VCRM: EMPOWERING EFFICIENT MEMORY FOR VISION-LANGUAGE MODELS WITH VISUAL CONTEXT RECORD MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

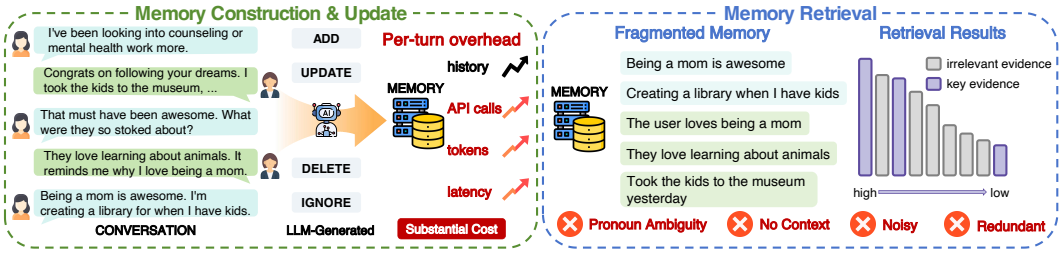
Although large language models excel at complex reasoning tasks, effectively leveraging continuously growing long-term historical information remains a critical challenge. Existing memory-augmented methods primarily rely on storing and retrieving historical information in textual form. However, as interaction history continues to grow, memory content faces a dual bottleneck of ever-increasing context overhead and key evidence dispersed across voluminous retrieval results. To address these issues, we propose Visual Context Record Memory (VCRM), which achieves significant token compression with stable performance gains. VCRM consists of two synergistic components: (1) Hierarchical Evidence Prospecting (HEP), which filters redundant information through tiered confidence routing and adaptive cliff pruning; (2) Discriminative Semantic Rendering (DSR), which renders critical evidence into two-dimensional visual records with non-uniform information density, enabling key evidence to be preserved more effectively under constrained token budgets. We evaluate our method on three widely adopted benchmarks spanning long-term conversational memory and multi-hop question answering, i.e., LoCoMo, LongMemEval-S, and HotpotQA. Results show that VCRM improves downstream task performance while substantially reducing token consumption, validating the effectiveness of our approach. Under constrained budgets, VCRM maintains near-full performance through visual memory, demonstrating strong robustness under compression. Our code is available at `https://anonymous.4open.science/r/VCRM-270A/`.

1 INTRODUCTION

Large Language Models (LLMs) (Liu et al., 2024; Achiam et al., 2023; Team et al., 2025) have demonstrated capabilities in single-turn reasoning and generation tasks, yet in dynamic interactive environments that rely on long-horizon historical context, such as open-domain dialogues and multi-turn task planning, their stateless nature presents a fundamental bottleneck (Bai et al., 2024; Tavakoli et al., 2026). As interaction rounds accumulate, the volume of historical information rapidly exceeds the constraints of context windows, causing models to frequently forget early critical information or generate hallucinatory conclusions when queried about historical facts (Wang et al., 2026a; Hsieh et al., 2024). To address this challenge, memory-augmented frameworks have emerged. Their core paradigm involves storing historical interaction fragments in external memory banks, dynamically retrieving relevant memories during inference and injecting them into the context, thereby extending the model's long-horizon perception capabilities (Hu et al., 2026). Such frameworks have become essential infrastructure for building continually adaptive agents (Zhang et al., 2025b).

However, as depicted in Fig. 1, existing frameworks still exhibit several key limitations that constrain their efficiency and accuracy in large-scale long-horizon interactions: (1) *Linear proliferation of token budgets*: existing frameworks predominantly rely on textual memory representations. During the memory construction phase, each dialogue turn requires LLM invocations for memory addition and relevant memory updates to maintain freshness. This turn-by-turn construction continuously consumes substantial API quotas and token resources while introducing significant cumulative latency (Xia et al., 2026). During the inference phase, text-based retrieval results proliferate linearly with history length, further escalating inference costs (Rasmussen et al., 2025). (2) *Diffusion of*

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) Linear proliferation of token budgets (b) Diffusion of key evidence

Figure 1: Two key limitations in existing memory frameworks: (a) linear proliferation of token budgets. (b) diffusion of key evidence.

key evidence: to ensure high recall, existing retrieval strategies often return numerous redundant entries; these noise entries submerge critical evidence (Zhou et al., 2025). Simultaneously, fragmented memory entries inevitably dilute or lose key details of the original context (Yan et al., 2026). In lengthy contexts, critical information gradually diffuses among redundant noise, making precise localization of required evidence extremely difficult.

To address the above challenges, we propose **V**isual **C**ontext **R**ecord **M**emory (**VCRM**), a training-free visual memory enhancement framework. The core insight of VCRM is not to simply convert textual memory into images, but to integrate retrieval filtering, semantic distillation, and visual encoding into a complete visual memory pipeline. As illustrated in Fig. 2, VCRM achieves efficient memory management through two synergistic mechanisms: (1) Hierarchical Evidence Prospecting (HEP), which adaptively filters redundant information at the source and localizes critical evidence; (2) Discriminative Semantic Rendering (DSR), which distills the filtered evidence into structured facts and renders them into two-dimensional visual records with non-uniform information density, keeping critical facts legible after compression through spatial layout and visual salience. This design frees memory representation from linear textual sequences while retaining the information-density gain. We evaluate our method on three widely adopted benchmarks spanning long-term conversational memory and multi-hop question answering. Experimental results show that, under settings that fully unleash the memory capacity of each method, VCRM achieves superior performance while substantially reducing memory token consumption. Moreover, when the memory budget is aggressively compressed to as few as 64 tokens, VCRM maintains near-full performance through the compression capability of visual memory, highlighting the robustness advantage and potential of visual representations under extreme compression.

In summary, our principal contributions are threefold:

- We propose VCRM, a training-free visual memory enhancement framework that integrates retrieval filtering with visual rendering, achieving high-ratio token compression while obtaining significant performance gains.
- We design Hierarchical Evidence Prospecting (HEP), a tiered confidence routing mechanism that filters redundant information at the source and alleviates the evidence diffusion problem in lengthy contexts.
- We propose Discriminative Semantic Rendering (DSR), a visual memory encoding mechanism that combines semantic fact extraction with density-differentiated rendering, enabling key evidence to be preserved effectively under constrained token budgets.

2 Related Work

Memory-Augmented Agents. Memory-augmented agents utilize external memory to manage long-horizon histories (Hu et al., 2025). Existing work primarily follows two directions: (i) structured storage and updating, including Mem0's incremental fact extraction (Chhikara et al., 2025), A-Mem's card-based networks (Xu et al., 2025), MemoryOS's system-level abstractions (Kang et al., 2025), and LightMem's hierarchical offline consolidation (Fang et al., 2026); and (ii) retrieval and policy optimization, such as True Memory's retrieval-centric reorganization (Adler & Zehavi, 2026) and RL-based memory construction in Mem-α (Wang et al., 2025) and MemAgent (Yu et al., 2026). However, these approaches rely solely on textual representations, requiring frequent LLM invocations for distillation during storage while suffering from granularity asymmetry between fine-grained

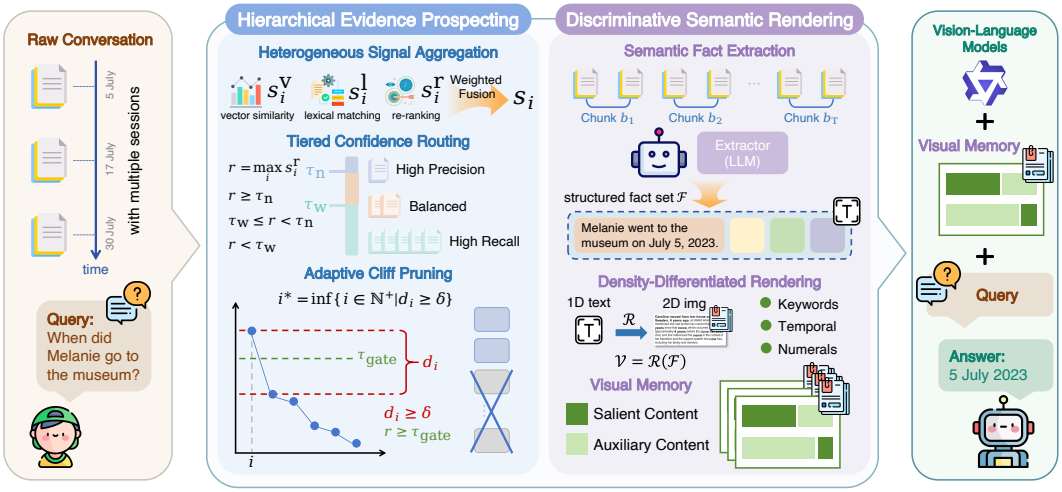


Figure 2: An overview of our VCRM framework, consisting of two key components: (1) Hierarchical Evidence Prospecting (HEP), which filters redundant information via tiered routing and adaptive pruning; (2) Discriminative Semantic Rendering (DSR), which encodes structured evidence into visual records via fact extraction and differentiated rendering.

distillation and coarse-grained retrieval, leading to linear scaling of API costs and token consumption with history length and limited information density.

