VisMem: Latent Vision Memory Unlocks Potential of Vision-Language Models

Supplementary Material

7. Theoretical Foundations

As the mainstream position in anthropological cognitive psychology since the 20th century, short-term memory and long-term memory are two distinct storage systems that can be differentiated based on their functional and neural underpinnings [3, 38]. Specifically, the *Dennis Norris Theory* [38] proposes that short-term memory requires processing new visual information, temporarily storing multiple tokens, and enabling variable signals. It relies neurologically on vision-specific brain regions, *e.g.*, the visual cortex and the posterior superior temporal lobe associated with verbal short-term memory), exhibiting visual dominance; long-term memory, however, centers on abstract semantic representations and relies on semantic-related brain regions like the medial temporal lobe and mid-temporal lobe.

Thus, we propose a framework termed VisMem to invoke dual short and long latent memory during the token-by-token autoregressive generation. Aligned with *Dennis Norris Theory* [38], we instantiate these roles in a VLM backbone via latent vision memory invocation and latent vision memory formation, which together produce distinct short and long latent memory tokens and integrate them into the generation stream of the model.

8. Methodology Details

8.1. Query Builder

As described in Sec. 3.3, the we initialize a lightweight transformer-based encoder as memory builder $\mathcal{B}$. We feed the concatenated memory query $\mathbf{Q}$ and hidden states of vision and output $\mathbf{H}$ into the builder to encoder query as memory hook (see Eq. (5)). The transformer-based builder has L layers of encoders, the output process of the ℓ layer could be summarized as:

$$\mathrm{SA}(x) = \mathrm{SM}\left(\frac{(xW_q)(xW_k)^\top}{\sqrt{d_k}} + M\right)(xW_v), \tag{9}$$

$$x^\ell = \mathrm{FF}\left(\mathrm{LN}\left(x^{\ell-1} + \mathrm{SA}\left(\mathrm{LN}\left(x^{\ell-1}\right)\right)\right)\right) + x^{\ell-1}, \tag{10}$$

where we simplify the input sequence to x, and SM, MHA, FF, LN denote the softmax, multi-head self-attention, feed-forward layer, layer normalization operations, respectively. In addition, M is the mask which only allows attention from memory query $\mathbf{Q}$ to hidden states $\mathbf{H}$, and blocks the reverse direction:

$$M_{ij} = \begin{cases} -C, & i < K \text{ and } j \geq K \\ 0, & otherwise \end{cases}, \tag{11}$$

where $C \gg 0$ is constant, thus the attention is close to $-\infty$.

8.2. Training Recipe

As mentioned in Sec. 3.4, we design a two-stage training pipeline: at the first stage, the main objective is to optimize the memory formation process (see Eq. (7)); at the second stage, the main objective is to optimize the memory invocation (see Eq. (8)). We update the models based on reinforcement learning, *i.e.*, GRPO strategy [43]. Specifically, for each instruction-vision pair (I, V), the policy model $\mathcal{P}$ generates a group of G distinct candidate trajectories, termed as $\mathcal{T} = \{\tau_1, \ldots, \tau_G\}$. For each trajectory, we utilize a $S(\cdot)$ to quantify the performance. Then, a group-relative baseline is calculated via averaging and standardizing all trajectories within the candidate group G:

$$\overline{S} = \frac{1}{G}\sum_{i=1}^{G} S(\tau_i), \hat{S} = \sqrt{\frac{1}{G}\sum_{i=1}^{G}\left(S(\tau_i) - \overline{S}\right)^2}. \tag{12}$$

Consequently, the group-relative advantage of each trajectory could be formulated as:

$$\hat{A} = \frac{S(\tau) - \overline{S}}{\hat{S} + \epsilon}. \tag{13}$$

At the **Stage I**, the reinforcement learning optimizes the memory formation process, whose final objective function is:

$$\begin{aligned} \mathcal{J}_{GRPO}^{stage1}(\phi) = \mathbb{E}_{\tau, \mathbf{M}_{s/l}, \mathbf{Q}} \Bigg[& \frac{1}{G}\sum_{i=1}^{G} \\ & \min\left(\rho_i(\phi)\hat{A}_i, \mathrm{clip}\left(\rho_i(\phi), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right) \Bigg] \\ & - \beta D_{\mathrm{KL}}\left[\pi_\tau^\phi \| \pi_{\mathrm{ref}}^\phi\right], \end{aligned} \tag{14}$$

where ϵ controls the group-relative advantage $\hat{A}$, β regulates the KL divergence penalty, and the updated policy parameters $\pi^\phi = \pi^\phi(\mathbf{Q} \mid \mathbf{H}) \cdot \pi^\phi(\mathbf{M}_{s/l} \mid \mathbf{Q})$.

