel}} = W_{\text{rel}}\big(Q_R, [Z_1^{\text{img}}; \cdots; Z_N^{\text{img}}]; Q\big). \tag{1}$$

The Global memory block $Z_G = [Z_1^{\text{img}}; \cdots; Z_N^{\text{img}}; \mathcal{Z}^{\text{rel}}]$ concatenates the image summaries and relational context. Constructed and cached before decoding, Z_G depends on the input images and

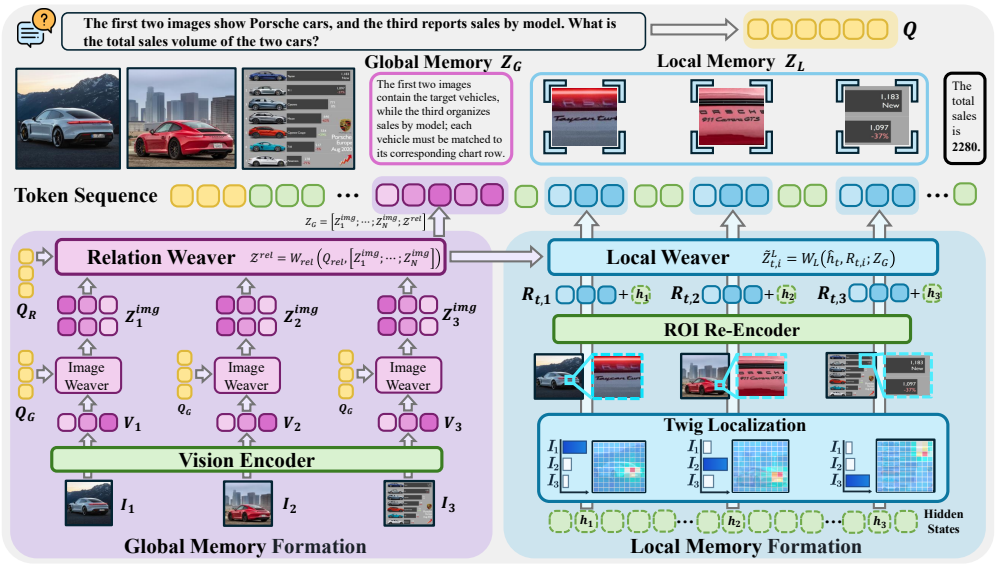


Figure 2: **Overview of ReMAP.** Global memory forms a question-conditioned summary of individual images and their relations before decoding. During reasoning, a Local request uses the current state to select sources and regions, re-encodes selected pixels, and compresses them with cached Global context. Global and Local returns are inserted as continuous latent blocks after their request markers, followed by end markers before autoregressive decoding resumes.

question and remains fixed throughout reasoning. When the decoder generates <mem_g>, the runtime inserts Z_G followed by <mem_g_end> and resumes reasoning.

Local memory. Figure 3(a) shows how Local memory dynamically retrieves fine-grained evidence as reasoning evolves. At state h_t, the Source Selector uses the layer-12 request state h_t^{12} to select images $\mathcal{S}_t$, and Twig uses the layer-18 prefix states $H_{<t}^{18}$ to localize a crop in each selected image. Twig extends the ROI predictor of SD-RPN (Shi et al., 2025) and Q-Zoom (Shi et al., 2026) from the input question to the current reasoning prefix, so successive requests can localize different regions. A frozen ROI encoder maps the crops to region features $R_{t,i}$. The Local Weaver takes the cached Global memory Z_G as an additional input while compressing each region:

$$\widetilde{Z}_{t,i}^{L} = W_L(h_t, R_{t,i}; Z_G), \qquad Z_L = \left[\widetilde{Z}_{t,i}^{L}\right]_{i \in \mathcal{S}_t}. \tag{2}$$

Z_G anchors each crop in the image-level and cross-image context that local evidence alone can miss, consistent with our first finding. A <mem_l> request similarly inserts Z_L followed by <mem_l_end> before generation resumes. The concatenated block contains K Local latents.

3.2 Memory Access

Latent memory keeps retrieved visual evidence within a fixed token budget. Replaying whole images at every request would repeatedly expand the context and compete with the ongoing reasoning sequence. Moreover, memory does not help at every step, since its value depends on when it is accessed, consistent with our third finding. The content of a return can be supervised directly, whereas the value of its timing appears only in later reasoning. We therefore separate memory formation from memory access. Stage I uses supervised fine-tuning to form useful memory and warm up request handling, while Stage II uses reinforcement learning to learn whether and when to invoke Global or Local memory.

Stage I: Supervised learning and warm-up. Stage I isolates memory formation from access-policy learning, so that each return localizes the right region and carries evidence that supports subsequent reasoning. The vision encoder, pretrained LLM, and decoder LoRA remain frozen throughout this stage. Existing datasets lack ROI annotations that change along the reasoning chain, so we

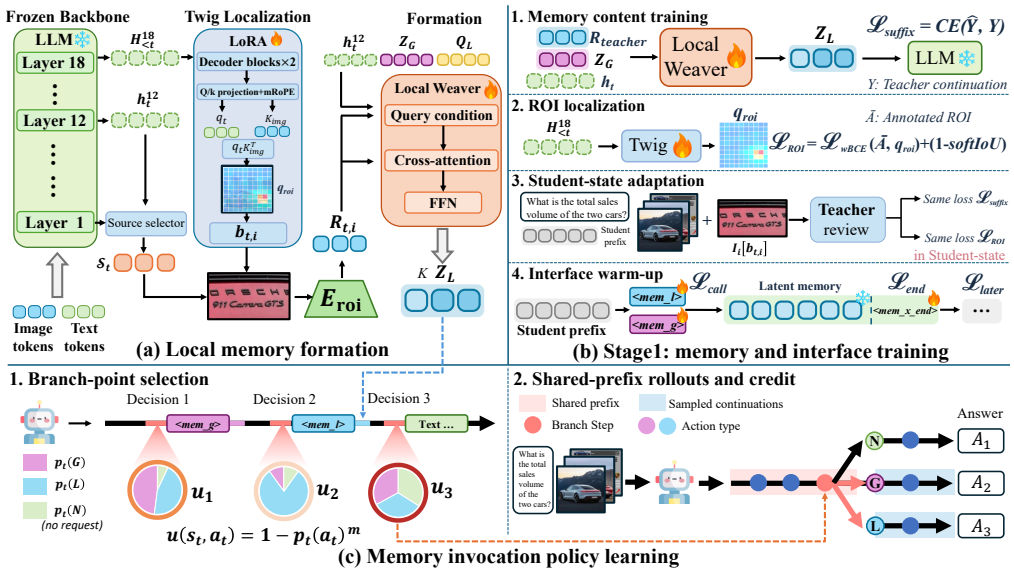


Figure 3: **Local memory and adaptive invocation.** (a) The request state h_t^{12} selects source images $\mathcal{S}_t$, Twig uses the prefix states $H_{<t}^{18}$ to localize a crop in each, and the Local Weaver compresses the crops with the Global anchor Z_G into K Local latents. (b) Stage I trains the Local Weaver with $\mathcal{L}_{\text{suffix}}$ and Twig with $\mathcal{L}_{\text{ROI}}$, adapts both to teacher-reviewed student states, and warms up the memory tokens. (c) Each root branches where siblings are most likely to take a different route, and the invocation gain compares requesting memory with the default continuation N at the same state.

annotate reasoning steps with target regions, together with reference CoTs from a teacher VLM, per-image facts, cross-image relations, and answers. We first optimize memory quality by training the Image and Relation Weavers and then the Local Weaver. Their returns are inserted into teacher CoTs, and the continuation cross-entropy $\mathcal{L}_{\text{suffix}}$ through the frozen LLM, together with fact, relation, and answer supervision, trains them to encode evidence for the teacher's subsequent reasoning. We then train Twig to localize the annotated regions with $\mathcal{L}_{\text{ROI}}$. Because student states differ from teacher states, we next train continuation on student rollouts, where the teacher reviews the student's prefixes and crops to provide continuation and ROI targets for the same two losses. Finally, with the formation modules fixed, a warm-up lets the model emit the memory special tokens, updating only the input embeddings of the four markers and the output-layer rows that score <mem_g> and <mem_l>. Figure 3(b) illustrates these signals, and Appendix A.3.3 gives the objectives.

Stage II: Reinforcement learning for memory access. Guided by our third finding, Stage II makes memory access helpful by learning when and which memory to request, with formation modules frozen. At a request boundary, the decoder emits a Global or Local request token or continues generating text. A trajectory's final reward cannot show whether a given request helped. We therefore branch from a shared prefix and compare requesting with continuing at the same state. Following ARPO (Dong et al., 2026a), each root branches at most once, at the state maximizing

$$u(s_t, a_t) = 1 - p_t(a_t)^m, \tag{3}$$

where $p_t(a_t)$ is the probability of the realized route a_t, with $p_t(\mathrm{N}) = 1 - p_t(\mathrm{G}) - p_t(\mathrm{L})$ for no request, and m is the number of sibling rollouts. This score is the probability that at least one sibling takes a different route, so the branch goes where siblings likely diverge from the root, as illustrated in Figure 3(c). A sibling i that requests memory at branch b receives the invocation gain $\Delta_i^{\text{call}} = R_i - B_b$, where R_i is its final-answer reward and B_b is the mean reward of siblings that continue without a request. These siblings may still request memory later, so the gain measures the value of requesting now. Normalized invocation gains update the request-token embeddings and output rows, and group-relative final-answer advantages (Shao et al., 2024b) update the decoder LoRA. Appendix A.3.4 gives the objectives.

Table 1: **Comparison with existing visual memory methods across ten benchmark families.** Results cover single-image perception, multi-image reasoning, and broader visual understanding. MIMIC Avg. is the mean across four subsets. The top-2 results in each column are highlighted as first and second, with the best results also shown in **bold**.

	Single-image evidence acquisition					Multi-image relational reasoning				Held-out generalization		
Method	V*	CV-2D	CV-3D	HR-4K	HR-8K	BLINK	Muir	MMSI	MIMIC Avg.	RWQA	MathVista	MMVet
Native	76.44	74.41	76.33	72.88	67.75	56.62	57.85	28.10	45.17	67.97	59.90	46.06
Coarse	67.54	73.10	75.42	61.13	54.62	57.79	58.15	28.40	43.56	67.19	59.50	47.06
Pixel reinspection												
Q-Zoom	79.06	76.35	74.08	73.00	66.63	53.97	50.38	25.40	42.29	70.85	54.60	47.43
DeepEyes	**84.29**	78.61	79.42	74.63	68.75	49.24	48.50	24.10	39.27	69.67	62.00	51.01
Pixel Reasoner	81.15	75.00	81.58	73.50	65.38	58.50	53.15	27.90	44.83	**71.90**	71.40	48.07
ReGround	73.30	73.73	77.00	68.13	60.63	54.65	56.12	25.40	28.01	66.93	69.30	43.21
Encoded-feature reuse												
ICoT	56.02	68.22	33.67	66.13	65.25	24.02	20.96	26.30	29.91	35.95	27.40	36.79
MINT-CoT	41.36	72.84	74.83	54.00	43.00	49.79	54.04	26.80	34.05	62.48	62.10	47.98
Latent representations												
DMLR	44.50	68.52	66.75	46.00	42.38	42.32	39.81	22.00	23.74	54.90	51.40	35.87
VaLR	70.68	74.79	71.75	51.00	47.88	44.88	45.08	19.50	17.93	45.75	49.30	33.21
V-Reflection	79.06	75.21	77.75	73.25	66.50	54.21	46.65	25.10	53.16	70.85	68.70	37.94
VisMem	60.21	76.64	82.50	56.13	47.88	56.95	57.00	21.70	41.19	68.10	61.10	45.55
SFT	58.64	75.04	75.58	56.62	48.25	54.13	55.73	29.00	52.70	70.59	72.10	55.05
Stage I_{sft}	76.44	77.96	83.33	75.00	68.38	58.92	57.81	27.30	**68.37**	71.24	70.10	48.35
Ours	82.20	**79.28**	**83.92**	**75.50**	**69.13**	**60.65**	**65.38**	**29.30**	68.00	71.76	**73.40**	**64.27**

4 EXPERIMENTS

We evaluate ReMAP on single-image perception, multi-image reasoning, and broader visual understanding. We then examine the contributions of memory curation, organization, and access through ablations, latent visualizations, reasoning trajectories, and visual-context cost.

4.1 EXPERIMENTAL SETUP

Benchmarks. We use ten benchmark families in three categories. **Single-image perception** includes V*Bench (Wu & Xie, 2024), CV-Bench-2D/3D (Tong et al., 2024), and HR-Bench-4K/8K (Wang et al., 2025b). **Multi-image reasoning** includes BLINK (Fu et al., 2024), Muir-Bench (Wang et al., 2025a), MMSI-Bench (Yang et al., 2026a), and MIMIC (Das et al., 2026). **Broader visual understanding** (Yu et al., 2024; Lu et al., 2024; xAI, 2024) uses MM-Vet, MathVista, and RealWorldQA for generalization tests.