Visual-Text Compression. Visual-Text Compression (VTC) reduces token consumption by rendering text as two-dimensional images. DeepSeek-OCR (Wei et al., 2025) and DeepSeek-OCR 2 (Wei et al., 2026) pioneer specialized visual encoders for high-ratio compression, while Glyph (Cheng et al., 2026) optimizes rendering configurations through genetic algorithms, and VIST (Xing et al., 2025) and LensVLM (Xie et al., 2026) explore selective compression and dynamic expansion. Recent works extend VTC to long-horizon reasoning and agent scenarios: MemOCR (Shi et al., 2026) employs layout-aware visual memory encoding, AgentOCR (Feng et al., 2026) enables autonomous compression rate control, and VTC-R1 (Wang et al., 2026b) leverages reinforcement learning to optimize real-time rendering during iterative reasoning. However, these approaches either rely on time-consuming reinforcement learning training, lack specialized mechanisms for memory retrieval, or proceed without pre-filtering, resulting in visual memory cluttered with redundant information.

3 Method

3.1 Preliminaries

Given a long-horizon historical context $\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$, where each h_i denotes the i-th historical unit, and a current query q, the objective of long-horizon memory-augmented question answering is to leverage a vision-language model f_{VLM} capable of generating accurate answers a based on historical context. Since N typically exceeds the model's context window constraints, directly encoding $\mathcal{H}$ is infeasible, necessitating external memory mechanisms (Yu et al., 2026).

We formulate this task as a two-stage process: Hierarchical Evidence Prospecting (HEP) and Discriminative Semantic Rendering (DSR). First, the HEP module retrieves and filters a candidate segment subset $\mathcal{C} = \psi_{\text{HEP}}(\mathcal{H}, q; \phi) \subset \mathcal{H}$ relevant to query q via configuration parameters ϕ, where $|\mathcal{C}| = K$ denotes the dynamically determined candidate budget, and $K \ll N$. Subsequently, the DSR module first distills $\mathcal{C}$ into a structured fact set $\mathcal{F} = f^{\text{ext}}_{\text{VLM}}(\mathcal{C}, q)$ via Semantic Fact Extraction, and then renders it into a visual memory image $\mathcal{V} = \mathcal{R}(\mathcal{F})$ through the lightweight renderer $\mathcal{R}$ in Density-Differentiated Rendering. Finally, the vision-language model f_{VLM} generates the predicted answer $a = f_{\text{VLM}}(\mathcal{V}, q)$ solely from the visual memory and the query.

The proposed framework exhibits two core characteristics: (1) **Adaptive budget control**, where K adjusts dynamically with query complexity through routing mechanisms rather than remaining fixed; (2) **Non-uniform information encoding**, wherein the visual memory $\mathcal{V}$ employs differentiated spatial layouts with critical evidence rendered at higher visual salience, decoupling answer-stage visual token consumption from history length.

3.2 HIERARCHICAL EVIDENCE PROSPECTING

We propose the Hierarchical Evidence Prospecting (HEP) module, which adaptively distills critical evidence from long-horizon contexts, tackling the evidence diffusion problem at its source. HEP not only filters redundant information via tiered confidence routing but also preserves evidence fidelity through dynamic truncation. It consists of three key components: (1) Heterogeneous Signal Aggregation, (2) Tiered Confidence Routing, and (3) Adaptive Cliff Pruning.

Heterogeneous Signal Aggregation. For each historical unit $h_i \in \mathcal{H}$, we first compute three complementary relevance scores conditioned on the query q: a semantic vector similarity score s_i^{v}, a lexical matching score s_i^{l}, and a relevance re-ranking score s_i^{r}. Then these are composed into a heterogeneous signal vector $\mathbf{v}_i = [s_i^{\mathrm{v}}, s_i^{\mathrm{l}}, s_i^{\mathrm{r}}]^\top$.

Since the numerical distributions and scales of these scores differ substantially, direct linear fusion would allow high-variance signals to dominate the aggregated result. To address this, we apply z-score normalization to map each score into a unified standard score space:

$$\hat{s}_i^\star = \frac{s_i^\star - \mu_\star}{\sigma_\star}, \quad \star \in \{\mathrm{v}, \mathrm{l}, \mathrm{r}\}, \tag{1}$$

where $\mu_\star$ and $\sigma_\star$ are the mean and standard deviation of each score over the full candidate set, respectively. To prevent numerical instability, the normalized score for that channel is set to zero when $\sigma_\star < \epsilon$, where ϵ is a small constant. We then adopt a weighted linear fusion strategy to obtain a comprehensive confidence score:

$$s_i = \mathbf{w}^\top \hat{\mathbf{v}}_i = w_{\mathrm{v}} \cdot \hat{s}_i^{\mathrm{v}} + w_{\mathrm{l}} \cdot \hat{s}_i^{\mathrm{l}} + w_{\mathrm{r}} \cdot \hat{s}_i^{\mathrm{r}}, \tag{2}$$

where $\mathbf{w} = [w_{\mathrm{v}}, w_{\mathrm{l}}, w_{\mathrm{r}}]^\top \in \mathbb{R}^3$ is the fusion weight vector, and $\hat{\mathbf{v}}_i = [\hat{s}_i^{\mathrm{v}}, \hat{s}_i^{\mathrm{l}}, \hat{s}_i^{\mathrm{r}}]^\top$ denotes the normalized signal vector.

Finally, all N historical units are sorted in descending order of s_i, yielding the full ranked sequence $\mathcal{C}_{\text{ranked}}$, which serves as the input for the next stage.

Tiered Confidence Routing. Given the full ranked sequence $\mathcal{C}_{\text{ranked}}$, we need to select the most relevant subset $\mathcal{C}_{\text{tier}}$ of candidate segments. Fixed-size truncation faces two limitations: (1) the number of required segments varies substantially across queries; (2) due to semantic ambiguity, the ranking confidence of ground-truth segments is inherently uncertain. Consequently, a uniform truncation strategy cannot simultaneously guarantee high recall and high compression.

We circumvent this limitation by introducing a tiered confidence routing mechanism. The maximum relevance re-ranking score $r = \max_i s_i^{\mathrm{r}}$ is adopted as a query-difficulty indicator, a confidence signal decoupled from the fused ranking. The candidate retention size K_{tier} is then determined by a piecewise function governed by two thresholds τ_{w} and τ_{n}:

$$K_{\text{tier}} = K_{\mathrm{w}} + \Delta_{\mathrm{m}} \mathbb{H}(r - \tau_{\mathrm{w}}) + \Delta_{\mathrm{n}} \mathbb{H}(r - \tau_{\mathrm{n}}), \tag{3}$$

where $\mathbb{H}(\cdot)$ denotes the Heaviside step function, $K_{\mathrm{w}} > K_{\mathrm{m}} > K_{\mathrm{n}} > 0$ represent the predefined candidate quotas for the wide, mid, and narrow tiers, respectively, with $\Delta_{\mathrm{m}} = K_{\mathrm{m}} - K_{\mathrm{w}}$ and $\Delta_{\mathrm{n}} = K_{\mathrm{n}} - K_{\mathrm{m}}$. When r is low, indicating indistinct matching scores across candidates and difficult retrieval, a broader scope is necessary to cover dispersed evidence; when r is high, the model exhibits sufficient confidence, and a narrower scope suffices for efficient compression. Finally, we select the top-K_{tier} historical units from $\mathcal{C}_{\text{ranked}}$ to form the intermediate candidate set $\mathcal{C}_{\text{tier}}$.

Adaptive Cliff Pruning. Tiered routing determines the intermediate candidate set $\mathcal{C}_{\text{tier}}$, yet redundancy may persist within $\mathcal{C}_{\text{tier}}$. In high-confidence scenarios where r is sufficiently large, the fused scores of top-ranked candidates tend to exhibit a pronounced lead over their successors, forming a discernible cliff in the score distribution, whereas tail candidates contribute marginally to the final answer. To this end, we introduce an adaptive cliff pruning strategy that identifies structural breakpoints within the fused score sequence of $\mathcal{C}_{\text{tier}}$, thereby eliminating low-value redundancy.

Specifically, given the fused score sequence $\{s_{(1)}, s_{(2)}, \ldots, s_{(K_{\text{tier}})}\}$ sorted in descending order from $\mathcal{C}_{\text{tier}}$, we denote by d_i the relative drop between adjacent scores. A cliff point is identified as the first position where this relative decay surpasses a threshold δ:

$$i^* = \inf\{i \in \mathbb{N}^+ \mid d_i \geq \delta\}, \tag{4}$$

and the final candidate count is then determined by clamping the detected breakpoint within the tier-prescribed bounds:

$$K = \max\big(\min(i^*, K_{\text{tier}}), \kappa\big), \tag{5}$$

where δ denotes the cliff detection threshold and κ represents a floor constraint that safeguards against excessive pruning that may discard critical segments. This pruning mechanism is activated exclusively under high-confidence conditions where $r \geq \tau_{\text{gate}}$, with τ_{gate} being a predefined confidence threshold; otherwise, the full set of tier-determined candidates $\mathcal{C}_{\text{tier}}$ is retained. Through this mechanism, the HEP module refines the candidate set, ultimately outputting the refined candidate segment set $\mathcal{C}$ with size K, eliminating low-value redundancy while preserving recall fidelity.

3.3 Discriminative Semantic Rendering

We present the Discriminative Semantic Rendering (DSR) module, which transcribes the HEP-filtered candidate evidence into a two-dimensional visual memory image with non-uniform information density, enabling compact and legible memory representation under constrained token budgets. DSR consists of two cascading components: (1) Semantic Fact Extraction, which distills noisy raw text into structured key facts; and (2) Density-Differentiated Rendering, which allocates visual priority based on semantic salience and drives final answer generation.