At the **Stage II**, the reinforcement learning optimizes the memory invocation process, whose final objective function is:

$$\begin{aligned} \mathcal{J}_{GRPO}^{stage2}(\theta) = \mathbb{E}_{\tau, x} \Bigg[& \frac{1}{G}\sum_{i=1}^{G} \\ & \min\left(\rho_i(\theta)\hat{A}_i, \mathrm{clip}\left(\rho_i(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\right) \Bigg] \\ & - \beta D_{\mathrm{KL}}\left[\pi_\tau^\theta \| \pi_{\mathrm{ref}}^\theta\right]. \end{aligned} \tag{15}$$

9. Experiment Details

9.1. Training Data

During the two-stage training procedure, we use the same training data to optimize both the memory invocation and memory formation in the latent vision memory system. Initially, we include the training split dataset of the selected benchmarks and retain their original data division. For benchmarks without a training phase, we use them solely for evaluation. Additionally, we incorporate the Visual CoT [42] and Mullberry [71], improving the reasoning abilities.

9.2. Benchmarks

To comprehensively evaluate the performance of the selected baselines, we involve 12 benchmarks, consisting of 5 benchmarks of understanding, 4 benchmarks of reasoning, and 3 benchmarks of generation:

- MMStar [7] is a high-quality vision-centric benchmark meticulously curated by human experts. This benchmark assesses 6 core capabilities across 18 detailed axes of visual understanding.
- MMVet [76] establishes 6 core visual understanding capabilities and investigates 16 critical integrations derived from their combinations. It uses an evaluator tailored for open-ended outputs.
- MMT [73] consists of carefully curated multi-choice visual questions, covering 32 core meta-tasks and 162 subtasks within the field of visual understanding.
- BLINK [15] reconstructs 14 classic computer vision tasks into multiple-choice questions. Each question is paired with either single or multiple images and supplemented with visual prompting.
- MuirBench [57] covers 12 diverse multi-image tasks, which involve 10 categories of multi-image relations. Each standard instance is paired with an unanswerable variant that differs only minimally in semantics.
- MMMU [79] comprises meticulously curated visual questions sourced from college exams, quizzes, and textbooks spanning 30 subjects and 183 subfields, which focus on advanced reasoning grounded in domain-specific knowledge.
- LogicVista [67] evaluates general logical cognition abilities across 5 logical reasoning tasks, which encompass 9 distinct capabilities. Each question is annotated with the correct answer and the human-written reasoning behind the selection.
- MathVista [59] unifies the challenges of heterogeneous mathematical and visual tasks, which are curated from math-oriented multimodal datasets.
- MV-Math [62] is a dataset comprising mathematical problems, integrating multiple images interleaved with text, and detailed annotations. It features multiple-choice, free-form, and multi-step questions across 11 subject areas at 3 difficulty levels.
- HallBench [19] consists of images paired with questions, designed by human experts to assess the hallucination level of generation.
- MultiTrust [82] covers five primary aspects: truthfulness, safety, robustness, fairness, and privacy, evaluating the trustworthiness of generation.
- MMVU [34] encompasses 12 categories, and designs evaluation metrics that measure the quality and error degree of generation.

9.3. Baselines

We select a total of 16 baselines, including the vanilla model [4], 5 direct training paradigms: SFT, Visual-RFT [35], VLM-R1 [44], Vision-R1 [26], and PAPO [66]; 5 image-level paradigms: Sketchpad [24], GRIT [13], PixelReasoner [48], DeepEyes [87], and OpenThinkImg [49]; 4 token-level paradigms: Scaffold [28], ICoT [16], MINT-CoT [8], and VPT [75]; and 1 latent space paradigm: Mirage [70].

Here, VLM-R1 [44] and Vision-R1 [26] follow the main GRPO [20] paradigm based on VLMs. To assess the effectiveness of different methods, our VisMem is trained on Qwen-2.5-VL-7B [4]. For strategies initially implemented on other base models, *e.g.*, GPT-4o [27] and Qwen2-VL [60], we transfer them to Qwen2.5-VL-7B [4] for fair comparison. Besides, we maintain identical training datasets across most counterparts; however, for those three methods with specially curated datasets, we follow their original settings. Namely, Mirage [70] requires additional labeled training images, so we follow its original training dataset; GRIT [13] uses a tailored training process with designed data; and MINT-CoT [8] curates high-quality mathematical samples with grids and annotations.

9.4. Implementations

The configurations and implementations of the experiments include three main parts: the core hyperparameters, the parameters of the LoRA adapter, and the parameters we use during training. The configurations and implementations of the experiments are listed in Tab. 4.