Baselines and implementation. Following the approaches discussed in the introduction, we compare **pixel reinspection** methods Q-Zoom (Shi et al., 2026), DeepEyes (Zheng et al., 2026), Pixel Reasoner (Su et al., 2025), and ReGround (Peng et al., 2026), **encoded-feature reuse** methods ICoT (Gao et al., 2025) and MINT-CoT (Chen et al., 2025), and **latent-representation** methods DMLR (Liu et al., 2026a), VaLR (Jeon et al., 2026), V-Reflection (Zhou et al., 2026), and VisMem (Yu et al., 2026). Native and Coarse provide backbone controls at native resolution and at most 1,024 initial visual tokens per image. ReMAP uses the Coarse input budget and returns 32 latents per Local call. All ReMAP rows and the Native and Coarse controls in Table 1 use Qwen2.5-VL-7B-Instruct (Bai et al., 2025); prior methods keep their released backbones. ReMAP is trained primarily on 16 NVIDIA A800 GPUs. Evaluation protocols, baseline configurations, and training details appear in Appendices A.6, A.4, and A.5.

4.2 OVERALL PERFORMANCE

❶ ReMAP outperforms prior methods on all four multi-image benchmarks. In Table 1, ReMAP surpasses the best prior results on MuirBench and MIMIC by 8.38 and 14.84 percentage points, respectively. These gains span cross-image comparison, spatial reasoning, and evidence aggregation.

Table 2: **Memory access across backbone families.** Base denotes the original backbone, and OFF denotes the trained ReMAP model with memory invocation disabled. Δ reports ReMAP−Base differences in score points. Evaluation details appear in Appendix A.6.

Backbone	Method	V*	CV-3D	HR-8K	BLINK	MuirBench	MMSI	MIMIC Avg.
Qwen2.5-VL-3B	Base	51.83	50.83	42.12	43.61	50.08	29.40	28.21
	OFF	56.02	70.58	49.75	46.66	50.77	28.70	36.25
	ReMAP	58.64	75.58	51.25	48.45	55.23	30.50	50.11
	Δ	↑ +6.81	↑ +24.75	↑ +9.12	↑ +4.84	↑ +5.15	↑ +1.10	↑ +21.90
LLaVA-OneVision-7B	Base	70.68	45.08	56.62	44.66	41.85	28.90	44.05
	OFF	71.73	70.17	56.25	51.60	41.65	27.50	52.79
	ReMAP	76.44	76.00	59.25	53.45	48.65	29.70	59.52
	Δ	↑ +5.76	↑ +30.92	↑ +2.62	↑ +8.78	↑ +6.81	↑ +0.80	↑ +15.47
Qwen3-VL-8B	Base	62.30	92.25	59.00	64.39	73.77	27.90	56.99
	OFF	63.87	91.50	57.12	65.33	72.69	31.60	58.70
	ReMAP	66.49	93.83	60.00	67.07	74.31	32.30	74.20
	Δ	↑ +4.19	↑ +1.58	↑ +1.00	↑ +2.68	↑ +0.54	↑ +4.40	↑ +17.21
InternVL3-8B	Base	59.16	68.67	62.88	53.34	51.81	27.30	36.94
	OFF	61.26	68.33	63.12	53.81	51.42	29.20	43.00
	ReMAP	65.45	69.25	64.12	55.44	52.88	30.30	57.00
	Δ	↑ +6.28	↑ +0.58	↑ +1.25	↑ +2.10	↑ +1.08	↑ +3.00	↑ +20.06

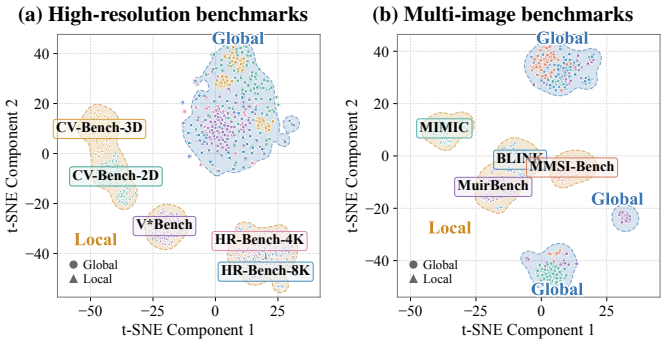


Figure 4: **Global and Local memory representations.** (a) High-resolution and (b) multi-image benchmark groups. Joint t-SNE projections show mean-pooled Global and Local memory vectors; colors denote benchmark groups, circles and triangles denote Global and Local, respectively, and shaded contours show density.

ReMAP also improves over the native-resolution backbone on all five single-image metrics, extending its benefits to fine-grained visual perception.

❷ ReMAP's learned memory access transfers across backbone families. We isolate the contribution of reasoning-time memory invocation by comparing each backbone's original Base model, its trained Memory-OFF counterpart, and ReMAP with memory enabled (Table 2). ReMAP improves over Memory-OFF on both single-image and multi-image metrics for all four backbones, showing that the gains come from dynamic evidence access during reasoning rather than from the trained model weights alone.

4.3 Interpreting Latent Memory Organization

Global and Local form two distinct types of latent memory. Figure 4 visualizes their benchmark-conditioned t-SNE projections. Local is benchmark-specific: its distribution changes across benchmarks. Global mixes across benchmarks because it captures abstract scene-level and cross-image relations. In the multi-image view, Global is more dispersed, suggesting that it represents a wider range of inter-image relations.

4.4 Memory Access and the Perceptual Cycle

Local provides fine-grained visual evidence, Global provides broader scene and cross-image context, and their interleaved calls restore the perceptual cycle. We analyze how the learned policy routes between these two forms of visual evidence as reasoning evolves. Figure 5 shows that memory returns are inserted into ongoing reasoning and change the subsequent reasoning state,

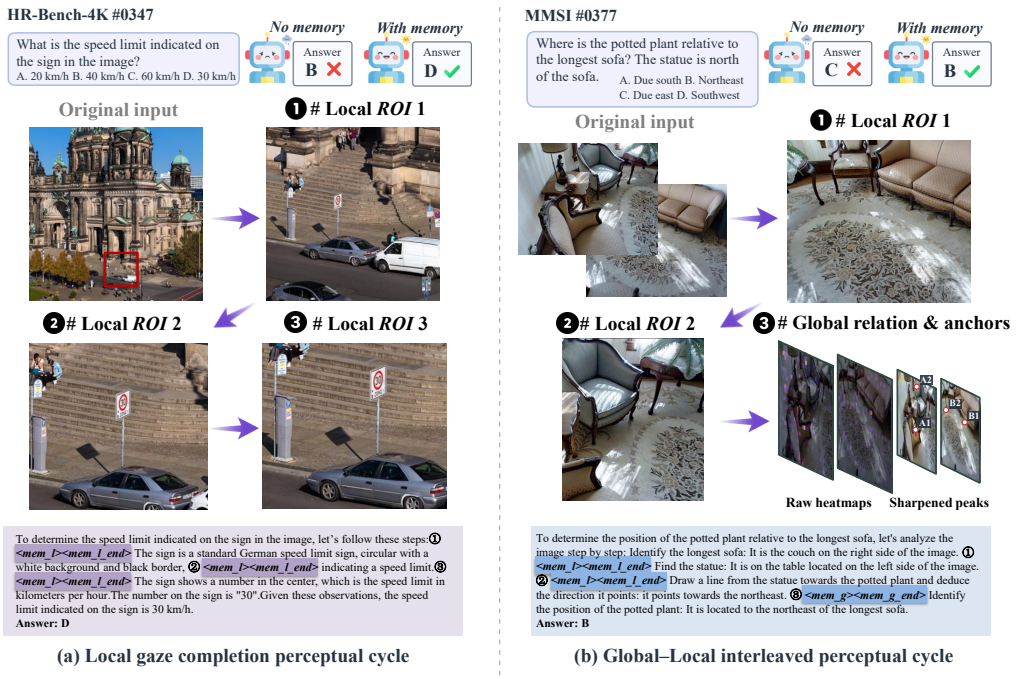


Figure 5: **Local refinement and Global–Local interaction.** (a) HR-Bench-4K #0347: repeated Local calls narrow the inspection from the street scene to a speed-limit sign, yielding 30 km/h rather than 40 km/h without memory. (b) MMSI #0377: two Local calls followed by a Global call accompany the northeast answer, whereas the paired no-call trajectory answers due east.

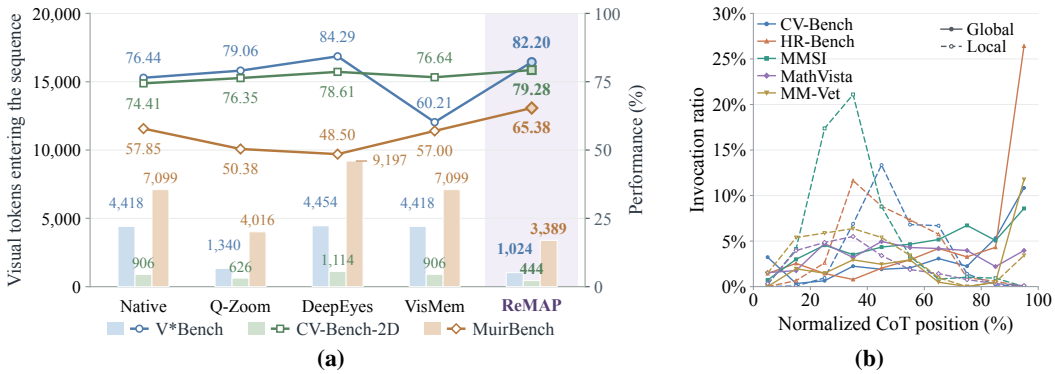


Figure 6: **Memory invocation and visual-context cost.** (a) Mean total visual tokens entering the sequence (bars) and performance (lines) on shared V*Bench, CV-Bench-2D, and MuirBench questions; the shaded column marks ReMAP. (b) Global (solid) and Local (dashed) invocation ratios over the normalized reasoning trajectory.

yielding a state-dependent evidence–reasoning loop rather than a fixed access schedule. Figure 6(b) shows that Local calls concentrate early and in the middle of the trajectory, whereas Global calls persist later.

4.5 Component, Capacity, and Access Ablations

Next, we study the gains from different components and design choices in ReMAP, including training, memory curation, memory access, and latent capacity.

Table 3: **Training, curation, and access ablations (%).** Base is Coarse; BLINK reports micro accuracy and MM-Vet mean graded scores.

Method	V*	CV-3D	Muir	BLINK	MM-Vet
Base	67.54	75.42	58.15	57.81	47.06
SFT	58.64	75.58	55.73	54.13	55.05
Stage I_{sft}	76.44	83.33	57.81	58.92	48.35
Memory curation					
w/o Memory	79.06	80.00	61.00	59.02	57.80
w/ Global memory	79.58	83.00	64.00	59.50	60.55
w/ Local memory	81.68	81.00	62.00	60.02	59.63
Memory access					
Fixed	76.96	83.00	61.46	59.39	58.72
Random	79.58	79.75	61.58	58.81	57.34
Entropy	79.06	81.17	62.00	59.39	59.63
ReMAP	**82.20**	**83.92**	**65.38**	**60.65**	**64.27**

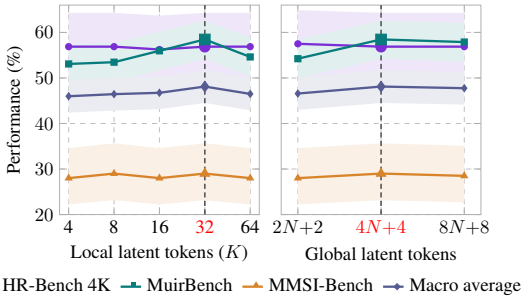


Figure 7: **Latent-capacity sensitivity.** Performance on 20% holdout splits from HR-Bench-4K, MuirBench, and MMSI-Bench. Curves show benchmark scores and their equal-weight macro average; shaded bands are item-level 95% intervals from one training seed. Dashed guides mark the selected $K = 32$ and $4N + 4$ settings.

Global and Local specialize by scope, and their combination is best. We analyze how curation scope affects benchmark performance. Global is stronger on CV-Bench-3D, MuirBench, and MM-Vet, whereas Local is stronger on V*Bench and BLINK. Combining both memories is best on every benchmark, exceeding the better of the Global-only and Local-only variants by 3.72 points on MM-Vet, as shown in Table 3.

Learned invocation is more reliable than fixed or heuristic schedules. We analyze whether the learned access policy selects more useful request points than Fixed, Random, and Entropy schedules. ReMAP is highest on every benchmark, exceeding the strongest of these alternatives by 3.38 points on MuirBench and 4.64 on MM-Vet. Fixed invocation falls below w/o Memory on V*Bench, showing that a fixed schedule can make added evidence harmful; Table 3 reports the access variants. The control definitions are given in Appendix A.9. Appendix A.10 examines branch-point selection and routing diversity.

ReMAP remains effective across a range of latent return sizes, making a compact capacity sufficient. We sweep the Local and Global return sizes to determine the default configuration (Figure 7). Performance is broadly stable across the tested capacities, while increasing the number of returned tokens does not bring consistent gains. We therefore use $K = 32$ Local tokens and $4N + 4$ Global tokens in the remaining experiments.

4.6 Visual-Context Cost

ReMAP maintains strong performance with substantially lower visual-context cost. We analyze the accuracy–context-cost tradeoff using the matched profiles in Figure 6(a). ReMAP remains competitive while using fewer visual tokens than Native, and its compact memory returns refresh visual evidence without repeatedly expanding the decoder context. Detailed accounting is provided in Appendix A.12.

5 Conclusion

ReMAP shows that visual memory becomes effective when the scope and timing of visual evidence follow the reasoning state. Global memory preserves question-conditioned scene and cross-image context, while Local memory reacquires fine-grained pixels as the trajectory demands. A learned policy coordinates these two forms of evidence within compact latent returns, improving over prior methods on all four multi-image benchmarks and over the Native backbone on all five single-image metrics while reducing the profiled visual context. Together, these components restore the perceptual cycle by letting each reasoning step use its current evidence to guide the next visual inspection. We hope this view helps future multimodal models treat perception as part of reasoning.