Semantic Fact Extraction. The candidate session set $\mathcal{C}$ output from HEP retains the full multi-turn conversational context, unlike the fragmented entries in existing memory frameworks, thereby providing sufficient context for downstream understanding. Nevertheless, it still contains substantial query-irrelevant content; rendering it directly would result in visual clutter. To address this, we perform semantic fact extraction to distill $\mathcal{C}$ into a temporally organized structured fact set $\mathcal{F}$.

Specifically, given that candidate segments c_i are lengthy and may exceed the effective attention window of the underlying LLM, $\mathcal{C}$ is partitioned into ordered text blocks $\mathcal{B} = \{b_1, b_2, \ldots, b_T\}$ according to a predefined token budget. For each block b_i, the vision-language model $f_{\text{VLM}}^{\text{ext}}$ is prompted to extract structured facts pertinent to query q:

$$\mathcal{F} = \{f_i^j \mid f_i^j = f_{\text{VLM}}^{\text{ext}}(b_i, q), b_i \in \mathcal{B}\}, \tag{6}$$

where f_i^j denotes the j-th fact extracted from the i-th text block. Compared with the original candidate segments, $\mathcal{F}$ effectively achieves substantial compression while faithfully preserving core information and its temporal context.

Density-Differentiated Rendering. The distilled structured fact set $\mathcal{F}$ is still organized as a textual sequence, whose token consumption grows with the scale of facts. To present critical information compactly and legibly under constrained token budgets, we introduce a lightweight renderer $\mathcal{R}$ that transcribes $\mathcal{F}$ into a single two-dimensional visual memory image $\mathcal{V} = \mathcal{R}(\mathcal{F})$. During rendering, semantically salient content—such as query-relevant keywords, temporal expressions, and key numerals—is rendered with larger font sizes and bold styling in visually prominent regions, while auxiliary details are compactly arranged with smaller fonts in peripheral areas. At query time, the vision-language model generates the final answer $a = f_{\text{VLM}}(\mathcal{V}, q)$ by reading solely from the rendered visual memory image and the query, without accessing the original text evidence. This non-uniform information density encoding ensures that critical evidence remains legible under subsequent downsampling compression.

4 Experiments

4.1 Experimental Settings

Datasets and Metrics. We evaluate our approach on three benchmark datasets. LoCoMo benchmark (Maharana et al., 2024) covers four types of memory reasoning tasks, LongMemEval-S benchmark (Wu et al., 2024) extends to six fine-grained memory dimensions, and HotpotQA benchmark (Yang et al., 2018) focuses on multi-hop question answering over multiple documents. Specifically, for HotpotQA, to evaluate long-context performance, the original context is extended to 30K tokens by padding with distractor documents. For LoCoMo and LongMemEval-S, we report J-score, F1, and BLEU scores. The J-score is computed via the LLM-as-a-Judge paradigm, which assesses the comprehensiveness and logical coherence of each response, and each question is judged three

Table 1: Performance comparison on the LoCoMo and LongMemEval-S benchmarks. Best results are highlighted in **bold**, and second-best results are underlined.

	LoCoMo			LongMemEval-S			Overall			
Method	J	F1	BLEU	J	F1	BLEU	J	F1	BLEU	Tokens
Qwen3-VL-4B-Instruct										
NaiveRAG	57.34	31.19	25.76	57.80	39.41	34.44	57.57	35.30	30.10	6001
Mem0 (2025)	44.61	21.13	17.42	46.92	7.64	3.49	45.77	14.39	10.46	2453
LightMem (2026)	71.95	42.07	35.62	65.60	7.80	3.41	68.78	24.94	19.52	2250
True Memory (2026)	77.40	27.90	19.62	71.40	46.63	40.35	74.40	37.27	29.99	9632
VCRM (Ours)	**82.86**	**46.47**	**39.26**	**75.80**	**54.56**	**49.39**	**79.33**	**50.52**	**44.33**	**377**
Qwen2.5-VL-7B-Instruct										
NaiveRAG	60.39	23.92	17.71	38.40	20.83	14.91	49.40	22.38	16.31	6093
Mem0 (2025)	51.56	23.56	19.49	45.81	10.57	4.90	48.69	17.07	12.20	2372
MemOCR (2026)	45.65	27.18	22.91	18.80	11.98	10.66	32.23	19.58	16.79	981
LightMem (2026)	65.06	**37.09**	**30.81**	48.00	11.86	5.80	56.53	24.48	18.31	1655
True Memory (2026)	71.62	25.34	18.20	42.20	27.20	21.39	56.91	26.27	19.80	8761
VCRM (Ours)	**75.39**	35.14	29.39	**56.40**	**37.22**	**30.58**	**65.90**	**36.18**	**29.99**	**326**

times independently with the final verdict determined by majority voting. For HotpotQA, we report F1, EM, and SubEM scores.

Implementation Details. We adopt Qwen3-VL-4B-Instruct (Yang et al., 2025) and Qwen2.5-VL-7B-Instruct (Qwen et al., 2025) as the vision-language backbones, both served via vLLM (Kwon et al., 2023) with temperature set to 0 for reproducibility. The lexical matching score is obtained via BM25, the semantic embedding via Qwen3-Embedding-0.6B (Zhang et al., 2025a), and the relevance re-ranking via bge-reranker-v2-m3. The accuracy evaluation is conducted with Qwen3-235B-A22B-Instruct-2507 as an LLM judge, with the evaluation protocol following True Memory (Adler & Zehavi, 2026). *Please refer to Appendix and code for more details.*

4.2 Overall Comparison

We evaluate VCRM against five competitive baselines across three benchmark datasets, with the main results presented in Table 1 and Table 2. LoCoMo (Maharana et al., 2024) comprises 1,540 non-adversarial questions spanning four reasoning categories: multi-hop, temporal, open-domain, and single-hop. LongMemEval-S (Wu et al., 2024) consists of 500 questions across six fine-grained memory dimensions: Knowledge-Update, Multi-Session, Single-Session-Assistant (SS-Assistant), Single-Session-Preference (SS-Preference), Single-Session-User (SS-User), and Temporal. HotpotQA (Yang et al., 2018) contains 1,928 multi-hop reasoning questions, for which we report F1, EM, and SubEM scores. The compared baselines include text-based methods NaiveRAG, Mem0 (Chhikara et al., 2025), LightMem (Fang et al., 2026), True Memory (Adler & Zehavi, 2026), and the visual method MemOCR (Shi et al., 2026) (only 7B released).

As shown in Table 1, under the unlimited-budget setting, all methods use their complete memory without any truncation or compression. With the 4B backbone, VCRM achieves consistent leading results across all three metrics on both benchmarks: it attains the highest J-score of 82.86% on LoCoMo, outperforming the strongest baseline True Memory by 5.46%, and 75.80% on LongMemEval-S, surpassing it by 4.40%. F1 and BLEU scores are also the highest, with 46.47% and 39.26% on LoCoMo, and 54.56% and 49.39% on LongMemEval-S, respectively. With the 7B backbone, VCRM likewise attains the highest J-scores on both benchmarks, 75.39% on LoCoMo and 56.40% on LongMemEval-S, with F1 and BLEU reaching the best or second-best levels. Meanwhile, VCRM not only improves performance but also substantially reduces the memory token consumption at the answer stage: on average only 377 tokens with the 4B backbone and 326 tokens with the 7B backbone, approximately one sixth and one third of the second-lowest baselines of 2,250 and 981 tokens, and far below the other baselines. On HotpotQA, as shown in Table 2, VCRM achieves the highest scores on all three metrics with the 4B backbone, with F1 of 79.16%, EM of 62.50%, and SubEM of 81.95%; with the 7B backbone, it attains the best or second-best results. These results jointly demonstrate that the visual memory paradigm of VCRM generalizes well, maintaining strong downstream task performance while substantially reducing token consumption.

Table 2: Performance comparison on the HotpotQA benchmark. Best results are highlighted in **bold**, and second-best results are underlined.

Method	F1	EM	SubEM
Qwen3-VL-4B-Instruct			
NaiveRAG	57.21	44.14	58.82
LightMem	59.23	49.36	61.83
Mem0	68.85	53.12	71.88
VCRM (Ours)	**79.16**	**62.50**	**81.95**
Qwen2.5-VL-7B-Instruct			
NaiveRAG	53.62	42.27	56.07
LightMem	56.14	40.46	59.80
Mem0	64.78	50.00	67.71
MemOCR	73.47	**60.11**	76.35
VCRM (Ours)	**74.70**	59.65	**77.33**

Table 3: Performance (LLM-as-a-Judge score, %) comparison under constrained budgets on LoCoMo and LongMemEval-S. Best in **bold**, second-best underlined.