10. Additional Results

10.1. Benchmark Subset Results towards Visual Sub-capacities

To precisely identify the capability boundaries and advantages of our VisMem, rather than relying solely on overall scores to judge its quality, we evaluate the results of subsets of MuirBench [57] and LogicVista [67] benchmarks. We select 9 subsets of the former benchmark, including: counting, grounding, matching, scene, difference, cartoon,

Table 4. Configurations of parameters.

Configurations	Parameters	Values	
Core	K	8	
	N_s	4	
	N_l	8	
LoRA [23]	$rank$	16	
	α	32	
	$drop\_out\_rate$	0.1	
	$target\_module$	$[q\_proj, v\_proj]$	
Training		**Stage I**	**Stage II**
	$batch\_size$	8	
	$epoch$	2	
	$warmup\_ratio$	0.2	0.1
	$num\_iteration$	1	
	$learning\_rate$	$5e^{-5}$	$1e^{-5}$
	$optimizer$	AdamW [36]	
	$scheduler$	Cosine	
	$group\_size$	16	
	$clip\_ratio$	0.2	
	$kl\_penalty\_coefficient\ \beta$	0.015	0.030
	$target\_kl\_per\_token$	0.03	0.05
	$penalty\_intensity\ \alpha$	-	0.3

Table 5. Results on 9 selected subsets of MuirBench [57]. We compare our VisMem with the second and third best scored counterparts, and separately use the short or long latent memory to assess the improvements of each.

Method	Counting	Grounding	Matching	Scene	Difference	Cartoon	Diagram	Geographic	Retrieval
Vanilla [4]	44.1	34.2	80.9	70.5	53.2	52.9	82.4	53.7	76.1
VLM-R1 [44]	52.5	38.1	83.6	73.5	58.1	55.1	86.8	56.7	79.4
Vision-R1 [26]	53.8	39.2	**84.5**	73.1	57.4	57.2	87.4	57.9	78.9
VisMem (Short Memory)	**61.3**	<u>49.4</u>	82.7	72.1	<u>58.9</u>	54.0	<u>88.9</u>	61.8	<u>87.5</u>
VisMem (Long Memory)	46.3	42.6	83.2	<u>74.3</u>	55.4	<u>59.4</u>	87.4	<u>62.7</u>	78.3
VisMem	<u>60.8</u>	**52.3**	<u>84.0</u>	**76.2**	**60.6**	**59.7**	**90.1**	**65.5**	**89.8**

diagram, geographic, and retrieval. While in the latter benchmark, we also select 10 subsets, including 5 reasoning skills: inductive, deductive, numerical, spatial, and mechanical, and 5 capacities: patterns, puzzles, OCR, graphs, and tables. It is worth noting that the selected subsets are only part of the benchmark, thus, the average values of the 10 subsets are not the results of the benchmarks.

As listed in Tab. 5, compared with VLM-R1 [44] and Vision-R1 [26], our VisMem achieves the best results on 7 subsets and ranks second on the remaining two subsets. Specifically, it has a generalized enhancement of at least 5% over the base model. Besides, VisMem improves the performance the vanilla model by 16.7% / 18.2% / 11.8% / 13.7% on the counting, grounding, geographic, and retrieval subtasks, vastly exceeding the second-best counterpart by 7.0-13.1%. These results indicate that our latent vision memory system significantly promote the fine-grained visual cognition and perception of the base VLMs.

As presented in Tab. 6, our VisMem outperforms two baseline models, *i.e.*, VLM-R1 [44] and Vision-R1 [26], by achieving the top performance across 8 subsets. Specifically, it delivers a generalized improvement of no less than 7% over the base model. Notably, on inductive, deductive, graph-based, and table-based sub-tasks, VisMem surpasses the vanilla model by 14.8%, 14.8%, 18.4%, and 21.1%, respectively, which exceeds the second-ranked model by a

Table 6. Results on 10 selected subsets (5 reasoning skills and 5 capabilities) of LogicVista [67]. We compare our VisMem with the second and third best scored counterparts, and separately use the short or long latent memory to assess the improvements of each.