AI USE STATEMENT

We used generative AI tools to aid and polish the writing, retrieve and discover related literature, and draft sections of the manuscript. We reviewed all AI-assisted material and take responsibility for the final content of this paper.

ETHICS STATEMENT

This work uses public research datasets and no human subjects or newly collected personal data. ReMAP does not remove inherited model biases or failure modes; additional visual operations can also make source-allocation or localization errors appear confident.

REPRODUCIBILITY STATEMENT

Details of the memory interface, learning objectives, training configuration, data, evaluation protocols, and ablations are provided in the appendices. We will publicly release our source code and model weights upon acceptance of the paper.

REFERENCES

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Chao Chen, Zhixin Ma, Yongqi Li, Yupeng Hu, Yinwei Wei, Wenjie Li, and Liqiang Nie. Reasoning in the dark: Interleaved vision-text reasoning in latent space. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 39117–39129, San Diego, California, United States, July 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1949. URL https://aclanthology.org/2026.findings-acl.1949/.

Harold Haodong Chen, Xinxiang Yin, Wen-Jie Shu, Hongfei Zhang, Zixin Zhang, Chenfei Liao, Litao Guo, Qifeng Chen, and Ying-Cong Chen. Show, don't tell: Morphing latent reasoning into image generation. *arXiv preprint arXiv:2602.02227*, 2026b. URL https://arxiv.org/abs/2602.02227.

Tao Chen, Kun Zhang, Qiong Wu, Xiao Chen, Chao Chang, Xiaoshuai Sun, Yiyi Zhou, and Rongrong Ji. Scaling the long video understanding of multimodal large language models via visual memory mechanism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31877–31888, June 2026c. URL https://openaccess.thecvf.com/content/CVPR2026/html/Chen_Scaling_the_Long_Video_Understanding_of_Multimodal_Large_Language_Models_CVPR_2026_paper.html.

Xinyan Chen, Renrui Zhang, Dongzhi Jiang, Aojun Zhou, Shilin Yan, Weifeng Lin, and Hongsheng Li. MINT-CoT: Enabling Interleaved Visual Tokens in Mathematical Chain-of-Thought Reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 69110–69139. Curran Associates, Inc., 2025. doi: 10.52202/085713-2324. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6403d7e0824e7be2e699f47d9703ed80-Paper-Conference.pdf.

Xu Chu, Xinrong Chen, Guanyu Wang, Zhijie Tan, Kui Huang, Wenyu Lv, Tong Mo, and Weiping Li. Qwen Look Again: Guiding vision-language reasoning models to re-attention visual information, 2025. URL https://arxiv.org/abs/2505.23558.

Anurag Das, Adrian Bulat, Alberto Baldrati, Ioannis Maniadis Metaxas, Bernt Schiele, Georgios Tzimiropoulos, and Brais Martinez. More images, more problems? a controlled analysis

of VLM failure modes. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7423–7442, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.366. URL https://aclanthology.org/2026.findings-acl.366/.

Yizhuo Ding, Mingkang Chen, Zhibang Feng, Tong Xiao, Wanying Qu, Wenqi Shao, and Yanwei Fu. VTPerception-R1: Enhancing multimodal reasoning via explicit visual and textual perceptual grounding. *arXiv preprint arXiv:2509.24776*, 2025. URL https://arxiv.org/abs/2509.24776.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://iclr.cc/virtual/2026/poster/10009306.

Shuai Dong, Siyuan Wang, Xingyu Liu, Chenglin Li, Haowen Hou, and Zhongyu Wei. Interleaved latent visual reasoning with selective perceptual modeling. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29316–29335, San Diego, California, United States, July 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1351. URL https://aclanthology.org/2026.acl-long.1351/.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A. Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *Computer Vision – ECCV 2024*, pp. 148–166, 2024. doi: 10.1007/978-3-031-73337-6_9. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/3356_ECCV_2024_paper.php.

Jun Gao, Yongqi Li, Ziqiang Cao, and Wenjie Li. Interleaved-Modal Chain-of-Thought. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19520–19529, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gao_Interleaved-Modal_Chain-of-Thought_CVPR_2025_paper.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah Smith, and Ranjay Krishna. Visual Sketchpad: Sketching as a Visual Chain of Thought for Multimodal Language Models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 139348–139379. Curran Associates, Inc., 2024. doi: 10.52202/079017-4423. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/fb82011040977c7712409fbdb5456647-Paper-Conference.pdf.

Siyuan Huang, Xiaoye Qu, Yafu Li, Tong Zhu, Zefeng He, Muxin Fu, Daizong Liu, Wei-Long Zheng, and Yu Cheng. Persistent visual memory: Sustaining perception for deep generation in LVLMs, 2026. URL https://arxiv.org/abs/2605.00814.

Byungwoo Jeon, Yoonwoo Jeong, Hyunseok Lee, Minsu Cho, and Jinwoo Shin. Vision-aligned latent reasoning for multi-modal large language model. *arXiv preprint arXiv:2602.04476*, 2026. URL https://arxiv.org/abs/2602.04476.

Chaoya Jiang, Yongrui Heng, Wei Ye, Han Yang, Haiyang Xu, Ming Yan, Ji Zhang, Fei Huang, and Shikun Zhang. VLM-R^3: Region recognition, reasoning, and refinement for enhanced multimodal chain-of-thought. *arXiv preprint arXiv:2505.16192*, 2025. doi: 10.48550/arXiv.2505.16192. URL https://arxiv.org/abs/2505.16192.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W. F. Ku, Qian Liu, and Wenhu Chen. MANTIS: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=skLtdUVaJa.

Houcheng Jiang, Jiajun Fu, Junfeng Fang, Chen Gao, Xiang Wang, Xiangnan He, and Yong Li. UniVLR: Unifying text and vision in visual latent reasoning for multimodal LLMs. *arXiv preprint arXiv:2605.11856*, 2026. doi: 10.48550/arXiv.2605.11856. URL https://arxiv.org/abs/2605.11856.

Boyi Li, Yifan Shen, Yuanzhe Liu, Yifan Xu, Jiateng Liu, Xinzhuo Li, Zhengyuan Li, Jingyuan Zhu, Yunhan Zhong, Fangzhou Lan, Jianguo Cao, James M. Rehg, Heng Ji, Ismini Lourentzou, and Xu Cao. Toward cognitive supersensing in multimodal large language model. *arXiv preprint arXiv:2602.01541*, 2026a. URL https://arxiv.org/abs/2602.01541.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. LLaVA-NeXT-Interleave: Tackling multi-image, video, and 3D in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. URL https://arxiv.org/abs/2407.07895.

Yifan Li, Zhenghao Chen, Ziheng Wu, Kun Zhou, Ruipu Luo, Can Zhang, Zhentao He, Yufei Zhan, Wayne Xin Zhao, and Minghui Qiu. Unleashing perception-time scaling to multimodal reasoning models. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/499b12df1531fe8ee0febcf08381f3a4-Abstract-Conference.html.

Yunheng Li, Hangyi Kuang, Hengrui Zhang, Jiangxia Cao, Zhaojie Liu, Qibin Hou, and Ming-Ming Cheng. Rethinking token-level policy optimization for multimodal chain-of-thought. *arXiv preprint arXiv:2603.22847*, 2026c. URL https://arxiv.org/abs/2603.22847.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, volume 8693 of *Lecture Notes in Computer Science*, pp. 740–755. Springer, 2014. doi: 10.1007/978-3-319-10602-1_48. URL https://doi.org/10.1007/978-3-319-10602-1_48.

Chengzhi Liu, Zhongxing Xu, Qingyue Wei, Juncheng Wu, James Zou, Xin Eric Wang, Yuyin Zhou, and Sheng Liu. More thinking, less seeing? assessing amplified hallucination in multimodal reasoning models, 2025. URL https://arxiv.org/abs/2505.21523.

Chengzhi Liu, Yuzhe Yang, Yue Fan, Qingyue Wei, Sheng Liu, and Xin Eric Wang. Reasoning within the mind: Dynamic multimodal interleaving in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9225–9236, June 2026a. URL https://openaccess.thecvf.com/content/CVPR2026F/html/Liu_Reasoning_Within_the_Mind_Dynamic_Multimodal_Interleaving_in_Latent_Space_CVPRF_2026_paper.html.

Xu Liu, Yongheng Zhang, Qiguang Chen, Yao Li, Sheng Wang, and Libo Qin. Let's think with images efficiently! an interleaved-modal chain-of-thought reasoning framework with dynamic and precise visual thoughts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32213–32221, 2026b. doi: 10.1609/aaai.v40i38.40494. URL https://ojs.aaai.org/index.php/AAAI/article/view/40494.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/663bce02a0050c4a11f1eb8a7f1429d3-Abstract-Conference.html.

Lei Peng, Shuai Lv, and Wei Hu. ReGround: Restoring visual grounding in multi-step reasoning through self-diagnosis and visual re-examination. In *Proceedings of the 34th ACM International Conference on Multimedia*, 2026. URL https://arxiv.org/abs/2608.04385.

Tan-Hanh Pham and Chris Ngo. Multimodal chain of continuous thought for latent-space reasoning in vision-language models. *arXiv preprint arXiv:2508.12587*, 2025. URL https://arxiv.org/abs/2508.12587.

Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual CoT: Advancing multi-modal language models with a comprehensive dataset and benchmark for chain-of-thought reasoning. *arXiv preprint arXiv:2403.16999*, 2024a. URL https://arxiv.org/abs/2403.16999.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024b. URL https://arxiv.org/abs/2402.03300.

Yuheng Shi, Xiaohuan Pei, Minjing Dong, and Chang Xu. Catching the details: Self-distilled RoI predictors for fine-grained MLLM perception. *arXiv preprint arXiv:2509.16944*, 2025. URL https://arxiv.org/abs/2509.16944.

Yuheng Shi, Xiaohuan Pei, Linfeng Wen, Minjing Dong, and Chang Xu. Q-Zoom: Query-aware adaptive perception for efficient multimodal large language models. *arXiv preprint arXiv:2604.06912*, 2026. URL https://arxiv.org/abs/2604.06912.

Alex Su, Haozhe Wang, Weiming Ren, Fangzhen Lin, and Wenhu Chen. Pixel Reasoner: Incentivizing Pixel Space Reasoning via Curiosity-Driven Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 8222–8251. Curran Associates, Inc., 2025. doi: 10.52202/085713-0280. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/0c38f54740062529aa4117a04b583f3c-Paper-Conference.pdf.

Hai-Long Sun, Zhun Sun, Houwen Peng, and Han-Jia Ye. Mitigating visual forgetting via take-along visual conditioning for multi-modal long CoT reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5158–5171. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.257. URL https://aclanthology.org/2025.acl-long.257/.

Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, Austin Wang, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9ee3a664ccfeabc0da16ac6f1f1cfe59-Abstract-Conference.html.

Fei Wang, Xingyu Fu, James Y. Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, Tianyi Lorena Yan, Wenjie Jacky Mo, Hsiang-Hui Liu, Pan Lu, Chunyuan Li, Chaowei Xiao, Kai-Wei Chang, Dan Roth, Sheng Zhang, Hoifung Poon, and Muhao Chen. MuirBench: A comprehensive benchmark for robust multi-image understanding. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/9cf6139382f98623d08cc595622f3fb1-Abstract-Conference.html.

Jiacong Wang, Zijian Kang, Haochen Wang, Haiyong Jiang, Jiawen Li, Bohong Wu, Ya Wang, Jiao Ran, Xiao Liang, Chao Feng, and Jun Xiao. VGR: Visual Grounded Reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 89506–89528, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/90aeee6dfe75ab6b5d6958bce40d3e16-Paper-Conference.pdf.

Wenbin Wang, Liang Ding, Minyan Zeng, Xiabin Zhou, Li Shen, Yong Luo, Wei Yu, and Dacheng Tao. Divide, conquer and combine: A training-free framework for high-resolution image perception in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 7907–7915, 2025b. doi: 10.1609/aaai.v39i8.32852. URL https://ojs.aaai.org/index.php/AAAI/article/view/32852.

Yubo Wang, Juntian Zhang, Yichen Wu, Yankai Lin, Nils Lukas, and Yuhan Liu. Forest before trees: Latent superposition for efficient visual reasoning. In *Proceedings of the 64th Annual*

Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 11272–11288, San Diego, California, United States, July 2026b. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.517. URL https://aclanthology.org/2026.acl-long.517/.

Mingyuan Wu, Jingcheng Yang, Jize Jiang, Meitang Li, Kaizhuo Yan, Hanchao Yu, Minjia Zhang, Chengxiang Zhai, and Klara Nahrstedt. VTool-R1: Vlms learn to think with images via reinforcement learning on multimodal tool use. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/7e6f445a74cdb71931aac64f1e3f49c9-Abstract-Conference.html.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13084–13094, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Wu_V_Guided_Visual_Search_as_a_Core_Mechanism_in_Multimodal_CVPR_2024_paper.html.

xAI. Grok-1.5 Vision Preview. Official xAI announcement introducing RealWorldQA, 2024. URL https://x.ai/news/grok-1.5v.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, Dahua Lin, Tai Wang, and Jiangmiao Pang. MMSI-Bench: A benchmark for multi-image spatial intelligence. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/fe5d4bd3e2af823701b5d8f70a3c7602-Abstract-Conference.html.