	LoCoMo			LongMemEval-S		
Method	1024	256	64	1024	256	64
Qwen3-VL-4B-Instruct						
NaiveRAG	30.06	15.84	6.10	23.80	13.00	12.40
Mem0	42.47	28.96	16.49	48.10	41.90	27.62
LightMem	66.17	54.87	36.36	65.20	48.60	34.80
True Memory	73.51	60.71	29.29	64.80	43.80	15.80
VCRM (Ours)	**82.86**	**82.01**	**77.73**	**75.80**	**72.20**	**70.40**
Qwen2.5-VL-7B-Instruct						
NaiveRAG	35.06	22.21	12.34	22.60	12.60	13.80
Mem0	44.81	32.08	25.00	44.83	35.47	26.11
MemOCR	45.65	44.81	43.05	18.80	18.40	16.60
LightMem	63.57	55.52	41.75	47.80	46.40	31.20
True Memory	70.18	60.65	27.34	**56.80**	40.40	16.00
VCRM (Ours)	**75.39**	**75.00**	**73.96**	56.40	**52.80**	**51.00**

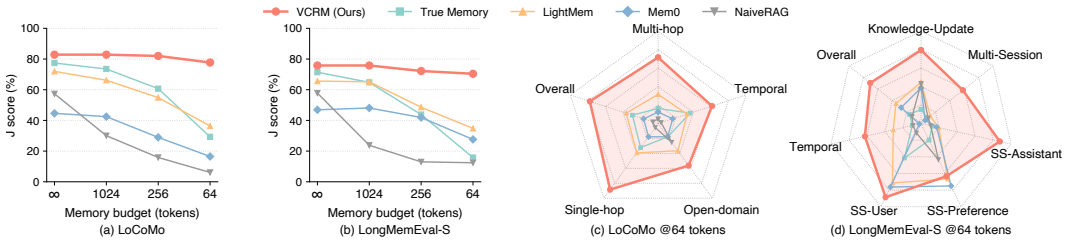


Figure 3: Performance analysis on LoCoMo and LongMemEval-S under Qwen3-VL-4B. (a) and (b) show J-score trends across memory budgets from unlimited to 64 tokens. (c) and (d) present per-category J-score comparisons under the 64-token budget.

To further evaluate the compression robustness of memory representations, we examine VCRM under explicit constrained memory budgets. In contrast to the unlimited budget comparison, this experiment progressively restricts the input memory tokens at the answer stage, simulating different LLM context-window constraints. We conduct controlled experiments under three settings of 1024, 256, and 64 tokens, where every method must satisfy the constraint that input tokens do not exceed the specified budget. As shown in Table 3, as the memory budget is compressed from 1024 to 64 tokens, VCRM maintains relatively stable performance: with Qwen3-VL-4B, the J-score on LoCoMo drops from 82.86 to 77.73, and from 75.80 to 70.40 on LongMemEval-S, with the degradation remaining within 6 percentage points in both cases; a similar trend is observed with Qwen2.5-VL-7B. In contrast, text-based baselines degrade substantially as the budget tightens.

This discrepancy stems from the different behaviors of the two memory representations under constrained windows: textual memory typically retains the highest-ranked evidence, while the diffusion of evidence causes information to be lost; VCRM, in contrast, significantly reduces memory capacity requirements through the complete visual memory pipeline and encodes facts via two-dimensional spatial layout, with critical evidence remaining identifiable at extreme compression thanks to visual salience. Therefore, VCRM not only achieves superior performance with memory overhead far below all baselines in the unconstrained setting, but also maintains near-full performance under memory-constrained scenarios. Fig. 3 (c) and (d) further present the per-category comparison under the 64-token budget. On LoCoMo, VCRM maintains a leading edge across all four reasoning categories, with the most pronounced gains in multi-hop and single-hop reasoning; on LongMemEval-S, VCRM outperforms all baselines on the vast majority of fine-grained memory dimensions.

4.3 Further Analysis

Ablation Study. We conduct ablation experiments on three benchmark datasets to validate the three key components of VCRM. Based on the results in Table 4, we draw the following conclusions: (1) The Hierarchical Evidence Prospecting (HEP) module is indispensable to overall performance. Re-

Table 4: Ablation study on the key components of our proposed VCRM method on the LoCoMo, LongMemEval-S, and HotpotQA datasets.

Component			LoCoMo			LongMemEval-S			HotpotQA		
HEP	SFE	DDR	J	F1	BLEU	J	F1	BLEU	F1	EM	SubEM
✓	–	–	69.09	38.58	31.98	45.00	32.16	28.04	73.15	57.00	74.64
–	✓	–	77.99	41.55	34.82	68.20	49.82	45.03	57.86	43.93	58.71
✓	–	✓	73.03	37.53	30.21	57.60	43.60	39.06	74.26	58.66	75.47
–	✓	✓	76.30	40.67	33.90	71.20	51.83	46.81	59.89	45.90	60.60
✓	✓	–	78.41	44.13	37.45	73.40	52.34	47.31	77.86	58.35	78.27
✓	✓	✓	**82.86**	**46.47**	**39.26**	**75.80**	**54.56**	**49.39**	**79.16**	**62.50**	**81.95**

moving HEP leads to significant degradation across all datasets, demonstrating that retrieval filtering is a critical step for downstream generation quality. (2) The Semantic Fact Extraction (SFE) module is equally crucial. The substantial performance drop when ablating SFE indicates that distilling structured facts from retrieved evidence effectively enhances the information density of memory. (3) The Density-Differentiated Rendering (DDR) module further improves performance. The variant without DDR corresponds to feeding the facts directly as plain text, and experiments show that adding the rendered visual record still yields clear improvements across all metrics, with relatively pronounced gains on numeric-sensitive questions, validating the effectiveness of visual salience encoding. Fig. 4 further presents a radar comparison of J-score across categories under w/o HEP, w/ HEP, and Oracle settings, where Oracle denotes the ideal retrieval upper bound by providing ground-truth sessions. VCRM consistently approaches the Oracle upper bound across all categories, confirming the effectiveness of the HEP module in precisely locating critical evidence.

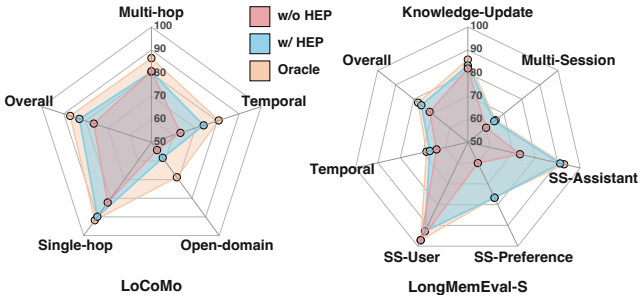


Figure 4: Oracle analysis of retrieval strategies on LoCoMo and LongMemEval-S. We compare VCRM with w/o HEP and gold-session Oracle.

Analysis on Tiered Confidence Routing. We evaluate routing efficacy on LoCoMo via Recall@k (average gold coverage), AllFound@k (complete coverage), and avgCount (retrieval session count). Fig. 5 (left) compares configurations. Single-tier with K=9 consumes 9.00 sessions for an AllFound of 0.900, while the optimal two-tier configuration of 12 and 6 reaches 0.904 with 8.09 sessions. In contrast, our three-tier routing achieves an AllFound of 0.923 with only 7.88 sessions, simultaneously improving coverage and compression. Notably, the three-tier variant without pruning attains comparable AllFound of 0.925 but requires 8.73 sessions, indicating that Adaptive Cliff Pruning removes redundant content while largely preserving recall. Fig. 5 (right) presents per-category statistics. Multi-hop and open-domain queries require more sessions at 11.55 and 14.76 respectively, yet achieve lower AllFound, reflecting greater retrieval difficulty with dispersed ground-truth evidence. Conversely, single-hop and temporal queries achieve near-perfect AllFound with fewer sessions. This validates the three-tier design: expanding retrieval for complex queries to cover dispersed evidence while compressing for simple ones.

Analysis on Chunk Size. To ensure that the retrieved content stays within the effective attention window of the underlying VLM during extraction, we partition the retrieved text into chunks and evaluate different chunk sizes on Qwen3-VL-4B to assess their impact on extraction quality. As shown in Fig. 6, the 4k chunk achieves the best J-score of 82.86 on LoCoMo, while the 6k chunk yields the best J-score of 75.80 on LongMemEval-S. The no-chunk setting, which extracts facts from the full retrieved evidence directly, performs worst with J-score dropping to 80.13 and 70.80 on the two datasets, respectively. Both smaller chunks of 2k and larger chunks of 8k are slightly below the optimal configuration, indicating that moderate chunking strikes a better balance between information processing efficiency and focused attention.

Qualitative Analysis. We further visualize a representative case from LoCoMo to qualitatively evaluate the effectiveness of our method. As illustrated in Fig. 7, the question "Where did Caroline move from 4 years ago?" is a multi-hop query that requires combining two pieces of information

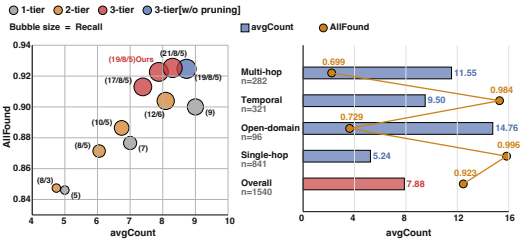


Figure 5: Analysis on tiered confidence routing on LoCoMo. Left: trade-off between AllFound and avgCount. Right: per-category performance with three-tier routing.

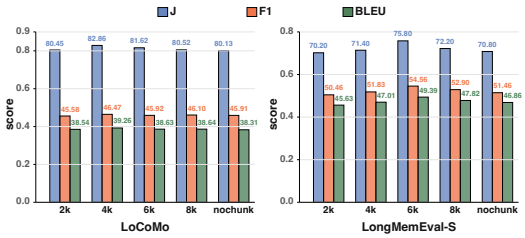


Figure 6: Analysis on chunk size configurations on LoCoMo and LongMemEval-S.