Method	Inductive	Deductive	Numerical	Spatial	Mechanical	Patterns	Puzzles	OCR	Graphs	Tables
Vanilla [4]	44.6	45.0	39.7	37.9	48.7	30.1	32.5	41.6	34.4	36.8
VLM-R1 [44]	53.7	52.7	45.8	44.1	57.3	35.8	42.8	<u>49.0</u>	46.5	52.6
Vision-R1 [26]	53.5	51.4	<u>46.7</u>	44.8	**58.9**	<u>36.5</u>	<u>43.6</u>	**49.7**	48.2	53.8
VisMem (Short Memory)	49.8	50.1	44.7	<u>45.2</u>	54.3	35.2	42.0	47.6	<u>50.3</u>	<u>54.1</u>
VisMem (Long Memory)	<u>57.5</u>	<u>58.4</u>	42.8	40.0	52.0	35.7	38.0	47.4	48.9	51.3
VisMem	**59.4**	**59.8**	**46.9**	**47.2**	<u>57.4</u>	**38.9**	**44.6**	48.5	**52.8**	**57.9**

substantial margin of 5.3–7.1%. These results demonstrate that our latent visual memory system delivers contextualized semantic knowledge, thereby enhancing visual reasoning and robust generation capabilities.

10.2. Cross-domain Generalization

To evaluate the cross-domain generalization capability of our model, we train it exclusively on general datasets, namely, Visual CoT [42] and Mullberry [71]), to verify whether latent visual memory can be transferred to unseen domains. As shown in Tab. 7 and Fig. 7, our method demonstrates superior performance, which exhibits a smaller performance drop than the fully trained model across all four selected benchmarks, confirming strong cross-domain generalization. Despite being trained on only two datasets, our method achieves a significant performance improvement of 9.1–20.5% across the four benchmarks, with a mere 2% performance gap relative to the fully trained model. When compared to other baselines, it still maintains a performance lead of 3.4% / 6.7% / 2.7% / 4.7% across the four evaluations, respectively.

In general, the image-level, token-level, and latent space paradigms suffer from smaller performance degradation, whereas the direct training paradigm exhibits inferior generalization ability. For example, VLM-R1 [44] experiences a 5.3% performance drop; by contrast, this value is only 2.1% for OpenThinkImg [49], 1.1% for MINT-CoT [8], and 2.3% for our method. These results indicate that while direct training optimizations notably improve performance on specific tasks, they compromise generalization ability to some extent.

10.3. Catastrophic Forgetting Mitigation

To assess the extent of catastrophic forgetting, we conducted continual learning experiments with our VisMem and other baselines. As presented in Tab. 8 and Fig. 8, our method effectively mitigates forgetting of earlier tasks. It consistently achieves the best performance at each stage, demonstrating strong robustness against catastrophic forgetting. Following four-stage sequential continual training, it retains 72.1% performance on MMVet [76], outperforming 68.4% of DeepEyes [87] and 67.0% of Mirage [70].

While the direct training paradigm significantly improves performance on specific tasks, it adapts to new tasks via direct updates to core parameters. This introduces conflicts when parameter update directions contradict the storage of existing knowledge, compounded by a lack of constraints from prior knowledge. Consequently, in stage 3, the performance of most direct training methods even falls below that of the vanilla model. In contrast, methods such as OpenThinkImg [49] and our proposed VisMem exhibit stronger knowledge retention and forward transfer capabilities. For instance, in stage 3, training on additional datasets further improves their performance on MMVet [76].

10.4. Versatility across Various Base Models

As presented in Tab. 2 and Fig. 11, we incorporate our latent visual memory paradigm into 9 base models, including Qwen2.5-VL-3B/7B/32B [4], LLaVA-OV-1.5-4B/8B [1], and InternVL-3.5-4B/8B/14B/38B [63]. Our VisMem consistently enhances the visual capabilities of all base models, spanning 3B to 38B parameter sizes across three VLM families. For the widely used medium-sized models (*i.e.*, 7B or 8B parameter models), our latent visual memory delivers substantial performance gains, which brings a 6.3–23.1% improvement across all benchmarks for Qwen2.5-VL-7B [4], a 5.5–20.2% improvement for LLaVA-OV-1.5-8B [1], and a 4.8–17.6% improvement for InternVL-3.5-8B [63], respectively.

Furthermore, in most benchmarks, smaller-parameter base models yield greater performance gains than their medium- or large-sized counterparts. This phenomenon may stem from an imbalance in task difficulty, which makes it more challenging for models with higher baseline scores to achieve further improvements. In contrast, larger models exhibit more significant gains in dense reasoning benchmarks: the integration of latent visual memory overcomes bottlenecks in visual reasoning by providing fine-grained visual evidence and semantic knowledge. Notably, this model-agnostic approach, independent of specific model ar-

Table 7. Results of various models with full training datasets and partial datasets (Visual CoT [42] and Mulberry [71]), and evaluated across four benchmarks.