Zeyuan Yang, Xueyang Yu, Delin Chen, Maohao Shen, and Chuang Gan. Machine Mental Imagery: Empower Multimodal Reasoning with Latent Visual Tokens. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 33510–33520, June 2026b. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yang_Machine_Mental_Imagery_Empower_Multimodal_Reasoning_with_Latent_Visual_Tokens_CVPR_2026_paper.html.

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, and Lanxiao Huang. Mastering complex control in MOBA games with deep reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/1912.09729.

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Auto-Controlled Image Perception in MLLMs via Visual Perception Tokens. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 21822–21831, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Yu_Auto-Controlled_Image_Perception_in_MLLMs_via_Visual_Perception_Tokens_ICCV_2025_paper.html.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. MM-Vet: Evaluating large multimodal models for integrated capabilities. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57730–57754. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24o.html.

Xinlei Yu, Chengming Xu, Guibin Zhang, Zhangquan Chen, Yudong Zhang, Yongbo He, Peng-Tao Jiang, Jiangning Zhang, Xiaobin Hu, and Shuicheng Yan. VisMem: Latent vision memory unlocks potential of vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 31544–31555, June 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Yu_VisMem_Latent_Vision_Memory_Unlocks_Potential_of_Vision-Language_Models_CVPR_2026_paper.html.

Guanghao Zhang, Tao Zhong, Yan Xia, Mushui Liu, Zhelun Yu, Haoyuan Li, Wanggui He, Dong She, Yi Wang, and Hao Jiang. CMMCoT: Enhancing Complex Multi-Image Comprehension via Multi-Modal Chain-of-Thought and Memory Augmentation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(15):12430–12438, 2026a. doi: 10.1609/aaai.v40i15.38236. URL https://ojs.aaai.org/index.php/AAAI/article/view/38236.

Guibin Zhang, Muxin Fu, and Shuicheng Yan. MemGen: Weaving Generative Latent Memory for Self-Evolving Agents. In *International Conference on Learning Representations*, volume 2026, pp. 22555–22588, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/26407588dfb08e48c459c074ab6adb7d-Paper-Conference.pdf.

Xin Zhang, Qiqi Tao, Jiawei Du, Moyun Liu, and Joey Tianyi Zhou. Visual latents know more than they say: Unsilencing latent reasoning in MLLMs. *arXiv preprint arXiv:2605.02735*, 2026c. URL https://arxiv.org/abs/2605.02735.

Yang Zhang, Xiaoshuai Sun, Rui Zhao, Wujin Sun, Yidong Chen, Jiayi Ji, Qian Chen, and Rongrong Ji. Look on demand: A cognitive scheduling framework for visual evidence acquisition in multimodal reasoning. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026d. URL https://arxiv.org/abs/2605.28160.

Ziwei Zheng, Minghao Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. DeepEyes: Incentivizing "thinking with images" via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cdb347d7516d52a9280ea9d6708911f6-Abstract-Conference.html.

Jiazhou Zhou, Yucheng Chen, Hongyang Li, Qing Jiang, Hu Zhou, Ying-Cong Chen, and Lei Zhang. V-Reflection: Transforming mllms from passive observers to active interrogators, 2026. URL https://arxiv.org/abs/2604.03307.

A Appendix

A.1 Related Work and Positioning

Memory curation: acquiring and selecting visual evidence. Memory curation determines which evidence reasoning acquires. DeepEyes learns crop requests (Zheng et al., 2026), while Pixel Reasoner combines image cropping and video-frame selection (Su et al., 2025). VLM-R^3 interleaves region selection, cropping, and zooming with textual reasoning (Jiang et al., 2025). VPT supports token-guided cropping and state-conditioned full-image re-encoding (Yu et al., 2025). CMMCoT generates image indices and boxes to re-encode entity regions during multi-image reasoning (Zhang et al., 2026a). Qwen Look Again, ICoT, and MINT-CoT instead select or reuse encoded visual tokens (Chu et al., 2025; Gao et al., 2025; Chen et al., 2025). VTPerception-R1 grounds reasoning in explicit perceptual descriptions (Ding et al., 2025), while perception-time scaling decomposes perceptual estimates into symbolic intermediate steps (Li et al., 2026b). VisMem uses state-conditioned formers to produce perceptual and semantic memory (Yu et al., 2026). V-Reflection trains latent reasoning states with spatial supervision (Zhou et al., 2026).

Memory organization: representing visual evidence. Visual Sketchpad represents intermediate evidence as generated sketches (Hu et al., 2024), VTool-R1 as annotated images (Wu et al., 2026), and CSMR as text (Zhang et al., 2026d). Take-along Visual Conditioning pools visual tokens (Sun et al., 2025), while Persistent Visual Memory projects image embeddings into a lower-dimensional space (Huang et al., 2026). CMMCoT retrieves cross-image cues from cached per-layer visual key–value features (Zhang et al., 2026a). FlexMem compresses and retrieves visual KV memories for long-video understanding (Chen et al., 2026c). MemGen and VisMem represent memory as latent token sequences (Zhang et al., 2026b; Yu et al., 2026). Laser and Mirage represent intermediate reasoning through continuous hidden states or self-generated visual tokens (Wang et al., 2026b; Yang et al., 2026b). MCOUT-Multi refines continuous thoughts by attending to multimodal input embeddings (Pham & Ngo, 2025), while Cognitive Supersensing predicts answer-image latents and

Table 4: **Qualitative positioning of representative visual-memory methods.** Green checks indicate explicit support, red crosses indicate non-core mechanisms, and gray dashes indicate dimensions not emphasized in the cited work.

Method	Pixel / Region Reacquisition	Compact Visual Memory	State-Conditioned Access	Cross-Image Context	Global / Local Split	Invocation vs. Continuation
DeepEyes	✓	✗	–	–	✗	✗
Pixel Reasoner	✓	✗	–	–	✗	✗
VLM-R^3	✓	✗	–	–	✗	✗
CMMCoT	✓	✓	–	✓	✗	✗
Qwen Look Again	✗	✓	–	–	✗	✗
ICoT / MINT-CoT	✗	✓	–	–	✗	✗
MemGen	✗	✓	✓	–	✗	✗
VisMem	✗	✓	✓	–	✗	✗
FlexMem	✗	✓	–	–	✗	✗
ReGround	✓	✗	✓	–	✗	✗
ReMAP (ours)	✓	✓	✓	✓	✓	✓

uses them to guide rationale training (Li et al., 2026a). IVT-LR combines recurrent hidden states with selected input-image embeddings for latent reasoning (Chen et al., 2026a). ILVR interleaves text with generated visual latents supervised by selected features from intermediate helper images (Dong et al., 2026b). UniVLR supervises latent reasoning with visual encodings of rendered reasoning text and auxiliary images (Jiang et al., 2026). Zhang et al. (2026c) identify underused visual latents and optimize them at inference time with a frozen backbone.

Memory access: timing and selecting memory use. Take-along Visual Conditioning and Qwen Look Again reactivate visual evidence at reflection events (Sun et al., 2025; Chu et al., 2025). ReGround uses self-diagnosis to guide re-examination (Peng et al., 2026), and VGR triggers region-specific replay from precomputed visual features (Wang et al., 2026a). DaP-ICoT inserts selected object images when the reasoning step's mean logit margin falls below a threshold (Liu et al., 2026b). MemGen learns a memory trigger (Zhang et al., 2026b). LatentMorph learns when to invoke latent reasoning over compact visual memory during image generation (Chen et al., 2026b). VisMem inserts perceptual or semantic latent memory through invocation tokens (Yu et al., 2026). Its two memories differ in abstraction level rather than in spatial scope, and a request is not compared with continuing from the same state. CSMR invokes a separate perception module (Zhang et al., 2026d). For policy training, ARPO allocates entropy-guided rollout branches after tool feedback (Dong et al., 2026a). PEPO combines visual hidden-state similarity and token entropy to reweight token advantages without auxiliary branches (Li et al., 2026c).

Positioning ReMAP. ReMAP bridges these dimensions by connecting active visual evidence acquisition with latent memory and learned access. Within memory curation, it emphasizes the complementarity between newly acquired local details and the scene and cross-image context needed to interpret them. Within memory organization, it brings pixel-level reacquisition into compact latent memory, connecting renewed perception with an efficient representation of visual evidence. Within memory access, it focuses on the utility of requesting evidence at a given reasoning state, using comparisons between invocation and continuation to guide selective access. Together, these choices support sustained visual grounding throughout reasoning.

A.2 Runtime and Memory Interface

Memory transactions. ReMAP begins autoregressive reasoning from a coarse visual prefix of the input images. Ordinary autoregressive generation continues by default. A memory transaction is performed only when the decoder emits a Global or Local request token. The corresponding transactions are

$$\big[\langle \texttt{mem_g} \rangle;\, Z_G;\, \langle \texttt{mem_g_end} \rangle\big], \qquad \big[\langle \texttt{mem_l} \rangle;\, Z_L;\, \langle \texttt{mem_l_end} \rangle\big]. \tag{4}$$

After the decoder emits a request token, the runtime inserts the corresponding latent block and end marker into the current decoder sequence. Generation resumes from the position following the end marker. End markers are not sampled as vocabulary outputs, and returned latents enter the decoder as continuous inputs. Each reasoning trajectory permits at most five memory requests. After the request budget is exhausted or the model commits an answer, Global and Local request tokens are masked.

Inserted latents and end markers occupy successive positions in the decoder sequence and KV cache. Request tokens and these inserted positions do not count toward the ordinary-text budget, but they still consume decoder context capacity.

Global memory. Global curation forms a question-conditioned latent block from the ordered input images. For an input containing N images, four tokens are allocated to each image and four additional tokens encode cross-image relations:

$$K_G(N) = 4N + 4. \tag{5}$$

The resulting block is formed before decoding, cached throughout the reasoning trajectory, and inserted at a new sequence position whenever the decoder emits a Global request. Repeated Global requests neither re-encode the input images nor regenerate the cached block from a later reasoning state.

Local memory. The Local request uses the current reasoning state to select image sources and regions. The selector retains at most three sources, stopping when cumulative selection probability reaches 0.85. A frozen vision encoder maps the selected crops to region features, and the Local Weaver combines these features with the reasoning state and cached Global memory:

$$Z_L = [W_L(h_t, R_{t,i}; Z_G)]_{i \in \mathcal{S}_t}. \tag{6}$$

Each Local request returns 32 latent tokens. Global memory supplies context for Local compression but is not concatenated into the 32-token Local output.

The $4N + 4$ Global layout, the 32-token Local return, the three-source cap, and the 0.85 threshold define the memory interface used in all main experiments. The Source Selector reads the layer-12 state exposed at a request boundary, and Twig uses layer-18 prefix states selected by a layer sweep. Stage II rollout branching follows the partial-rollout strategy of ARPO (Dong et al., 2026a) and is specified with the access objective below.

A.3 Memory Formation and Access Objectives

A.3.1 Global Layout and Caching

Each per-image Image-Weaver span contains four latent queries and is restricted by its attention mask to the visual activations of its own source image. Thus $Z_i^{\text{img}} \in \mathbb{R}^{4\times d}$. The Relation Weaver uses four queries over the concatenated image summaries $[Z_1^{\text{img}}; \cdots ; Z_N^{\text{img}}]$; its relation span can attend to all sources, giving $\mathcal{Z}^{\text{rel}} \in \mathbb{R}^{4\times d}$. Both weavers are conditioned on the encoded question, as in Equation 1. The Global block is therefore

$$Z_G = [Z_1^{\text{img}}; \cdots ; Z_N^{\text{img}}; \mathcal{Z}^{\text{rel}}] \in \mathbb{R}^{(4N+4)\times d}, \tag{7}$$

which is the $K_G(N) = 4N + 4$ layout used in the capacity sweep of Figure 7. We compute this block once from the prompt-time images and question, retain it as Z_G, and reuse the same latent values at every later Global request. Only the decoder insertion position and its resulting KV entries are new; later reasoning states determine whether a request is made, but do not recompute or alter the cached block.

A.3.2 Local Selection and Compression

The Source Selector uses the layer-12 request state $h_t = h_t^{12}$ to rank the source images. If $p_{t,i}$ denotes the normalized score of source i, the selector orders sources by decreasing score and keeps

the shortest prefix whose cumulative score reaches 0.85, with a maximum of three sources. When the threshold is not reached within three sources, it keeps the three highest- scoring sources. The layer-12 request tap and layer-18 Twig tap are ReMAP settings selected by internal layer sweeps: layer 12 supplies the state at a request boundary, while layer 18 provides the spatially arranged activations used for localization. Twig Localization uses the layer-18 prefix states $H^{18}_{<t}$ to locate a region $b_{t,i}$ in each selected image I_i. Let V^c_i denote the coarse visual features of image I_i. Source selection, region localization, and frozen crop encoding are written as

$$\begin{aligned} \mathcal{S}_t &= f_{\text{sel}}\big(h_t, \{V^c_i\}_{i=1}^N\big), \\ b_{t,i} &= f_{\text{loc}}\big(H^{18}_{<t}, V^c_i\big), \qquad R_{t,i} = E_{\text{roi}}\big(I_i[b_{t,i}]\big). \end{aligned} \tag{8}$$

Here, $I_i[b_{t,i}]$ is the selected crop, and E_{roi} is the frozen vision encoder that produces its region features $R_{t,i}$.