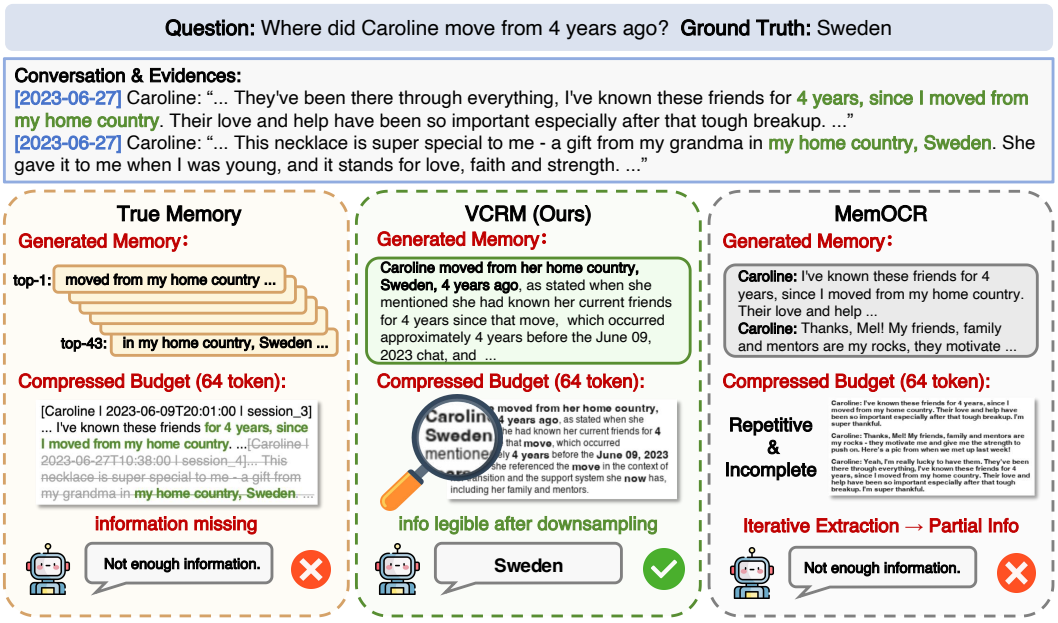


Figure 7: Qualitative analysis of the multi-hop reasoning case on the LoCoMo dataset.

across different sessions. Under the extreme 64-token budget, True Memory loses the specific home country information and outputs "Not enough information." MemOCR requires iterative fact extraction, with an average latency of about 9s, and the facts are repetitive and incomplete. In contrast, VCRM successfully retrieves both evidence pieces via HEP, extracts and integrates them via DSR, and performs visual salience encoding, with an average end-to-end latency of only 0.98s. After 64-token downsampling, the critical evidence remains clearly legible, yielding the correct answer "Sweden." Additional efficiency details are in the Appendix.

5 Conclusion

In this work, we proposed a novel Visual Context Record Memory (VCRM) framework. Specifically, we introduced Hierarchical Evidence Prospecting (HEP), which adaptively filters redundant information through tiered confidence routing to alleviate evidence diffusion at the source, and Discriminative Semantic Rendering (DSR), which transcribes critical evidence into two-dimensional visual records with non-uniform information density to break the linear token scaling constraint. Extensive experiments demonstrated that VCRM consistently outperformed five competitive baselines across LoCoMo, LongMemEval-S, and HotpotQA, while exhibiting marked robustness under extreme token compression budgets.

AI use statement

In this work, we used generative AI tools for aiding and polishing the writing of the manuscript and for information retrieval and discovery (e.g., locating relevant prior work and confirming submission

guidelines). We have not used generative AI tools for drafting sections of the paper or for research ideation, and the remaining tasks of research execution, synthetic dataset generation, or mathematical proof verification are not applicable to this work. We have reviewed all AI-assisted work, and the AI-assisted text and suggestions were checked and edited by the authors before inclusion. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

Our anonymized code is available at `https://anonymous.4open.science/r/VCRM-270A/`. The artifact includes all source code, prompts, the LoCoMo dataset, and the result files used to obtain the reported numbers. Large prebuilt artifacts omitted due to repository size limits are reproducible from the included scripts. Detailed dataset construction, baseline reproduction, and metric definitions are provided in the appendix.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Joshua Adler and Guy Zehavi. Storage is not memory: A retrieval-centered architecture for agent recall, 2026. URL `https://arxiv.org/abs/2605.04897`.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137, August 2024. URL `https://aclanthology.org/2024.acl-long.172/`.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, et al. Glyph: Scaling context windows via visual-text compression. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37145–37158, 2026.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025.

Zijie Dai, Shiyuan Deng, Sheng Guan, Yizhou Tian, Xin Yao, Xiao Yan, and James Cheng. RecMem: Recurrence-based memory consolidation for efficient and effective long-running LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32353–32376, July 2026. URL `https://aclanthology.org/2026.findings-acl.1619/`.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=dyJ0GWpjJB`.

Lang Feng, Fuchao Yang, Feng Chen, Xin Cheng, Haiyang Xu, Zhenglin Wan, Ming Yan, and Bo An. AgentOCR: Reimagining agent history via optical self-compression. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2026.

Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, and Boris Ginsburg. RULER: What's the real context size of your long-context language models? In *First Conference on Language Modeling*, 2024. URL `https://openreview.net/forum?id=kIoBbc76Sy`.

Chuanrui Hu, Xingze Gao, Zuyi Zhou, Dannong Xu, Yi Bai, Xintong Li, Hui Zhang, Tong Li, Chong Zhang, Lidong Bing, and Yafeng Deng. EverMemOS: A self-organizing memory operating system for structured long-horizon reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2026.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, et al. Memory in the age of ai agents. *arXiv preprint arXiv:2512.13564*, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Haodong Lei, Junming Liu, Yirong Chen, Ding Wang, and Hongsong Wang. Memcot: Test-time scaling through memory-driven chain-of-thought, 2026. URL https://arxiv.org/abs/2604.08216.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, August 2024.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. Zep: A temporal knowledge graph architecture for agent memory, 2025. URL https://arxiv.org/abs/2501.13956.

Zhanyu Shen, Sijie Cheng, Zhicheng Guo, Weiqin Wang, Yile Wang, and Hui Huang. AnchorMem: Anchored facts with associative contexts for building memory in large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 34784–34798, July 2026. doi: 10.18653/v1/2026.findings-acl.1736. URL https://aclanthology.org/2026.findings-acl.1736/.

Yaorui Shi, Shugui Liu, Yu Yang, Wenyu Mao, Yuxin Chen, Qi GU, Hui Su, Xunliang Cai, Xiang Wang, and An Zhang. MemOCR: Layout-aware visual memory for efficient long-horizon reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=vv4TMkyfWA.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J Ross Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=y59hf5lrMn.

Kimi Team, Angang Du, Bohong Yin, Bowei Xing, Bowen Qu, Bowen Wang, Cheng Chen, Chenlin Zhang, Chenzhuang Du, Chu Wei, et al. Kimi-vl technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Xinyu Wang, Mingze Li, Peng Lu, Xiao-Wen Chang, Lifeng Shang, Jinpeng Li, Fei Mi, Prasanna Parthasarathi, and Yufei Cui. Infmem: Learning system-2 memory control for long-context agent. In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems*, 2026a. URL https://openreview.net/forum?id=zJirFEiqem.

Yibo Wang, Yongcheng Jing, Shunyu Liu, Hao Guan, Rong-cheng Tu, Chengyu Wang, Jun Huang, and Dacheng Tao. Vtc-r1: Vision-text compression for efficient long-context reasoning. *arXiv preprint arXiv:2601.22069*, 2026b.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-$alpha$: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*, 2025.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*, 2025.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr 2: Visual causal flow. *arXiv preprint arXiv:2601.20552*, 2026.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2024.

Menglin Xia, Xuchao Zhang, Shantanu Dixit, Paramaguru Harimurugan, Rujia Wang, Victor Rühle, Robert Sim, Chetan Bansal, and Saravan Rajmohan. Memora: A harmonic memory representation balancing abstraction and specificity. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=zSrvkj0ers.

Roy Xie, Dan Friedman, Donghan Yu, Bowen Pan, Christopher Fifty, Jang-Hyun Kim, Xianzhi Du, Zhe Gan, Vivek Rathod, and Bhuwan Dhingra. Lensvlm: Selective context expansion for compressed visual representation of text. *arXiv preprint arXiv:2605.07019*, 2026.

Ling Xing, Alex Jinpeng Wang, Rui Yan, Xiangbo Shu, and Jinhui Tang. Vision-centric token compression in large language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=YdggdEL41C.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=FiM0M8gcct.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, July 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, October-November 2018.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context LLM with multi-conv RL-based memory agent. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=k5nIOvYGCL.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. Qwen3 embedding: Advancing text embedding and reranking through foundation models, 2025a. URL https://arxiv.org/abs/2506.05176.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, 43(6):1–47, 2025b.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. Memento: Fine-tuning llm agents without fine-tuning llms, 2025. URL https://arxiv.org/abs/2508.16153.

A Experiment Details

A.1 Datasets and Baselines

Datasets. The LoCoMo benchmark Maharana et al. (2024) is designed to evaluate the very long-term conversational memory of LLM agents. It consists of 10 complete conversations with 1,540 questions in total (approximately 150 questions per conversation), where each conversation covers all four reasoning categories: Single-hop (841), Multi-hop (282), Temporal (321), and Open-domain (96). The average full conversation length is approximately 18K tokens. Each conversation spans 19–32 sessions, with an average session length of approximately 665 tokens.