Method	MMVet		MuirBench		MV-Math		MultiTrust	
	Full	Part	Full	Part	Full	Part	Full	Part
Vanilla [4]	66.0		57.4		18.9		64.8	
SFT	67.5	65.8	58.7	57.2	22.8	21.2	67.0	65.4
Visual-RFT [35]	70.5	65.3	62.9	57.8	26.5	24.2	70.7	66.0
VLM-R1 [44]	73.0	67.7	63.8	59.0	34.6	32.1	69.9	66.1
Vision-R1 [26]	71.7	68.4	64.0	59.8	38.7	35.6	72.6	67.1
PAPO [66]	69.8	68.6	56.7	56.4	34.8	32.8	67.7	66.4
DeepEyes [87]	70.5	67.9	63.0	60.6	31.5	27.9	72.6	68.5
OpenThinkImg [49]	71.6	69.5	61.7	59.7	28.0	25.9	74.0	68.3
ICoT [16]	67.9	67.1	57.0	56.4	30.8	28.3	69.1	68.4
MINT-CoT [8]	69.5	68.4	58.9	57.8	39.2	36.4	71.4	70.2
Mirage [70]	71.8	70.2	59.0	57.2	35.4	33.1	66.1	64.0
VisMem (Ours)	**75.1**	**72.9**	**69.8**	**66.4**	**41.4**	**39.1**	**77.0**	**74.9**

Table 8. Results of various models on MMVet [76] with four-stage continual learning. Stage 0: MMVet [76]; Stage 1: BLINK [15], and MuirBench [57]; Stage 2: LogicVista [67], and Math-V [59]; Stage 3: MultiTrust [82], and MMVU [34].

Method	Stage 0	Stage 1	Stage 2	Stage 3	Original
Vanilla [4]	66.0				
SFT	71.4	70.6	62.3	60.1	67.5
Visual-RFT [35]	74.0	72.2	67.3	65.7	70.5
VLM-R1 [44]	77.8	74.1	66.4	66.9	73.0
Vision-R1 [26]	76.9	74.0	66.1	66.3	71.7
PAPO [66]	75.0	74.5	63.4	62.9	69.8
DeepEyes [87]	74.1	74.6	68.9	68.4	70.5
OpenThinkImg [49]	76.2	74.7	66.5	67.9	71.6
ICoT [16]	71.9	71.3	67.1	64.7	67.9
MINT-CoT [8]	72.4	71.8	65.8	66.2	69.5
Mirage [70]	**79.1**	77.8	68.7	67.0	71.8
VisMem (Ours)	78.6	**78.9**	**71.3**	**72.1**	**75.1**

chitectures or structures, bolsters the prospects for broad practical application.

10.5. Ablation Study

The vanilla model establishes a baseline characterized by the shortest inference time and highest speed across all benchmarks, yet exhibits the lowest performance. This confirms that latent vision memory is indispensable for enhancing task performance. For the random memory invocation variants, increasing the invocation probability (25%–100%) results in longer inference time and reduced speed. Performance peaks at a 75% probability before declining, indicating that excessive memory invocation impairs efficiency without yielding additional performance benefits. Ablation studies of the short-term and long-term memory components reveal task-specific advantages: the short-term memory component outperforms on MuirBench [57] and MultiTrust [82], while the long-term component demonstrates superior performance on MV-Math [62]. Notably, the complete VisMem framework achieves the highest performance across all benchmarks, validating the value of integrating dual-component vision memory for balanced and robust visual capacities.

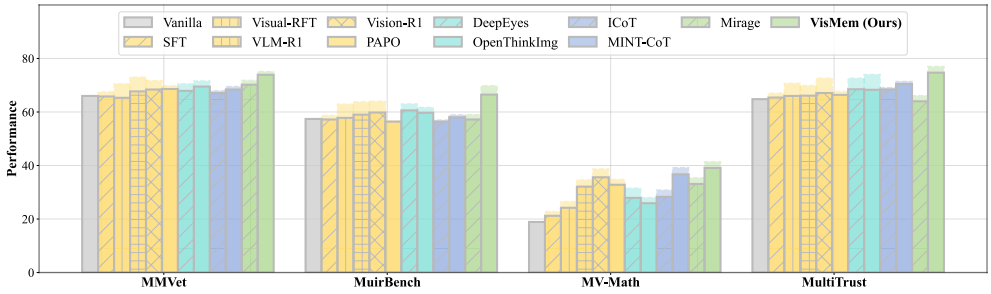


Figure 7. Results of various models of the cross-domain generalization study. Models are only trained on Visual CoT [42] and Mulberry [71], and are evaluated on four benchmarks.