After processing $H^{18}_{<t}$ with its decoder blocks, Twig applies query/key projections and multimodal rotary position embeddings (mRoPE). In Figure 3(a), q^h_t denotes the query at the current reasoning position for head h, and K^h_{img} contains the keys at the selected image's visual-token positions. The ROI heatmap is the head-averaged scaled dot product reshaped to the image's coarse spatial grid:

$$q_{\text{roi}} = \text{reshape}_{G_h \times G_w}\left(\frac{1}{H}\sum_{h=1}^{H}\frac{q^h_t (K^h_{\text{img}})^\top}{\sqrt{d_h}}\right). \tag{9}$$

For readout, q_{roi} is treated as a logit map consistent with the weighted-BCE localization objective. We retain grid cells with $q_{\text{roi}} > 0$ (equivalently, sigmoid probability at least 0.5), take the min–max box of their row and column coordinates, map the box from the coarse grid to image pixels, and clip it to the image boundary. The resulting box is $b_{t,i}$ for crop encoding.

For local compression, the Local Weaver conditions its learnable compression queries Q_L on the reasoning state h_t and cached Global memory Z_G. Twig's q_t is used only for region localization. The crop features $R_{t,i}$ provide fine-grained evidence, while Z_G supplies image summaries and cross-image relational context. Equation 2 summarizes source-specific compression and concatenation into a return with $K = 32$ latent tokens in the default configuration.

A.3.3 Formation and Warm-up Objectives

Continuation distillation. The Image, Relation, and Local Weavers are trained through the frozen LLM. For a teacher trajectory with prefix $\mathcal{P}_t$ and continuation Y, we insert a Global or Local transaction at position t and minimize

$$\mathcal{L}_{\text{suffix}} = -\frac{1}{|Y|}\sum_{k=1}^{|Y|}\log p_{\text{LLM}}\big(Y_k \mid \mathcal{P}_t,\, [\langle\texttt{mem}\rangle; Z; \langle\texttt{mem_end}\rangle],\, Y_{<k}\big), \tag{10}$$

where Z is Z_G or Z_L and the markers match its memory type. The LLM is frozen, so gradients reach only the weaver that produced Z. The loss therefore trains each weaver to encode the evidence that the teacher's subsequent reasoning uses. Together with $\mathcal{L}_{\text{suffix}}$, fact, relation, and answer supervision from the Stage I annotations (Appendix A.5) trains the weavers.

ROI localization. Twig produces ROI logits q_{roi} on the coarse grid of each selected image. The teacher-reviewed box is rasterized to a binary target $\bar{A}$ on the same grid. With $P = \text{sigmoid}(q_{\text{roi}})$, the localization objective is

$$\begin{aligned} \mathcal{L}_{\text{ROI}} &= \mathcal{L}_{\text{wBCE}}(\bar{A}, q_{\text{roi}}) + 1 - \text{softIoU}(\bar{A}, P), \\ \text{softIoU}(\bar{A}, P) &= \frac{\sum \bar{A} \odot P}{\sum \bar{A} + \sum P - \sum \bar{A} \odot P}. \end{aligned} \tag{11}$$

Q-Zoom (Shi et al., 2026) trains its ROI predictor with binary cross-entropy. We weight positive cells by the ratio of negative to positive cells, so that a small target region is not dominated by background cells. We cap this weight at 50 based on our experience. The soft-IoU term matches the overall extent of the heatmap to the target box.

Student-state adaptation. Teacher trajectories do not contain the states that the student reaches at inference. We therefore run the student autoregressively and apply the same two objectives to its states. For a student state s_t with complete text-and-latent prefix $\mathcal{P}_t$, the teacher reviews $\mathcal{P}_t$ together with the selected crop $I_i[b_{t,i}]$ and returns an ROI target $\bar{A}$ and a continuation target Y. Twig is updated with $\mathcal{L}_{\mathrm{ROI}}$, and the weavers with $\mathcal{L}_{\mathrm{suffix}}$ conditioned on $\mathcal{P}_t$.

Interface warm-up. Each warm-up sequence either contains no memory transaction or contains one Global or Local transaction at a request boundary. The frozen formation modules compute Z, which enters the sequence as a constant input. Warm-up minimizes next-token cross-entropy over the supervised positions $\mathcal{T}$:

$$\mathcal{L}_{\mathrm{warm}} = -\frac{1}{|\mathcal{T}|} \sum_{k \in \mathcal{T}} \log \pi(y_k \mid y_{<k}), \qquad \mathcal{T} = \mathcal{T}_{\mathrm{call}} \cup \mathcal{T}_{\mathrm{end}} \cup \mathcal{T}_{\mathrm{later}}. \tag{12}$$

Here $\mathcal{T}_{\mathrm{call}}$ contains the request token, $\mathcal{T}_{\mathrm{end}}$ the first text token after the end marker, and $\mathcal{T}_{\mathrm{later}}$ the remaining reasoning and answer tokens. The three sets give $\mathcal{L}_{\mathrm{call}}$, $\mathcal{L}_{\mathrm{end}}$, and $\mathcal{L}_{\mathrm{later}}$ in Figure 3(b). Prompt tokens, inserted latents, and end markers are excluded from $\mathcal{T}$. Sequences without a transaction supervise ordinary text at states where no memory is requested, so the decoder does not learn to request memory at every state. Only the four marker input rows and the two request output rows are updated. The end-marker input rows receive gradients through the text predicted after the return.

A.3.4 Access Learning Objective

Policy parameters. Let $\pi_{\phi,\psi}$ denote the decoding policy. The symbol ϕ denotes the decoder LoRA parameters, and $\psi = (\eta_{\mathrm{G}}, \eta_{\mathrm{L}}, w_{\mathrm{G}}, w_{\mathrm{L}})$ denotes the request interface, where η_{G} and η_{L} are the request-token input embeddings and w_{G} and w_{L} are their output-classifier rows. All other parameters, including the end-marker embeddings and the memory-formation modules, stay fixed. Let $\mathcal{V}_t$ be the legal vocabulary and $h_{\phi,\eta}(\mathcal{P}_t)$ the decoder state over the complete text-and-latent prefix. The request rows share the output softmax with the frozen ordinary rows w_v:

$$\ell(v \mid s_t) = w_v^{\top} h_{\phi,\eta}(\mathcal{P}_t), \qquad \pi_{\phi,\psi}(v \mid s_t) = \frac{\exp \ell(v \mid s_t)}{\sum_{u \in \mathcal{V}_t} \exp \ell(u \mid s_t)}. \tag{13}$$

This softmax gives $p_t(\mathrm{G})$ and $p_t(\mathrm{L})$, and the remaining mass is $p_t(\mathrm{N}) = 1 - p_t(\mathrm{G}) - p_t(\mathrm{L})$. An ordinary output z has probability $p_{\phi,\psi}(\mathrm{N} \mid s_t)\, q_{\phi,\eta}(z \mid s_t, \mathrm{N})$. Updating w_{G} and w_{L} changes the request logits and the shared normalizer, and updating η_{G} and η_{L} changes the hidden states after a request. Invocation credit therefore acts on the whole suffix after a request, including the text generated after the returned latents.

Rollouts and branch selection. Each prompt samples four root trajectories from the behavior policy π_{old}. Candidate branch states precede answer commitment and admit at least one legal request, including the initial state. Each root branches once, at the candidate with the largest $u(s_t, a_t)$ in Equation 3, with ties resolved by the earliest position. From the shared prefix, two siblings ($m = 2$) are sampled independently, each with its own subsequent requests and reasoning. Requests in the shared prefix count toward the five-request budget. The roots and their siblings form the natural group $\mathcal{G}$, so $4 \le |\mathcal{G}| \le 12$.

Rewards. For a completed trajectory τ_i, the terminal reward scores its extracted answer $\hat{y}(\tau_i)$ against the reference $y^{\star}$:

$$R_i = \mathrm{Score}(\hat{y}(\tau_i), y^{\star}) \in [0, 1]. \tag{14}$$

Typed answer checks are used where applicable, and semantic evaluation gives graded scores; for example, listing questions receive per-example F1. A prompt enters the policy losses only when all natural trajectories and the required continuation controls have valid scores. Unavailable scores are not treated as zero.

Same-state continuation baseline. For branch b, the siblings at b are the root and its resampled continuations, and $\mathcal{G}_b$ contains them. Let $D_b = \{i \in \mathcal{G}_b : a_{i,b} = \mathrm{N}\}$ contain the siblings that continue without a request at b, which can include the root. The baseline B_b in Section 3.2 is their mean reward:

$$B_b = \begin{cases} |D_b|^{-1} \sum_{i \in D_b} R_i, & D_b \neq \varnothing, \\ R(\tau_b^{\mathrm{aux}}), & D_b = \varnothing, \end{cases} \tag{15}$$

where τ_b^{aux} starts from the same branch state and is forced to emit an ordinary token at its first decision. Its later decisions follow π_{old} and may request memory. The auxiliary continuation only supplies B_b and does not enter $\mathcal{G}$.

Advantages. The decoder uses the group-relative advantage of GRPO (Shao et al., 2024b), and the request interface uses the normalized gain over continuation at the same branch state:

$$\hat{A}_i^{\text{dec}} = \frac{R_i - \bar{R}_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \delta}, \qquad \hat{A}_i^{\text{call}} = \frac{R_i - B_{b(i)}}{\tilde{\sigma}_{b(i)}}, \qquad \tilde{\sigma}_b = \begin{cases} \sigma_b + \delta, & \sigma_b > \delta, \\ 1, & \sigma_b \le \delta. \end{cases} \tag{16}$$

Here $\bar{R}_{\mathcal{G}}$ and $\sigma_{\mathcal{G}}$ are the mean and sample standard deviation of rewards in $\mathcal{G}$, and $b(i)$ is the branch at which trajectory i requests memory. The branch scale σ_b is the sample standard deviation of the rewards in $\mathcal{G}_b$ together with one additional entry B_b. We set $\delta = 10^{-6}$. Equal rewards give zero advantage in both terms.

Policy objectives. Let z_{ij} be the vocabulary output at decision j of trajectory i, including ordinary text, request tokens, and EOS. Its likelihood ratio $\rho_{ij} = \pi_{\phi,\psi}(z_{ij} \mid s_{ij}) / \pi_{\text{old}}(z_{ij} \mid s_{ij})$ uses the same legal-output mask as sampling. We use the dual-clipped policy loss (Ye et al., 2020):

$$\begin{aligned} \ell_{\text{clip}}(\rho, A) &= -\min\{\rho A,\ \text{clip}(\rho, 1-\epsilon, 1+\epsilon)A\}, \\ \ell_{\text{dc}}(\rho, A) &= \begin{cases} \min\{\ell_{\text{clip}}(\rho, A), -cA\}, & A < 0, \\ \ell_{\text{clip}}(\rho, A), & A \ge 0, \end{cases} \end{aligned} \tag{17}$$

where ϵ is the ratio-clipping width and $c > 1$ bounds the loss for negative advantages. The decoder term uses all T_i sampled decisions of each natural trajectory, including request tokens, EOS, and the shared prefix of each sibling, with $N_{\text{dec}} = \sum_{i \in \mathcal{G}} T_i$. The invocation term uses the set $\mathcal{C}$ of decisions from natural trajectories that request memory at their branch, from the request through answer commitment; it includes ordinary text as well as later requests. Returned latents and end markers are inserted by the runtime and contribute no policy term. With $N_{\text{call}} = \max(1, |\mathcal{C}|)$, the two objectives are

$$\begin{aligned} \mathcal{L}_{\text{dec}} &= \frac{1}{N_{\text{dec}}} \sum_{i \in \mathcal{G}} \sum_{j=1}^{T_i} \ell_{\text{dc}}(\rho_{ij}, \hat{A}_i^{\text{dec}}) + \lambda_{\text{CE}} \mathcal{L}_{\text{CE}}, \\ \mathcal{L}_{\text{access}} &= \frac{1}{N_{\text{call}}} \sum_{(i,j) \in \mathcal{C}} \ell_{\text{dc}}(\rho_{ij}, \hat{A}_i^{\text{call}}) + \lambda_{\text{pref}} \mathcal{L}_{\text{pref}}. \end{aligned} \tag{18}$$

The policy part of $\mathcal{L}_{\text{access}}$ is zero when $\mathcal{C}$ is empty. Only ϕ receives gradients from $\mathcal{L}_{\text{dec}}$, and only ψ receives gradients from $\mathcal{L}_{\text{access}}$. Each objective is computed per prompt and averaged over the prompts eligible for it. The values of ϵ, c, λ_{CE}, and λ_{pref} are listed in Table 5.

Post-call answer supervision. Let $\mathcal{P}_t^+$ be the complete prefix just after a memory return, including the request, its latents, and its end marker. The target Y contains the answer-format prefix, the reference answer, and EOS. With memory requests masked, the auxiliary loss is

$$\mathcal{L}_{\text{CE}} = -\frac{1}{|Y|} \sum_{k=1}^{|Y|} \log \pi_{\phi,\psi}^{\text{text}}(Y_k \mid \mathcal{P}_t^+, Y_{<k}). \tag{19}$$

The return is taken from the last pre-answer request of the first root that makes one. If no root requests memory, a forced initial Global or Local probe supplies the return. The prefix $\mathcal{P}_t^+$ contains no later returns, reasoning suffix, or reference answer. This term trains the decoder to answer from returned evidence and needs only a valid reference, independent of the rollout reward.