The LongMemEval dataset Wu et al. (2024) is designed to benchmark long-term interactive memory in conversational agents. It provides two versions with different lengths: LongMemEval-S with approximately 115K tokens per problem, and LongMemEval-M extending up to 1.5 million tokens across 500 sessions. We adopt the LongMemEval-S version in this work. It comprises 500 independent conversations, covering six question types: single-session-user, single-session-assistant, single-session-preference, temporal-reasoning, knowledge-update, and multi-session, among which 30 are abstention variants (question_id ending with `_abs`). The average full conversation length is approximately 105K tokens. Each conversation contains 48 sessions on average, with an average session length of approximately 2,200 tokens.

The HotpotQA benchmark Yang et al. (2018) is a widely adopted dataset for multi-hop question answering, containing 113K Wikipedia-based question-answer pairs where each question requires reasoning over multiple supporting documents. Since the original distractor setting only provides 10 paragraphs per question, which is insufficient for evaluating long-context memory, we extend the context to 30K tokens by greedily padding with additional renderable distractor documents sampled from a large-scale document pool, while always preserving the original gold paragraphs. This yields 1,928 hard-level evaluation questions with substantially extended contexts, transforming the original multi-hop QA task into a long-context memory reasoning challenge.

Baselines. We compare our approach against several representative baselines of conversational memory modeling. ① NaiveRAG: A retrieval-augmented baseline that retrieves relevant historical context via embedding similarity and concatenates it as input for the VLM. ② Mem0 Chhikara et al. (2025): Extracts memories from dialogue turns through a combination of global summaries and recent context, maintaining them via LLM-guided operations. ③ LightMem Fang et al. (2026): Organizes conversational memory in a three-stage hierarchy inspired by the Atkinson–Shiffrin model, with sensory memory for rapid filtering, topic-aware short-term memory for consolidation, and long-term memory with offline sleep-time updates. ④ True Memory Adler & Zehavi (2026): A six-layer architecture that shifts the focus from storage schema design to a multi-stage retrieval pipeline, prioritizing retrieval effectiveness over storage organization. ⑤ MemOCR Shi et al. (2026): A multimodal memory agent that employs layout-aware visual memory encoding, adaptively compressing interaction histories into visual layouts with non-uniform information density.

Reproduction of baselines. To ensure fair and reliable comparisons across all baselines, we configure each baseline to faithfully reflect its original design choices while unifying the evaluation framework. All baselines share the same LLM-as-a-Judge evaluation protocol following True Memory Adler & Zehavi (2026) to ensure that performance differences originate solely from the memory mechanisms rather than evaluation criteria. All baselines use their official embedding models and default configurations; we only replace the backbone LLM with our evaluated models for consistency. Specifically, for NaiveRAG, we follow RecMem Dai et al. (2026) with a 2048-token chunking strategy that preserves message integrity: messages are accumulated sequentially until the chunk reaches the budget, and if adding the next message would exceed the limit, the entire message is still included rather than truncated. Retrieval is performed via embedding similarity search over chunks. For LightMem, we use its official codebase and default configuration. For Mem0, we adopt the reproduction scripts provided by the LightMem repository Fang et al. (2026). For True Memory, we use its official implementation with the Base tier configuration. For MemOCR, we use its official codebase with the released 7B checkpoint.

A.2 METRICS

LLM-as-a-Judge score. For LoCoMo and LongMemEval-S, we adopt the LLM-as-a-Judge paradigm and use Qwen3-235B-A22B-Instruct-2507 as the judge. For each (question, prediction) pair the judge issues a binary verdict, *Correct* or *Wrong*. To reduce the stochasticity of a single judgment, every question is judged three times independently and the final verdict is determined by majority voting. The J-score is the fraction of questions whose majority verdict is *Correct*. Unlike surface-form matching, the J-score measures the semantic correctness of a response—specifically its comprehensiveness and logical coherence—and is therefore tolerant to paraphrasing.

BLEU. BLEU-1 measures the unigram lexical overlap between the prediction and gold answers. The clipped unigram precision is defined as

$$p_1 = \frac{\sum_{g \in \hat{y}} \min\left(\mathrm{Count}(\hat{y}, g),\ \max_{y \in \mathcal{A}} \mathrm{Count}(y, g)\right)}{\sum_{g \in \hat{y}} \mathrm{Count}(\hat{y}, g)}, \tag{7}$$

where clipping limits each unigram count to its maximum frequency in any gold answer. To penalize predictions shorter than the reference, the final score is $\mathrm{BLEU} = \mathrm{BP} \cdot p_1$, with the brevity penalty $\mathrm{BP} = e^{1-r/|\hat{y}|}$ when $|\hat{y}| \le r$ and 1 otherwise, where r denotes the reference length.

Exact Match (EM). For HotpotQA, whose answers are short factual spans, we adopt the standard HotpotQA string-level metrics. Let $\hat{y}$ be the prediction and $\mathcal{A}$ the set of gold answers. EM is 1 if the normalized prediction equals some normalized gold answer and 0 otherwise:

$$\mathrm{EM} = \mathbf{1}[\exists\, y \in \mathcal{A} :\ \hat{y} = y]\,. \tag{8}$$

Sub-Exact Match (Sub-EM). Sub-EM relaxes EM into a substring match: it is 1 if some normalized gold answer occurs as a substring of the normalized prediction, or vice versa, and 0 otherwise:

$$\text{Sub-EM} = \mathbf{1}[\exists\, y \in \mathcal{A} :\ y \sqsubseteq \hat{y} \ \vee\ \hat{y} \sqsubseteq y]\,, \tag{9}$$

where $\sqsubseteq$ denotes the substring relation.

F1 Score. F1 is the token-level F1 between the normalized prediction and a gold answer. With $T(\cdot)$ the whitespace tokenization, $c = |T(\hat{y}) \cap T(y)|$ the token intersection size, $P = c/|T(\hat{y})|$, $R = c/|T(y)|$, and $\mathrm{F1}_y = 2PR/(P+R)$, the per-question score takes the maximum over gold answers:

$$\mathrm{F1} = \max_{y \in \mathcal{A}} \mathrm{F1}_y. \tag{10}$$

A.3 IMPLEMENTATION DETAILS

All experiments are conducted on hardware equipped with 8 PPU-ZW810E GPUs, dual Intel Xeon Platinum 8575C CPUs with 96 cores and 192 threads, and 768 GB of RAM. The VLMs Qwen3-VL-4B-Instruct and Qwen2.5-VL-7B-Instruct are served locally via vLLM Kwon et al. (2023) in bfloat16 with greedy decoding at temperature 0 and seed 42. The embedding model Qwen3-Embedding-0.6B and the reranker bge-reranker-v2-m3 are downloaded and run locally, while the LLM judge Qwen3-235B-A22B-Instruct-2507 is accessed via API. The distilled fact set is rendered into a single visual memory image via Pillow and downsampled proportionally for budget experiments.

The reranker accepts at most 8,000 characters per document. Since LongMemEval-S sessions can be far longer up to approximately 18.9K tokens, each session is split into chunks of at most 8,000 characters at line or message boundaries, reranked in batches of six, and the session's final rerank score is the maximum over its chunks. The shorter LoCoMo and HotpotQA sessions are reranked directly without splitting.

B EMPIRICAL RESULTS

B.1 PERFORMANCE ANALYSIS

Per-category Analysis. Tables 5 and 6 present the per-category J-scores of all methods under different memory budgets on LoCoMo and LongMemEval-S, respectively. On LoCoMo, VCRM

achieves the best or second-best performance across nearly all categories and budget settings. Under the unlimited budget, VCRM substantially leads in multi-hop, single-hop, and overall scores, while trailing slightly behind True Memory on temporal reasoning and tying with NaiveRAG on open-domain questions. When the budget is compressed to 64 tokens, VCRM's advantage becomes more pronounced: its overall J-score drops only marginally, whereas other baselines such as True Memory degrade severely, and NaiveRAG and Mem0 nearly lose their reasoning capability entirely. The same trend holds under the 7B backbone, where VCRM maintains strong performance under extreme compression while all text-based baselines decline substantially. Results on LongMemEval-S further validate this pattern. VCRM consistently leads in overall J-score across all budget settings, and under the extreme 64-token budget it still significantly outperforms the strongest baseline. At the fine-grained level, VCRM demonstrates particularly notable advantages on categories requiring precise factual retrieval and numeric sensitivity, such as Single-Session User, Temporal, and Knowledge-Update, benefiting from the accurate filtering of the HEP module and the visual saliency encoding of the DSR module. In contrast, text-based baselines suffer from severe performance degradation under long-context scenarios due to dispersed key evidence, while MemOCR performs poorly across multiple categories due to its weak extraction capability and limited ability to aggregate evidence across long conversational contexts.

Table 5: Per-category J-score (%) on LoCoMo across memory budgets (∞/1024/256/64 tokens). Best in **bold**, second-best underlined. MemOCR is released only under the 7B backbone, so its 4B cells are marked "–".