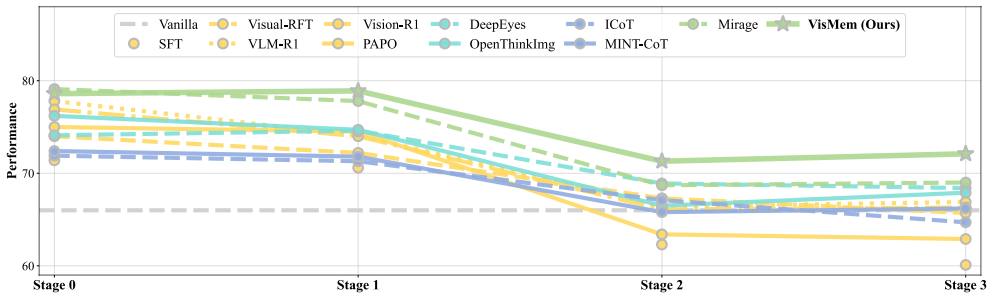


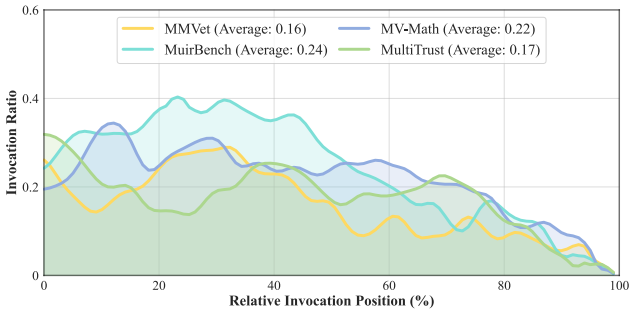
Figure 8. Results of four-stage continual learning on MMVet [76]. The model is sequentially trained on each training data combination (Stage 0 → Stage 1 → Stage 2 → Stage 3). Stage 0 only includes MMVet [76] as training data, while Stage 1, 2, 3 add data targeting visual understanding [15, 57], reasoning [59, 67], and generation [34, 82].

Table 9. Ablations of latent vision memory invocation and dual vision memory formation. Following [81], "Random Invocation" denotes that the latent memory is inserted into the output sequence with a certain probability when outputting delimiter symbol tokens, and short or long latent memory is inserted with equal probability. When only utilizing short or long latent memory, we directly skip the formation of the specific memory if invocation tokens are predicted and continue the process of decoding.

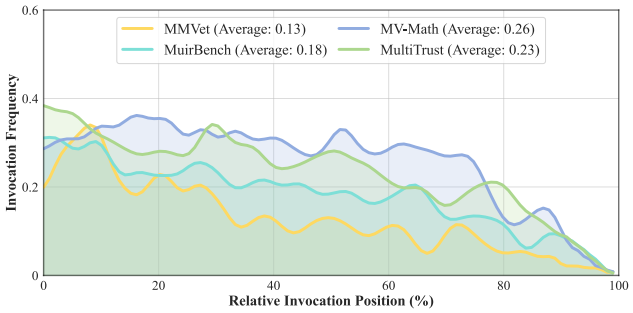
Ablation	**MMVet**			**MuirBench**			**MV-Math**			**MultiTrust**		
	Time	Speed	Perf.	Time	Speed	Perf.	Time	Speed	Perf.	Time	Speed	Perf.
Vanilla	**0.76**	**1.32**	66.0	**3.79**	**0.26**	57.4	**5.47**	**0.18**	18.9	**3.62**	**0.28**	64.8
Random Invocation (25%)	0.80	1.25	69.2	3.94	0.25	59.4	8.79	0.11	29.8	6.14	0.16	69.4
Random Invocation (50%)	0.83	1.20	71.9	4.12	0.24	63.2	11.68	0.09	26.1	8.62	0.12	68.5
Random Invocation (75%)	0.86	1.16	73.6	4.27	0.23	62.7	14.78	0.07	21.9	10.11	0.10	63.7
Full Invocation (100%)	0.88	1.14	73.4	4.43	0.23	56.0	17.87	0.06	17.5	13.43	0.07	62.6
Short-term Memory	0.79	1.27	71.5	4.00	0.25	65.6	7.64	0.12	29.6	4.96	0.20	73.6
Long-term Memory	0.81	1.23	69.4	3.95	0.25	60.2	7.61	0.12	36.1	4.80	0.21	69.8
Complete VisMem (Ours)	0.84	1.19	**75.1**	4.10	0.24	**69.8**	7.87	0.13	**41.4**	5.85	0.17	**77.0**

10.6. Analysis of Latent Vision Memory

We visualize the invocation ratio and relative invocation position, as presented in Fig. 5 and 9: the former illustrates benchmark-specific differences between the two memory components, while the latter depicts type-specific variations across the four benchmarks. In addition, as reported in Tab. 5 and 6, the short- and long-term latent visual memory components exhibit task-specific advantages for different visual sub-tasks. For instance, the short-term memory provides supplementary visual information to support enhanced visual understanding, such as counting, grounding, and visual retrieval. By contrast, the long-term memory encodes contextualized semantic knowledge, which strengthens complex visual reasoning. These results reveal that our



(a) Short Memory Invocation



(b) Long Memory Invocation

Figure 9. Results of memory invocation ratio and relative position across four benchmarks. The former denotes the proportion of invoked samples to all samples, while the relative position denotes the position in the whole output sequence when the invocation occurred. We apply gaussian smoothing to the curves to highlight their main trends.