Initial-state request preferences. Three probes start from the initial state s_0. One is forced to emit an ordinary token, and the other two emit the Global and Local request tokens; later decisions follow π_{old}. Let $\Delta_a = R(\tau_0^a) - R(\tau_0^{\text{N}})$ for $a \in \{\text{G}, \text{L}\}$. The preference loss is

$$\mathcal{L}_{\text{pref}} = \frac{1}{2} \sum_{a \in \{\text{G},\text{L}\}} |\Delta_a| \ \text{softplus}\left[-\text{sgn}(\Delta_a) \log \frac{p_\psi(a \mid s_0)}{p_\psi(\text{N} \mid s_0)} \right]. \tag{20}$$

The loss raises the probability of a request relative to continuation when that request improves the reward, and lowers it otherwise. The initial hidden state is detached, so only w_{G} and w_{L} receive gradients. The probes do not enter either policy term. A prompt uses this term only when all three probes have valid scores.

A.4 Training Configuration

Figure 8 shows the five training steps. The vision encoder and the pretrained LLM weights stay frozen in every step, and each step updates only the modules marked as trainable. The Stage I objectives are defined in Appendix A.3.3, and the Stage II objectives in Appendix A.3.4.

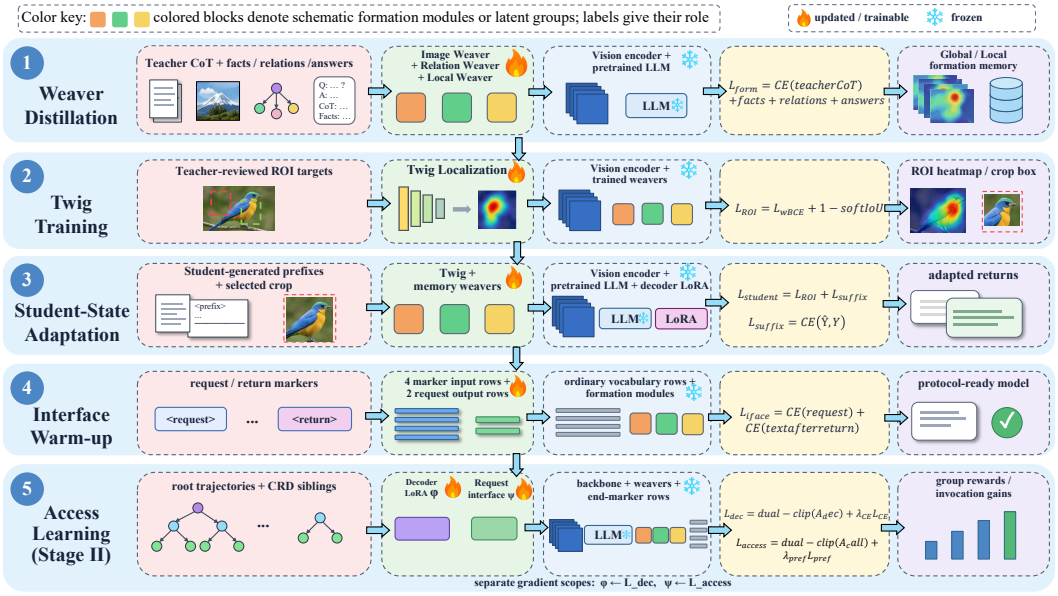


Figure 8: **Five-step training pipeline.** Each row lists the inputs of one step, the updated modules (flame), the frozen components (snowflake), the objective, and the output. Steps 1–4 form Stage I, and step 5 is Stage II access learning, in which the decoder LoRA ϕ and the request interface ψ receive gradients only from $\mathcal{L}_{\mathrm{dec}}$ and $\mathcal{L}_{\mathrm{access}}$, respectively.

Optimization. Table 5 lists the optimizer settings. The ratio-clipping width $\epsilon = 0.2$ and the dual-clip constant $c = 3$ follow Ye et al. (2020), and $\delta = 10^{-6}$ only stabilizes the normalization in Equation 16. The decoder LoRA (Hu et al., 2021) is attached to the query, key, value, output, and MLP projections of the top four decoder layers, with rank $r = 16$ and $\alpha = 32$. We set the learning rates, the LoRA configuration, the auxiliary loss weights, and the gradient-norm limit based on our experience. The Global Weavers use a separate learning rate for the output projection that maps their latents to the decoder embedding space. Each Stage I step makes one pass over its supervision. The random seed is 20260911.

Rollouts and training length. Stage II runs on 16 A800 GPUs with one prompt per GPU and no gradient accumulation, so each step schedules 16 prompts. Each prompt samples four root trajectories at temperature 1, and each eligible root adds two siblings, so a step samples 64–192 natural trajectories. Continuation controls and preference probes are sampled in addition. Rollouts use a 768-token ordinary-text budget, a 64-token answer-completion grace period, and at most five memory requests. These rollout sizes and budgets are set empirically. The 40,000-example schedule contains 2,500 steps.

Checkpoint selection. We evaluate the model every 100 steps and at the end of training on a fixed development set whose task list is set before training. The selected checkpoint maximizes the equally weighted mean of per-task rewards, and ties select the earlier checkpoint. The same weights are used for all final benchmarks, and final benchmark scores do not enter the selection.

Table 5: **Optimization settings for memory formation and access learning.**

Stage	Parameter group or setting	Value
I: Global formation	Image and Relation Weavers	10^{-6}
	Output projection	10^{-5}
I: Local formation	Source Selector, Twig, and Local Weaver	10^{-6}
I: interface warm-up	Marker and request rows	10^{-4}
II: access learning	Decoder LoRA	10^{-6}
	Decoder LoRA rank r / α	16 / 32
	Decoder LoRA layers	top 4 of 28
	Request rows	3×10^{-4}
	Ratio-clipping width ϵ	0.2
	Dual-clip constant c	3
	Answer-loss weight λ_{CE}	0.1
	Preference-loss weight λ_{pref}	0.1
All stages	Optimizer / weight decay	AdamW / 0
	Maximum gradient norm	1

A.5 Training Data

Sources and allocation. We construct 50,000 image–question examples for Local memory formation, 30,000 for Global memory formation, and 40,000 for Stage II access learning. Each example contains one or more images, a question, and a reference answer. Table 6 lists the source datasets. All examples come from the training splits of these sources and are deduplicated against every evaluation unit in Table 7, so there is no leakage between training and evaluation data.

Table 6: **Training data sources.** Source datasets provide question–answer pairs, region annotations, or instance annotations from which additional questions are constructed.

Source	Content and use
Visual-CoT (GQA) (Shao et al., 2024a)	Object, attribute, and spatial-relation questions with region annotations.
COCO train2017 (Lin et al., 2014)	Multi-image questions built from instance annotations.
Mantis-Instruct (Jiang et al., 2024): ChartQA, Spot-the-Diff, ImageCoDe, NLVR2, and Birds-to-Words	Chart reading and arithmetic, image comparison, description-to-image matching, and cross-image relational judgments.
Q-Zoom's VStar-COCO data (Shi et al., 2026)	Spatial-relation questions on COCO train2017 images.
ScanQA via M4-Instruct (Li et al., 2024)	Multi-view scene question answering, restricted to examples from the original ScanQA training split.

Stage I annotations. Each Stage I example carries a reference CoT, per-image facts, cross-image relations, the answer, and target regions. Qwen3-VL-32B-Instruct generates the reference CoTs. Answers come from the source datasets. Target regions come from source region annotations and from model-generated step annotations, which are checked against the source images and the proposed crops. Per-image facts and cross-image relations are derived from source annotations and from model-generated pseudo-labels. Facts and relations supervise what Global memory encodes, region annotations supervise Local source selection and localization, and answers supervise task completion.

Student-state targets. For student-state adaptation, we roll out Qwen2.5-VL-7B-Instruct and record each text-and-latent prefix with its source selections, crops, and memory returns. The teacher reviews each prefix together with its crop and returns only the ROI target $\bar{A}$ and the continuation target Y. We do not reuse the full teacher CoT as the continuation of a student state, because that CoT need not follow from the student's prefix.

Access-learning data. Stage II examples cover single-image perception, multi-image reasoning, and general visual question answering. Rewards and the post-call answer loss use the source answers, and the Stage I memory annotations are not used in Stage II.

A.6 Evaluation Splits and Scoring

The evaluation suite contains ten benchmark families, divided into fifteen evaluation units with 19,913 examples in total (Table 7). Paired comparisons evaluate both conditions on the same examples in every unit. BLINK uses its labelled validation split, and MathVista uses test-mini. Mechanistic subsets and development probes are reported separately from these evaluation units.

Table 7: **Evaluation units and aggregation rules.** Scores are reported as percentages. The example count is the full evaluation denominator.

Evaluation unit	Examples	Aggregated metric
Single-image perception		
V*Bench, released test	191	Micro accuracy
CV-Bench, test-2D	1,438	Mean source accuracy
CV-Bench, test-3D	1,200	Micro accuracy
HR-Bench, 4K	800	Mean category accuracy
HR-Bench, 8K	800	Mean category accuracy
Multi-image reasoning		
BLINK, labelled validation	1,901	Mean subtask accuracy
MuirBench, released test	2,600	Micro accuracy
MMSI-Bench, non-circular set	1,000	Micro accuracy
MIMIC, Common test	1,000	Accuracy
MIMIC, Counting test	5,000	Accuracy
MIMIC, Odd-One-Out test	1,000	Accuracy
MIMIC, Listing test	1,000	Corpus micro set F1
Broader visual understanding		
MM-Vet v1, released set	218	Mean fractional judge credit
MathVista, test-mini	1,000	Accuracy
RealWorldQA, released test	765	Micro accuracy
Total	19,913	

Aggregation. CV-Bench-2D averages the accuracies of its two sources with equal weight. Each HR-Bench split averages four category accuracies, and BLINK averages its fourteen subtask accuracies. These macro averages differ from micro accuracy when components differ in size. MIMIC keeps its released metrics: accuracy for Common, Counting, and Odd-One-Out, and corpus-level micro set F1 for Listing, which is not an average of per-example F1 scores. The MIMIC Avg. column in the main tables is the arithmetic mean of these four scores and is not an official MIMIC metric. Unresolved predictions stay in the denominator and receive no credit.

Judged benchmarks. MM-Vet is scored by the mean fractional credit from a fixed DeepSeek-V4-Flash judge with a fixed JSON prompt. MathVista uses deterministic scoring for multiple-choice answers and the same judge for mathematical equivalence of free-form answers. These judges replace the online GPT-based judging of the original benchmarks.

Comparison protocol. All methods in Table 1 are evaluated on the same examples with the same released questions, answer options, image order, answer-format instructions, scoring rules, and judge. Each baseline uses its authors' released checkpoint, tools, stopping rules, and generation caps and is not retrained on our data; single-image methods receive an ordered contact sheet for multi-image inputs. Because baselines differ in backbone and training data, the main table also reports the backbone controls Native and Coarse and the same-data controls SFT and Stage $\mathrm{I}_{\mathrm{sft}}$, which share ReMAP's backbone and training data.

Prompts and generation settings. Native-protocol comparisons keep the released question, answer options, image order, and answer-format instruction. Baselines that accept only one image

receive an ordered contact sheet of all input images, marked with a dagger in Table 8. Generation budgets, stopping rules, and the memory-request cap are reported with each evaluation condition.

Additional baseline configurations. MINT-CoT and ICoT use their Qwen2-VL-7B implementations, and Pixel Reasoner uses its released Qwen2.5-VL-7B policy. ICoT keeps all input images but replays visual tokens from the last image through its released selection rule. Pixel Reasoner uses a 65,536-token runtime context for multi-image inputs, caps each crop at 1,024 visual tokens and each generation round at 1,024 output tokens, keeps a 5,120-token cap on initial images, and preserves its released crop tools and stopping rules. Q-Zoom caps its initial image or ordered contact sheet and each RoI at 576 visual tokens, with at most 576 output tokens. These caps apply to each input or crop, not to the whole trajectory.

Cross-backbone evaluation. In Table 2, Base denotes the original backbone, OFF the trained model with Global and Local requests disabled, and ReMAP the full model. The six benchmark columns are computed as $100n/N$ from the correct counts n, with $N = 191, 1200, 800, 1901, 2600, 1000$ in column order. For HR-Bench-8K and BLINK, these are micro accuracies rather than the official four-category and fourteen-subtask macro averages used in the main table. For every backbone, MIMIC Avg. is the mean over the four MIMIC subsets. Differences are computed from unrounded ratios and rounded to two decimals.

Paired scoring. The mechanistic comparisons share source examples, decoding settings, and scoring rules; each study specifies its memory interface, request positions, and content controls. Every expected example stays in the denominator, and an unresolved prediction is scored as wrong. We report aggregate accuracy and paired HELP/HARM counts. Confidence intervals use a sample-clustered bootstrap that resamples all states of an example together. The full-cohort visual-influence and static pixel-budget analyses use 5,000 percentile resamples with seed 20260828, and the state-conditioned utility and content-control analyses use 4,000 resamples with seed 20260830. Budget-sweep intervals are pointwise across the five tested values of K.

MIMIC components. Table 8 lists the four MIMIC component scores behind the MIMIC Avg. column of the main table.

Table 8: **MIMIC component scores.** A dagger denotes an ordered contact-sheet adaptation. VisMem is the public-code reproduction used in the main table.

Method	Common	Counting	Odd-One-Out	Listing	Mean
Native	62.60	44.98	60.40	12.69	45.17
Coarse	61.10	38.42	54.70	20.02	43.56
Q-Zoom†	51.90	37.44	54.80	25.00	42.29
DeepEyes†	52.60	40.38	53.20	10.90	39.27
Pixel Reasoner	64.50	43.34	30.00	41.46	44.83
ReGround	53.60	36.76	13.00	8.66	28.01
ICoT	48.80	15.38	54.90	0.55	29.91
MINT-CoT	54.30	30.06	18.50	33.33	34.05
DMLR†	38.70	16.14	17.70	22.41	23.74
VaLR	38.10	12.38	20.60	0.62	17.93
V-Reflection	85.10	31.04	58.20	38.28	53.16
VisMem	60.30	37.82	55.10	11.55	41.19
ReMAP	92.00	71.76	80.00	28.24	68.00

A.7 Visual Influence and State-Conditioned Utility

The two probes in this section measure how much the initial images still affect predictions as reasoning proceeds, and whether the value of renewed visual access depends on the reasoning state. Both use a fixed Q-Zoom-adapted Qwen2.5-VL-7B checkpoint.