		Qwen3-VL-4B-Instruct				**Qwen2.5-VL-7B-Instruct**			
Category	Method	∞	1024	256	64	∞	1024	256	64
Multi-hop (282)	NaiveRAG	67.73	37.94	20.57	4.96	63.12	39.01	25.53	10.99
	Mem0	49.65	46.10	29.79	12.41	51.77	46.10	26.95	20.21
	LightMem	71.63	68.79	54.26	31.56	66.31	63.12	53.90	39.01
	MemOCR	–	–	–	–	51.77	51.77	51.06	48.58
	True Memory	68.79	69.50	48.94	16.31	67.73	67.38	51.06	13.12
	VCRM (Ours)	**80.50**	**80.50**	**76.95**	**71.63**	**69.15**	**69.15**	**69.15**	**67.38**
Temporal (321)	NaiveRAG	34.58	17.45	8.72	3.43	41.12	19.63	11.21	4.05
	Mem0	28.66	28.35	21.18	16.51	39.25	35.51	28.35	23.99
	LightMem	52.65	46.11	43.30	33.02	45.48	46.11	42.37	36.14
	MemOCR	–	–	–	–	27.73	27.73	27.10	27.10
	True Memory	**76.01**	73.21	66.04	36.45	**66.04**	**66.04**	57.01	34.89
	VCRM (Ours)	73.83	**73.83**	**73.83**	**61.37**	61.68	61.68	**61.68**	**61.06**
Open-domain (96)	NaiveRAG	**58.33**	48.96	38.54	25.00	54.17	45.83	34.38	28.13
	Mem0	35.42	31.25	29.17	17.71	33.33	35.42	31.25	28.12
	LightMem	52.08	50.00	46.88	36.46	46.88	48.96	42.71	38.54
	MemOCR	–	–	–	–	27.08	27.08	28.12	27.08
	True Memory	51.04	51.04	35.42	17.71	53.13	51.04	30.21	11.46
	VCRM (Ours)	**58.33**	**58.33**	**57.29**	**56.25**	**58.33**	**58.33**	**56.25**	**54.17**
Single-hop (841)	NaiveRAG	62.43	30.08	14.39	5.35	67.54	38.41	23.90	14.15
	Mem0	50.06	47.92	31.63	17.72	58.26	48.99	35.32	26.63
	LightMem	81.69	74.79	60.40	39.24	74.20	72.06	62.54	45.18
	MemOCR	–	–	–	–	52.56	52.56	51.37	49.11
	True Memory	83.83	77.53	65.52	32.22	77.17	74.88	68.73	31.03
	VCRM (Ours)	**89.89**	**89.89**	**89.66**	**88.47**	**84.66**	**84.66**	**84.19**	**83.35**
Overall (1540)	NaiveRAG	57.34	30.06	15.84	6.10	60.39	35.06	22.21	12.34
	Mem0	44.61	42.47	28.96	16.49	51.56	44.81	32.08	25.00
	LightMem	71.95	66.17	54.87	36.36	65.06	63.57	55.52	41.75
	MemOCR	–	–	–	–	45.65	45.65	44.81	43.05
	True Memory	77.40	73.51	60.71	29.29	71.62	70.18	60.65	27.34
	VCRM (Ours)	**82.86**	**82.86**	**82.01**	**77.73**	**75.39**	**75.39**	**75.00**	**73.96**

Discussion on F1 and BLEU Score. F1 and BLEU scores are token-level exact-matching metrics widely adopted in prior work Xu et al. (2025); Shen et al. (2026); Lei et al. (2026); Chhikara et al. (2025), yet they exhibit clear limitations in long-context memory evaluation. These metrics penalize semantically correct answers that differ in phrasing from the gold answer. For instance, when the gold answer is "16 March, 2023" and the model generates "Gina opened her online clothing store

Table 6: Per-category J-score (%) on LongMemEval-S across memory budgets (∞/1024/256/64 tokens). Best in **bold**, second-best underlined. MemOCR is released only under the 7B backbone, so its 4B cells are marked "–".

		Qwen3-VL-4B-Instruct				Qwen2.5-VL-7B-Instruct			
Category	Method	∞	1024	256	64	∞	1024	256	64
Knowledge-Update (78)	NaiveRAG	65.38	24.36	14.10	12.82	41.03	21.79	15.38	14.10
	Mem0	51.61	48.39	45.16	38.71	38.71	35.48	35.48	32.26
	LightMem	74.36	75.64	65.38	44.87	61.54	61.54	64.10	46.15
	MemOCR	–	–	–	–	15.38	15.38	15.38	15.38
	True Memory	**83.33**	**87.18**	62.82	15.38	47.44	73.08	53.85	15.38
	VCRM (Ours)	**83.33**	83.33	**79.49**	**79.49**	**78.21**	**78.21**	**69.23**	**70.51**
Multi-Session (133)	NaiveRAG	41.35	10.53	9.77	9.77	21.05	10.53	8.27	9.02
	Mem0	45.28	43.40	28.30	5.66	**41.51**	37.74	26.42	5.66
	LightMem	**64.66**	63.16	30.08	12.78	39.10	**39.10**	**35.34**	9.02
	MemOCR	–	–	–	–	9.02	9.02	9.02	7.52
	True Memory	57.14	40.60	24.06	9.02	31.58	36.84	24.81	9.02
	VCRM (Ours)	**64.66**	**64.66**	**57.14**	**57.89**	33.08	33.08	31.58	**33.08**
Single-Session Assistant (56)	NaiveRAG	91.07	53.57	14.29	10.71	**76.79**	55.36	16.07	17.86
	Mem0	21.74	17.39	17.39	17.39	39.13	30.43	21.74	26.09
	LightMem	32.14	30.36	21.43	19.64	23.21	23.21	21.43	23.21
	MemOCR	–	–	–	–	57.14	57.14	55.36	44.64
	True Memory	**96.43**	**94.64**	67.86	12.50	**76.79**	**96.43**	**69.64**	16.07
	VCRM (Ours)	91.07	91.07	**89.29**	**87.50**	67.86	67.86	67.86	**57.14**
Single-Session Preference (30)	NaiveRAG	56.67	60.00	40.00	43.33	33.33	36.67	30.00	43.33
	Mem0	**86.67**	**91.67**	75.00	**75.00**	75.00	**83.33**	**75.00**	**66.67**
	LightMem	83.33	90.00	73.33	66.67	**80.00**	76.67	66.67	53.33
	MemOCR	–	–	–	–	23.33	23.33	23.33	20.00
	True Memory	56.67	56.67	40.00	20.00	30.00	46.67	43.33	23.33
	VCRM (Ours)	76.67	76.67	**86.67**	63.33	50.00	50.00	33.33	30.00
Single-Session User (70)	NaiveRAG	74.29	32.86	15.71	11.43	55.71	34.29	14.29	12.86
	Mem0	86.84	89.47	89.47	76.32	77.42	77.42	61.29	58.06
	LightMem	**92.86**	88.57	85.71	71.43	67.14	67.14	65.71	60.00
	MemOCR	–	–	–	–	20.00	20.00	20.00	20.00
	True Memory	91.43	**95.71**	78.57	41.43	64.29	**91.43**	75.71	38.57
	VCRM (Ours)	**92.86**	92.86	**92.86**	**88.57**	**90.00**	90.00	**87.14**	**88.57**
Temporal (133)	NaiveRAG	47.37	11.28	7.52	9.02	30.08	12.03	9.02	10.53
	Mem0	26.42	26.42	22.64	1.89	32.08	35.85	26.42	15.09
	LightMem	57.14	57.89	43.61	30.83	42.11	42.11	42.86	27.82
	MemOCR	–	–	–	–	12.78	12.78	12.03	12.03
	True Memory	60.90	48.87	24.81	9.77	26.32	34.59	24.06	9.77
	VCRM (Ours)	**66.92**	**66.92**	**61.65**	**62.41**	**45.86**	**45.86**	**44.36**	**39.85**
Overall (500)	NaiveRAG	57.80	23.80	13.00	12.40	38.40	22.60	12.60	13.80
	Mem0	46.92	48.10	41.90	27.62	45.81	44.83	35.47	26.11
	LightMem	65.60	65.20	48.60	34.80	48.00	47.80	46.40	31.20
	MemOCR	–	–	–	–	18.80	18.80	18.40	16.60
	True Memory	71.40	64.80	43.80	15.80	42.20	**56.80**	40.40	16.00
	VCRM (Ours)	**75.80**	**75.80**	**72.20**	**70.40**	**56.40**	56.40	**52.80**	**51.00**

on 2023-03-16", both F1 and BLEU-1 approach zero despite the answer being entirely factually correct. Consequently, we prioritize LLM-as-a-Judge for semantic evaluation in our main analysis.

Table 7 presents the overall F1 and BLEU-1 scores of each method under the Qwen3-VL-4B backbone. VCRM achieves the highest scores across all budget settings while maintaining remarkable stability: on LoCoMo, its F1 ranges from 44.4% to 46.6%; on LongMemEval-S, from 50.9% to 54.6%. In contrast, the strongest baseline LightMem reaches 42.1% under the unlimited budget on LoCoMo but drops to 20.0% as the budget shrinks, a decline of more than half. On LongMemEval-S, True Memory scores 46.6% under the unlimited budget but plummets to 6.0% at 64 tokens. These results further validate VCRM's advantage in preserving critical factual information within long contexts.

Table 7: Overall F1 and BLEU-1 (%) of each memory baseline under Qwen3-VL-4B across memory budgets (∞/1024/256/64 tokens). Upper block: LoCoMo; lower block: LongMemEval-S.