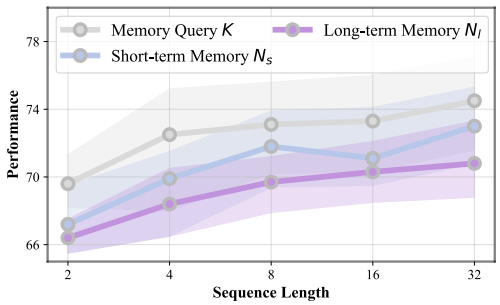


Figure 10. Results of sensitivity analysis on the sequence length of memory query K, short- and long-term memory N_s and N_l.

Table 10. Results of different length of memory query K.

K	MMVet	MuirBench	MV-Math	MultiTrust
Vanilla	66.0	57.4	18.9	64.8
2	69.6	66.0	34.7	71.9
4	72.5	68.9	40.6	74.8
8	73.1	69.8	41.1	77.0
16	73.3	70.0	**41.4**	77.7
32	**74.5**	**70.3**	40.9	**78.2**

proposed VisMem dynamically adjusts invocation position and frequency according to task characteristics, thereby balancing efficiency and performance.

10.7. Sensitive Analysis of Sequence Lengths

We conduct an analysis on MMVet [76] focused on the lengths of three key sequences: the memory query K, the short-term latent visual memory N_s, and the long-term latent visual memory N_l. It is observed that as the lengths of these three sequences increase from 2 to 32, model performance improves accordingly, but this is accompanied by increased computational costs.

Table 11. Results of different length of short latent vision memory N_s and the length of long latent vision memory N_l across four benchmarks.

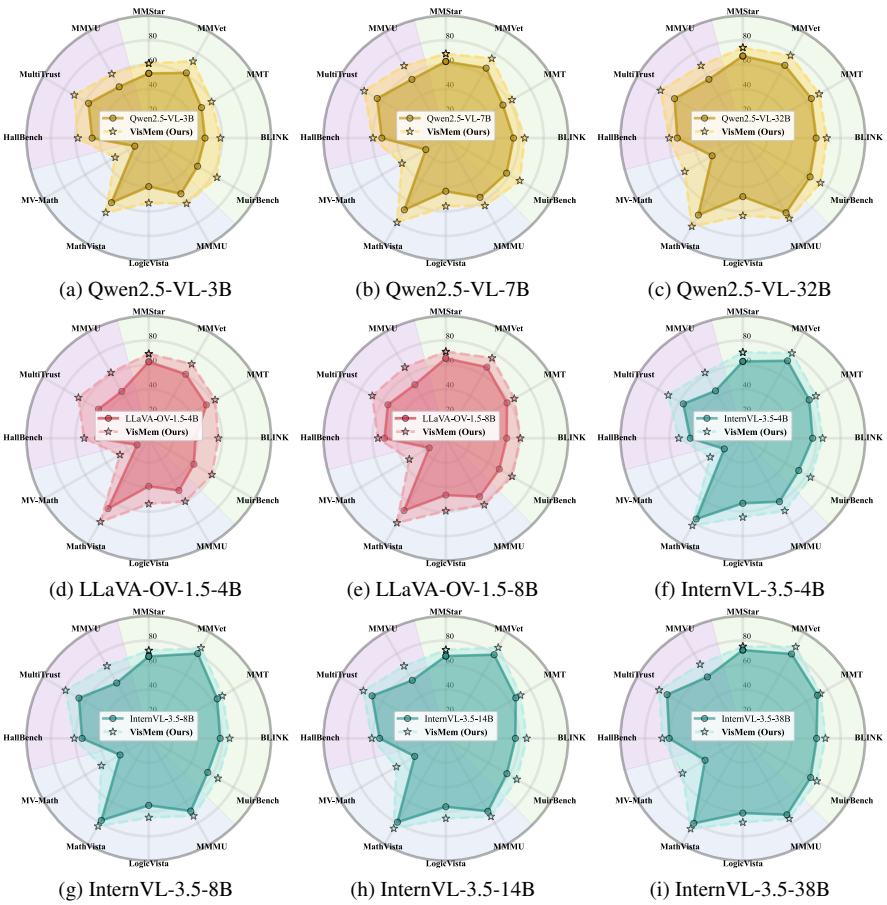
N_s	N_l	MMVet	MuirBench	MV-Math	MultiTrust
Vanilla		66.0	57.4	18.9	64.8
2	-	67.2	63.7	28.2	69.3
4	-	69.9	64.6	31.5	71.4
8	-	71.8	65.2	33.8	73.4
16	-	71.1	67.8	34.0	73.3
32	-	73.0	69.1	34.4	72.7
-	2	66.4	60.3	29.3	71.0
-	4	68.4	61.8	32.4	72.8
-	8	69.7	63.0	33.5	74.2
-	16	70.3	63.4	34.8	74.9
-	32	70.8	63.1	35.5	75.3
8	16	**75.1**	**69.8**	**41.1**	**77.0**