A.7.1 Visual Influence Decay

For each example, we generate one deterministic reasoning trajectory. We then teacher-force the same text while replacing all input images with a same-layout patch-shuffled or blank version, and compare the next-token distributions p_t and $\widetilde{p}_t$ under the original and perturbed images:

$$f_t = \mathbf{1}[\arg\max p_t \neq \arg\max \widetilde{p}_t], \qquad d_t = \mathrm{JS}(p_t, \widetilde{p}_t). \tag{21}$$

The flip indicator f_t records a change of the most likely next token, and the Jensen–Shannon divergence d_t also records changes that keep that token. Positions are normalized by trajectory length and pooled into eight bins. Both statistics measure the size of the visual influence, not its direction. Figure 9 plots the flip rate across the eight bins, and Table 9 compares the first and last parts of each trajectory.

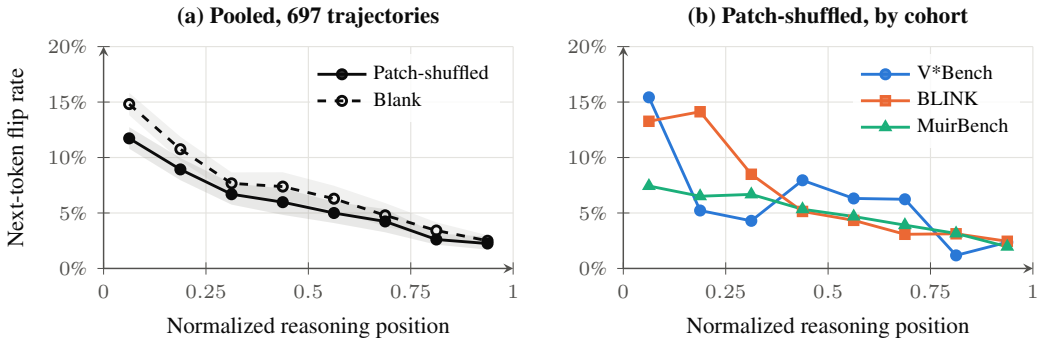


Figure 9: **Visual influence decays along the reasoning trajectory.** Rate at which the next-token top-1 decision changes when the input images are replaced, in eight bins of normalized reasoning position. (a) Patch-shuffled and blank images, pooled over V*Bench, BLINK, and MuirBench, with 95% bootstrap intervals. (b) Patch-shuffled images for each cohort.

Table 9: **Patch-shuffle visual influence across reasoning.** Eligible/selected counts the trajectories analyzed out of the sampled examples. First → last bin gives the flip rate in the first and eighth bins. Paired change is the per-example mean flip rate over the last quarter of the trajectory minus that over the first quarter, with a 95% bootstrap interval.

Cohort	Eligible / selected	First → last bin	Paired change
V*Bench	189/191	15.43% → 2.36%	−8.79 [−10.48, −7.00] pp
BLINK	254/256	13.27% → 2.45%	−10.88 [−12.15, −9.63] pp
MuirBench	254/256	7.43% → 1.96%	−4.50 [−5.48, −3.55] pp
Pooled	697/703	11.73% → 2.25%	−7.98 [−8.78, −7.21] pp

Blank images give the same pattern, with the flip rate falling from 14.82% to 2.50%. Under patch shuffling, the mean JS divergence falls from 0.0320 to 0.0039. The signed change in the log probability of the generated token is not consistent across cohorts: pooled, it is -0.0211 with interval $[-0.0450, 0.0026]$, and V*Bench has the opposite sign from BLINK and MuirBench. The initial images therefore influence later predictions less, but removing them does not uniformly lower the probability of the generated text.

A.7.2 State-Conditioned Utility

We insert visual content at selected states of the same deterministic trajectory and compare the gold-answer log probability. Candidate states are the prompt-only state and the positions at 20%, 40%, 60%, and 80% of the chain of thought. All arms share the textual prefix, the native visual user-turn interface, and a 16-token return budget. The pair of states in each contrast is chosen on development examples, and all estimates and intervals in Table 10 are computed on an independent confirmation split.

Table 10: **State and content contrasts in gold-answer log probability.** Late–early contrasts insert the same content at the later and the earlier selected state. Content contrasts compare the correct region or scene with a random wrong region, a matched wrong scene, or a shuffled scene. NS marks an interval that contains zero.

Cohort	Contrast	Estimate	95% CI
V*Bench ($n = 125$)	late–early correct-content utility	$+0.1771$	$[+0.1327, +0.2227]$
V*Bench ($n = 125$)	correct–random-wrong RoI	-0.0112	$[-0.0754, +0.0486]$ NS
CV-Bench-3D ($n = 152$)	late–early current-scene utility	$+0.0726$	$[+0.0405, +0.1050]$
CV-Bench-3D ($n = 152$)	current–matched-wrong scene	$+0.2265$	$[+0.1361, +0.3171]$
CV-Bench-3D ($n = 152$)	current–shuffled scene	$+0.0264$	$[-0.0358, +0.0882]$ NS

The same correct content is more useful at the later state on both V*Bench ($+0.1771$) and CV-Bench-3D ($+0.0726$), so the value of identical evidence depends on when it is accessed. On CV-Bench-3D, the current scene is also more useful than a matched wrong scene ($+0.2265$). The contrasts against a random wrong region on V*Bench and against a shuffled scene on CV-Bench-3D are not significant.

A.8 Memory-Content Requirements and Controls

The two probes in this section test which memory content helps: freshly encoded pixels of a region versus features from the initial encoding, and correct relational context across images versus content-matched controls. Both use the fixed checkpoint of Appendix A.7.

A.8.1 Same-Box Pixel Re-encoding

On the 191 V*Bench examples, both arms use the gold RoI, the same insertion point in the prefill, and the same number K of inserted tokens. The feature arm pools the RoI features from the initial encoding. The pixel arm crops the region from the source image, encodes it at high resolution, and compresses the result to K tokens. Figure 10 reports five values of K.

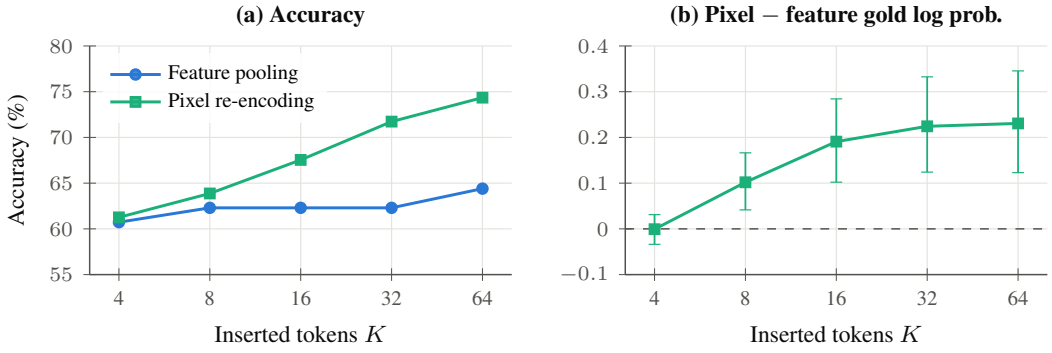


Figure 10: **Pixel re-encoding versus feature pooling on V*Bench.** Both arms return the gold RoI with K inserted tokens at the same insertion point on 191 examples. (a) Accuracy. (b) Pixel minus feature gold-answer log probability, with 95% CIs.

Pixel re-encoding matches the feature arm at $K = 4$ (-0.0007 in gold-answer log probability) and exceeds it from $K = 8$ onward. At $K = 32$, it raises the log probability by 0.2243 (95% CI $[0.1241, 0.3325]$) and accuracy from 62.30% to 71.73%. The high-resolution RoI pass uses 278.4 visual tokens on average, compared with 1,244.4 for the high-resolution full image. At $K = 16$, pixel memory also exceeds wrong-RoI, shuffled, blank, and zero controls. Splitting the examples into target-size tertiles gives the same direction in every tertile, with small differences between tertiles.

A.8.2 RELATIONAL-ROLE PROBE

We build 192 non-overlapping part–whole image pairs from COCO train2017 instance annotations, covering 800 unique images. In the role-changing condition, the focal image stays pixel-identical while its relation to the other image changes. In the appearance control, the appearance changes while the focal role stays fixed. Sensitivity is the total-variation distance between the model's answer distributions before and after a change, and fixation is the rate at which the model keeps predicting the focal option after the roles are swapped.

Table 11: **Relational-role probe.** The upper rows use total-variation sensitivity or fixation rate on 192 pairs. The lower rows compare the oracle relational context with each control in gold-answer log probability on the 71 selected late failure states. NS marks an interval that contains zero.

Contrast	Estimate	95% CI
Role change − appearance control	+0.0482	[+0.0254, +0.0725]
Late − early role sensitivity	+0.0039	[−0.0056, +0.0138] NS
Late − early fixation rate	+0.0182	[−0.0026, +0.0391] NS
Oracle − no context	+2.5797	[+2.1222, +2.9881]
Oracle − blank block	+2.4765	[+2.1251, +2.8061]
Oracle − zero block	+2.8028	[+2.5116, +3.0891]
Oracle − object list only	+1.3852	[+1.1268, +1.6433]
Oracle − another sample	+1.2927	[+1.0276, +1.5654]
Oracle − wrong role	+3.1493	[+2.7849, +3.5274]

The model is more sensitive to a role change than to an appearance change (+0.0482), so its predictions depend on cross-image relations and not only on the focal image (Figure 11(a)). Neither role sensitivity nor fixation changes significantly between early and late states.

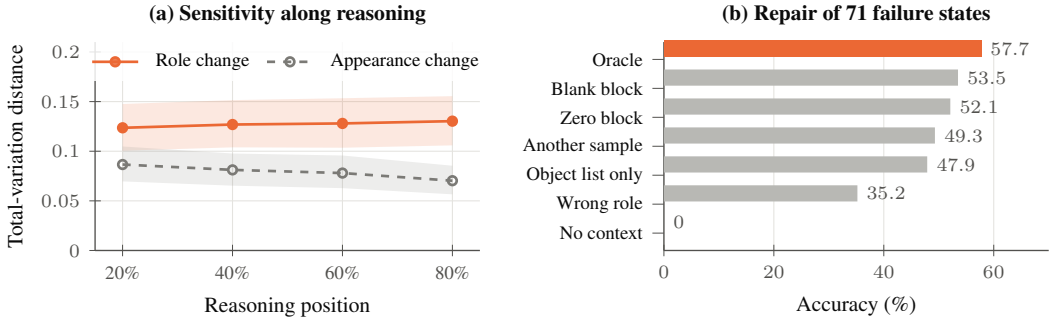


Figure 11: **Relational-role probe.** (a) Total-variation sensitivity to a role change and to an appearance change at four reasoning positions on 192 pairs, with 95% intervals. (b) Accuracy on the 71 selected late failure states with the oracle relational context and six controls; without context, every state is wrong by construction.

To test whether correct relational context can repair these errors, we select the 71 late states at which the model predicts the focal option for both the true and the role-swapped prefix, so that its role-swapped prediction is wrong. At each state, we insert either the oracle relational context or one of six controls: no context, a blank block, a zero block, an object list only, another sample's context, or the wrong role. Without context, all 71 states are wrong by construction. The oracle context repairs 41 of them, an accuracy of 57.7%, compared with 53.5%, 52.1%, 47.9%, 49.3%, and 35.2% for the blank, zero, object-list, other-sample, and wrong-role controls (Figure 11(b)). In gold-answer log probability, it exceeds every control (Table 11). Because the states are selected as failures and the relation comes from an oracle, this probe measures what correct relational content can contribute, not the effect of the learned Global memory.

A.9 Additional Ablations

Shared evaluation setting. The memory-curation and memory-access variants in Table 3 use the complete ReMAP weights and share the input examples, preprocessing, scoring, and decoding limits. Base is the Coarse backbone with at most 1,024 visual tokens per image. The table uses all 191 V*Bench, 1,200 CV-Bench-3D, 2,600 MuirBench, 1,901 BLINK, and 218 MM-Vet examples. V*Bench, CV-Bench-3D, MuirBench, and BLINK report micro accuracy, and MM-Vet reports the mean graded score. BLINK in this table therefore uses micro accuracy rather than the mean-subtask aggregation of Appendix A.6. Unresolved answers count as incorrect.

Memory-curation variants. w/o Memory masks both request tokens. w/ Global memory and w/ Local memory enable only one request type and keep the decoder adapter, request parameters, and answer path of ReMAP. When Global requests are disabled, Local still uses the cached Global memory Z_G, and the unused return capacity is not reassigned. Each Local request returns 32 latent tokens, including on multi-image examples. All variants use a 768-token ordinary-text budget, a 64-token answer grace period, and at most five memory requests. Memory requests do not count toward the text budget.