10.8. Inference Efficiency

As presented in Tab. 12 and the bubble plots in Fig. 6, we compare the average inference time, average inference speed, and task performance across the four benchmarks. Our approach achieves an optimal performance-efficiency balance, with minimal additional time overhead. For instance, image-level paradigms exhibit nearly twice the inference time of the vanilla model, resulting in significant latency and substantial inference overhead. In contrast, our VisMem introduces only controllable computational latency increments, ranging from 8.2% to 43.8% relative to the vanilla model, which are on par with those of other direct training and token-level paradigms.

Table 12. Average inference time per sample (seconds), average inference speed (samples / seconds), and task performances across four benchmarks on various methods. Perf. indicates Performance.

Method	MMVet			MuirBench			MV-Math			MultiTrust		
	Time	Speed	Perf.	Time	Speed	Perf.	Time	Speed	Perf.	Time	Speed	Perf.
Vanilla [4]	0.76	1.32	66.0	**3.79**	**0.26**	57.4	**5.47**	**0.18**	18.9	**3.62**	**0.28**	64.8
SFT	**0.75**	**1.33**	67.5	3.82	0.26	58.7	6.35	0.16	22.8	3.68	0.27	67.0
Visual-RFT [35]	0.76	1.32	70.5	3.81	0.26	62.9	5.66	0.17	26.5	3.65	0.27	70.7
VLM-R1 [44]	0.77	1.30	73.0	3.83	0.26	63.8	7.88	0.13	34.6	3.69	0.27	69.9
Vision-R1 [26]	0.77	1.30	71.7	3.83	0.26	64.0	8.42	0.12	38.7	3.71	0.27	72.6
PAPO [66]	0.76	1.32	69.8	3.81	0.26	56.7	6.74	0.15	34.8	3.68	0.27	67.7
Sketchpad [24]	2.39	0.42	64.5	8.90	0.11	52.8	9.10	0.11	24.6	5.47	0.18	66.2
GRIT [13]	0.80	1.25	67.8	4.07	0.25	51.0	8.45	0.12	22.4	4.06	0.25	67.3
PixelReasoner [48]	1.45	0.69	67.1	7.34	0.14	60.5	9.96	0.10	25.9	5.60	0.18	69.9
DeepEyes [87]	3.21	0.31	70.5	8.46	0.12	63.0	11.72	0.09	31.5	6.14	0.16	72.6
OpenThinkImg [49]	3.68	0.27	71.6	8.69	0.12	61.7	10.38	0.10	28.0	6.43	0.16	74.0
Scaffold [28]	0.83	1.20	67.0	4.35	0.23	52.9	7.01	0.14	21.0	3.88	0.26	68.5
ICoT [16]	0.97	1.15	67.9	4.57	0.22	57.0	8.94	0.11	30.8	4.20	0.24	69.1
MINT-CoT [8]	0.81	1.23	69.5	4.18	0.24	58.9	7.89	0.13	39.2	4.03	0.25	71.4
VPT [75]	2.98	0.34	70.8	9.63	0.10	63.5	9.59	0.10	34.7	5.79	0.17	64.7
Mirage [70]	0.86	1.16	71.8	4.02	0.25	59.0	7.71	0.13	35.4	3.82	0.26	66.1
VisMem (Ours)	0.84	1.19	**75.1**	4.10	0.24	**69.8**	7.87	0.13	**41.4**	3.85	0.26	**77.0**



(a) Qwen2.5-VL-3B (b) Qwen2.5-VL-7B (c) Qwen2.5-VL-32B

(d) LLaVA-OV-1.5-4B (e) LLaVA-OV-1.5-8B (f) InternVL-3.5-4B

(g) InternVL-3.5-8B (h) InternVL-3.5-14B (i) InternVL-3.5-38B

Figure 11. Results on different base models.