Training variants. SFT distills the teacher CoT without memory, and Stage $\mathrm{I}_{\mathrm{sft}}$ distills both the CoT and the memory. ReMAP adds access learning after Stage I. These variants differ in training. w/o Memory instead shares all of ReMAP's weights, including the Stage II decoder LoRA, and masks Global and Local requests only at inference. The gap between ReMAP and w/o Memory therefore measures the contribution of reasoning-time memory access on top of the full SFT and RL training with parameters held fixed; it is an inference-time ablation, not a model trained without memory.

Invocation controls. Fixed, Random, and Entropy use the complete ReMAP weights and change only when requests are issued at inference. Fixed requests Global at positions 0 and $\lfloor 0.7B \rfloor$ and Local at $\lfloor 0.3B \rfloor$ and $\lfloor 0.5B \rfloor$ of the text budget $B = 768$, that is, at positions 0, 230, 384, and 537. Random and Entropy consider a request after each generated delimiter, excluding position 0. Random issues a request with a fixed probability, following the random insertion of MemGen and VisMem (Zhang et al., 2026b; Yu et al., 2026); as in VisMem, it samples requests at delimiters. Entropy issues a request when the entropy of the next-token distribution over ordinary text tokens exceeds a fixed threshold. Memory tokens are excluded from this distribution, and EOS is retained. A triggered request chooses Global or Local with equal probability, as in VisMem. In all three controls, committing an answer or emitting EOS cancels later requests, and the five-request cap applies. Because each policy decides its own requests, the number of requests differs across policies.

A.10 Branch-Point Selection Diagnostic

This diagnostic compares four rules for choosing the state at which a root trajectory branches, with the policy frozen. The rules are the score $u(s_t, a_t)$ in Equation 3, labeled CRD in Figure 12; the route entropy $H\big(p_t(\mathrm{N}), p_t(\mathrm{G}), p_t(\mathrm{L})\big)$; uniform random selection among eligible states; and the entropy-change signal of ARPO (Dong et al., 2026a). The ARPO-style signal is the mean entropy of the last four ordinary tokens minus the mean entropy of the first four generated tokens. We branch at its maximum under the shared budget instead of using ARPO's stochastic branching gate. Unlike the route entropy, $u(s_t, a_t)$ conditions on the realized route and measures how likely a sibling is to depart from it.

All rules receive the same root trajectories and eligible candidate states. The cohort contains 192 training prompts, 32 from each of six ability categories: attribute, counting, existence, local detail, relation, and spatial reasoning. The prompts are selected without using rewards or correctness. Each prompt has four root trajectories, and each root yields a family of the original continuation and two siblings resampled from the selected state, giving 12 trajectories per rule and prompt.

Routing Yield@H16 is the fraction of families whose trajectories produce different sets of Global and Local requests within the first 16 generated positions after the branch. Joint EAY multiplies this diversity indicator by the mean absolute deviation of the family's final-answer rewards, so it is positive only when routing diversity comes with different rewards. Both metrics are averaged

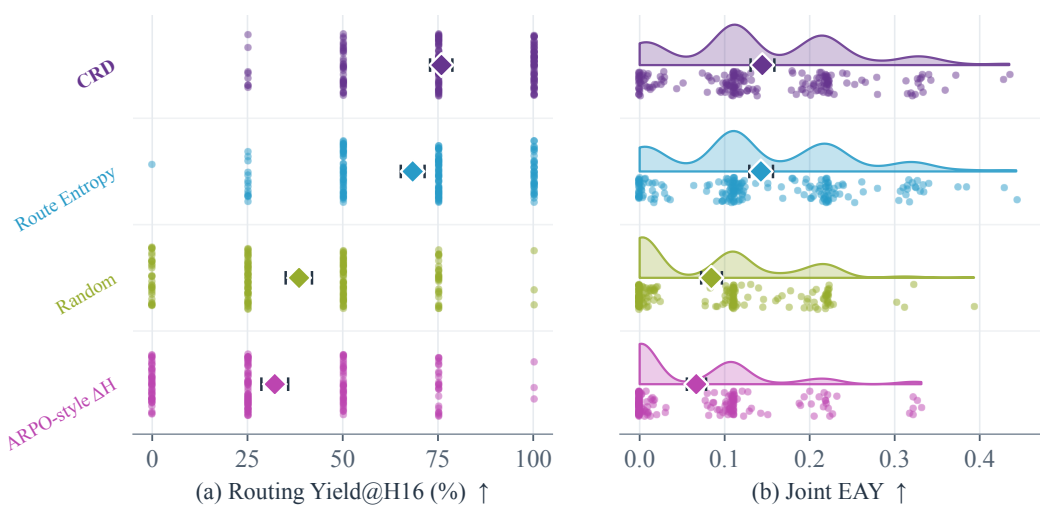


Figure 12: **Branch-point selection with a fixed trajectory budget.** (a) Routing Yield@H16, the fraction of branch families whose trajectories make different Global and Local requests within 16 positions after the branch. (b) Joint EAY, the reward variation that co-occurs with this routing diversity. Each point averages four root families of one prompt; diamonds and bars mark the mean and 95% bootstrap interval. All rules use the same roots, candidate states, and 12 trajectories per prompt on 192 prompts, with the policy frozen. ARPO-style ΔH is the fixed-budget adaptation of ARPO's entropy-change signal.

over the four families of a prompt, and we report prompt means. Confidence intervals use 10,000 prompt-level bootstrap resamples stratified by ability category, with seed 26090601 and the same resamples for all rules.

The score $u(s_t, a_t)$ gives the highest routing yield, 75.91% (95% CI $[72.92, 78.78]$), compared with 68.36% $[65.23, 71.48]$ for the route entropy, 38.54% $[35.03, 41.93]$ for random selection, and 32.16% $[28.65, 35.68]$ for the ARPO-style signal. Its Joint EAY of 0.144 $[0.131, 0.159]$ is close to that of the route entropy, 0.143 $[0.129, 0.157]$, and above those of random selection, 0.084, and the ARPO-style signal, 0.067. Route-based selection therefore exposes more alternative memory decisions at the branch, and these alternatives come with larger reward differences than those found by random or entropy-change selection.

A.11 Complete Qualitative Case Traces

Figure 13 shows the complete reasoning of the two main-text cases, with and without memory.

A.12 Visual-Context Accounting

Figure 6(a) compares the visual tokens that enter the decoder sequence with the performance in Table 1. For each question, we count the initial image tokens and every crop token or memory latent returned during reasoning. A repeated Global request inserts the cached block again, and its latents are counted again. Ordinary text tokens, request and end markers, and representations used only inside the vision encoder or the memory weavers are excluded. The count therefore measures the visual context of the decoder, not total inference compute.

The comparison uses the questions measured for all five methods: 78 V*Bench, 60 CV-Bench-2D, and 107 MuirBench questions. Means are unweighted over these questions and include questions without memory requests. The performance lines use the full-benchmark scores of the main table. ReMAP starts from at most 1,024 visual tokens per image and returns 32 latents per Local request and $4N + 4$ latents per Global request. Its counts come from the recorded memory insertions of its evaluation run. For Q-Zoom and DeepEyes on MuirBench, the counts sum the passes over each input image.

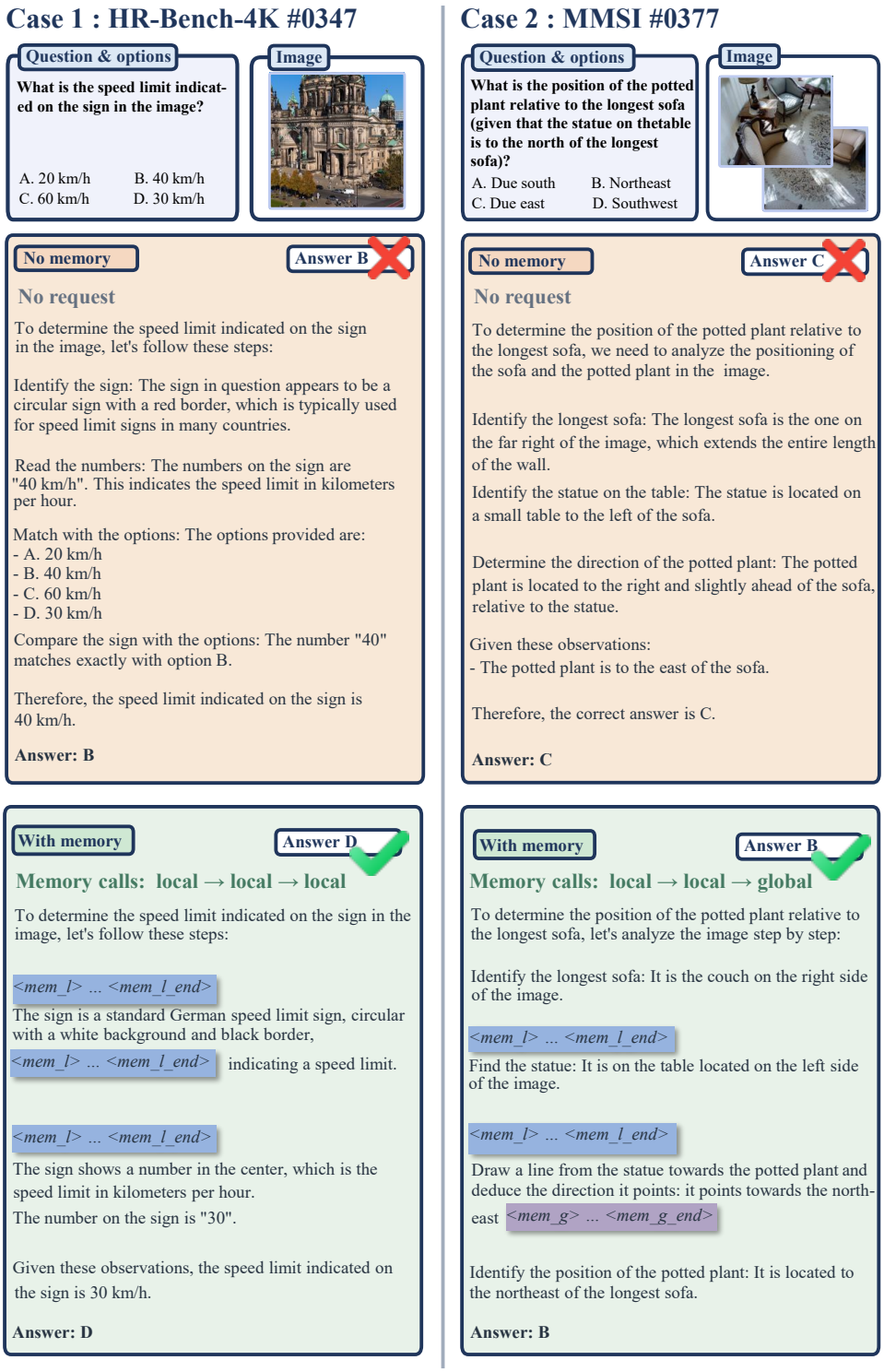


Figure 13: **Complete paired reasoning traces.** Cases 1 and 2 expand panels (a) and (b) of Figure 5 with the full recorded reasoning of the paired no-call and memory-enabled trajectories. Each memory return is marked by its request and end tokens. Case 1 (HR-Bench-4K #0347): without memory, the model reads 40 km/h and answers B; after three Local calls, it reads 30 km/h and answers D. Case 2 (MMSI #0377): without memory, the model answers due east (C); after two Local calls and one Global call, it answers northeast (B).

Table 12 lists the per-question means. ReMAP adds 16.0, 18.7, and 29.3 tokens per question through memory returns, and it uses 76.8%, 51.0%, and 52.3% fewer visual tokens than Native on V*Bench, CV-Bench-2D, and MuirBench, respectively.

Table 12: **Visual tokens entering the decoder sequence per question.** Means over the questions shared by all five methods, with the question count given for each benchmark. Added counts crop tokens and memory latents returned during reasoning. Totals are computed before rounding.

Benchmark	Method	Initial	Added	Total
V*Bench ($n = 78$)	Native	4,417.9	0.0	4,417.9
	Q-Zoom	1,008.0	331.8	1,339.8
	DeepEyes	4,417.9	35.9	4,453.8
	VisMem	4,417.9	0.0	4,417.9
	ReMAP	1,008.0	16.0	1,024.0
CV-Bench-2D ($n = 60$)	Native	906.0	0.0	906.0
	Q-Zoom	450.6	175.3	625.9
	DeepEyes	906.0	208.1	1,114.1
	VisMem	906.0	0.0	906.0
	ReMAP	425.3	18.7	444.0
MuirBench ($n = 107$)	Native	7,099.5	0.0	7,099.5
	Q-Zoom	3,515.4	500.8	4,016.2
	DeepEyes	7,099.5	2,097.6	9,197.0
	VisMem	7,099.5	0.0	7,099.5
	ReMAP	3,359.9	29.3	3,389.2

A.13 LIMITATIONS

We note three limitations of ReMAP that we hope to address in future work. First, ReMAP operates within a fixed memory and access budget: each trajectory permits at most five memory requests, each Local request re-encodes at most three selected images, and each Global block has a fixed $4N + 4$ layout. These limits keep computation low but may omit evidence when an input contains many images or when a task requires extended visual search. Second, Twig is trained only in Stage I on teacher-reviewed ROI targets and receives no further training dedicated to localization. When its box misses the relevant region, the subsequent zoom-in contributes less useful detail, so we expect a stronger localizer, for example one trained with dedicated grounding data, to further improve Local memory. Third, the memory modules are attached at fixed positions of the backbone: the Source Selector reads layer 12, and Twig reads layer 18. Internal layer sweeps selected these taps from the quality of the tapped representations, but we have not yet studied how placing the components at other layers affects end-to-end performance, and we leave this analysis to future work.