	∞		1024		256		64		Avg	
Method	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU	F1	BLEU
LoCoMo										
NaiveRAG	31.19	25.76	17.20	13.99	9.72	7.91	4.69	3.67	15.70	12.83
Mem0	21.13	17.42	19.22	15.83	13.78	11.31	8.00	6.45	15.53	12.75
LightMem	42.07	35.62	38.90	32.63	31.87	26.68	19.97	16.41	33.20	27.83
True Memory	27.90	19.62	24.53	16.62	20.39	13.72	9.61	6.43	20.61	14.10
VCRM (Ours)	**46.47**	**39.26**	**46.47**	**39.26**	**46.58**	**39.49**	**44.44**	**37.47**	**45.99**	**38.87**
LongMemEval-S										
NaiveRAG	39.41	34.44	16.33	13.73	8.60	6.10	7.24	4.95	17.89	14.80
Mem0	7.64	3.49	7.98	3.75	9.58	4.95	9.49	5.17	8.67	4.34
LightMem	7.80	3.41	7.66	3.32	9.77	6.08	9.52	5.67	8.69	4.62
True Memory	46.63	40.35	25.76	18.40	16.47	10.92	5.99	3.56	23.71	18.31
VCRM (Ours)	**54.56**	**49.39**	**54.56**	**49.39**	**50.86**	**45.94**	**50.92**	**45.88**	**52.73**	**47.65**

B.2 Cost-Efficiency Analysis

During the answer stage, VCRM feeds only the rendered visual memory image together with the query into the VLM, without re-reading the original long-text history. The image is rendered from HEP-filtered facts via DSR and downsampled to keep its visual token count within the budget upper bound, a cost that is independent of the original context length. As shown in Table 8, regardless of whether the history contains approximately 19K tokens as in LoCoMo or 106K tokens as in LongMemEval-S, VCRM consumes only 334 and 420 tokens with the 4B backbone, and 289 and 363 tokens with the 7B backbone. In contrast, text-based baselines incur memory that grows with history length: the full context requires 19,092 and 106,278 tokens, and even the most compact text-based baseline still consumes 2,723 tokens on LoCoMo, several times more than VCRM. VCRM replaces linearly growing textual memory with budget-bounded visual memory, fundamentally avoiding the linear scaling of token costs.

Table 8: Average input memory token consumption. Best results are highlighted in **bold**, and second-best results are underlined.

Method	**LoCoMo**	**LongMemEval-S**	**Avg**
Qwen3-VL-4B-Instruct			
Full Context	19,092	106,278	62,685
NaiveRAG	5,983	6,019	6,001
True Memory	6,370	12,894	9,632
LightMem	3,363	1,136	2,250
Mem0	2,723	2,182	2,453
VCRM (Ours)	**334**	**420**	**377**
Qwen2.5-VL-7B-Instruct			
Full Context	19,092	106,278	62,685
NaiveRAG	6,147	6,038	6,093
True Memory	5,773	11,748	8,761
LightMem	2,250	1,059	1,655
Mem0	2,649	2,094	2,372
MemOCR	972	989	981
VCRM (Ours)	**289**	**363**	**326**

In terms of runtime, Table 9 compares the average per-instance latency of VCRM and the visual baseline MemOCR across datasets, where construction time denotes the total latency of building the memory from the raw context, covering the full pipeline in both paradigms—including retrieval, extraction, and all other sub-processes. MemOCR lacks a retrieval module and employs a fixed 5000-token chunking strategy, requiring a VLM invocation for key fact extraction on each chunk. Consequently, its memory construction time grows linearly with context length: on LoCoMo, it processes approximately 4 chunks, taking 6.106 seconds; on LongMemEval-S, it processes approximately 20 chunks, taking 35.804 seconds. In contrast, VCRM, through the tiered filtering of the HEP module, performs fact extraction only on a small set of high-confidence candidate segments, thereby substantially reducing the number of VLM invocations. On LoCoMo, VCRM's total per-

instance latency is only 0.987 seconds, nearly 9 times faster than MemOCR's 9.323 seconds. On LongMemEval-S, VCRM requires only 2.478 seconds, 15 times faster than MemOCR's 37.201 seconds. This clearly highlights the core role of the HEP module in reducing computational overhead: replacing exhaustive chunk-by-chunk extraction with precise retrieval filtering, fundamentally avoiding linearly growing inference costs.

Table 9: Average runtime (seconds/sample) of MemOCR and VCRM across different datasets and memory budgets.

Dataset	Method	Construction (s)	Render (s)	Answer (s)	Total (s)
LoCoMo	MemOCR	6.106	0.180	3.036	9.323
	VCRM (Ours)	0.654	0.032	0.300	**0.987**
LongMemEval-S	MemOCR	35.804	0.189	1.209	37.201
	VCRM (Ours)	1.975	0.083	0.420	**2.478**
HotpotQA	MemOCR	6.800	0.035	0.405	7.241
	VCRM (Ours)	0.183	0.105	0.298	**0.585**

C Case Analysis

We present two qualitative case studies from LongMemEval-S Wu et al. (2024) to illustrate VCRM's reasoning process. The first case (Figure 8) requires temporal reasoning across sessions to identify the most recently used mode of transport. The second case (Figure 9) involves multi-session aggregation to count the total number of owned musical instruments. Both figures show that VCRM accurately retrieves and integrates dispersed evidence through HEP, renders it into visually salient memory via DSR, and produces correct answers even under severe compression.

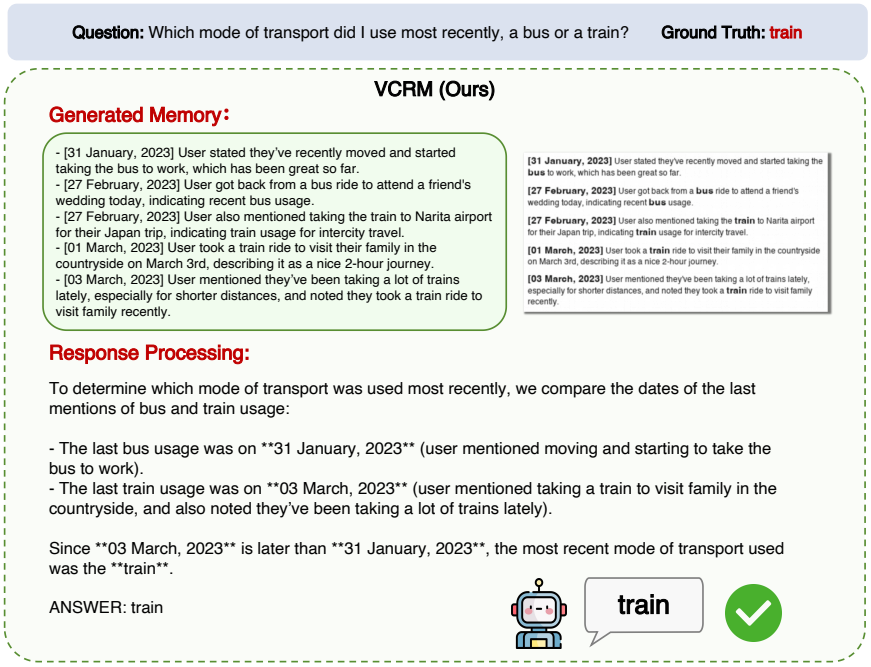


Figure 8: A case study from LongMemEval-S on the temporal-reasoning task.

D Prompts

We provide the prompt templates used in our experiments. The LLM-as-a-Judge evaluation protocol follows True Memory Adler & Zehavi (2026), with separate judge prompts for LoCoMo and LongMemEval-S. The answer generation prompts are dataset-specific for LoCoMo, LongMemEval-S, and HotpotQA. The fact extraction prompt is used by VCRM's DSR module to distill structured facts from retrieved evidence.

Question: How many musical instruments do I currently own?
Ground Truth: I currently own **4 musical instruments**. I've had the Fender Stratocaster electric guitar for 5 years, the Yamaha FG800 acoustic guitar for 8 years, the 5-piece Pearl Export drum set for an unspecified amount of time, and the Korg B1 piano for 3 years.

VCRM (Ours)

Generated Memory:

- [2023/05/22 (Mon) 04:35] User owns a Korg B1 piano, which they've, ···
- [2023/05/22 (Mon) 04:35] User owns a 5-piece Pearl Export drum set, ···
- [2023/05/22 (Mon) 21:42] User owns a Yamaha FG800 acoustic guitar, ···
- [2023/05/20 (Sat) 17:20] User owns a black Fender Stratocaster electric guitar, ···
- [2023/05/22 (Mon) 21:42] User recently bought a new overdrive pedal from Guitar ···
- [2023/05/22 (Mon) 04:35] User is considering buying a new Cordoba ukulele, ···



Response Processing:

Based on the image memories:
- User owns a Korg B1 piano (≈3 years old) — confirmed as owned.
- User owns a 5-piece Pearl Export drum set — confirmed as owned.
- User owns a Yamaha F680 acoustic guitar (≈8 years old) — confirmed as owned.
- User owns a black Fender Stratocaster electric guitar (≈5 years old) — confirmed as owned.
- User recently bought a new overdrive pedal — this is a pedal, not an instrument.
- User is considering buying a new Cordoba ukulele — this is not yet owned.

So, the instruments currently owned are:
1. Korg B1 piano
2. 5-piece Pearl Export drum set
3. Yamaha F680 acoustic guitar
4. Black Fender Stratocaster electric guitar
That's 4 instruments.

ANSWER: 4



Figure 9: A case study from LongMemEval-S on the multi-session task.

Prompt Template for LLM as a Judge (LoCoMo)

You are a strict answer grader. Output ONLY valid JSON.
Determine if the generated answer is CORRECT or WRONG compared to the gold answer.
Be generous: if the generated answer mentions the same core topic/fact, mark CORRECT.
For time questions: same date/period in any format counts as CORRECT.

Question: {question}
Gold answer: {gold}
Generated answer: {generated}

Output ONLY: {{"label": "CORRECT"}} or {{"label": "WRONG"}}"""

Figure 10: LLM-as-a-Judge prompt for LoCoMo.

Prompt Template for LLM as a Judge (LongMemEval-S)

```
You are a strict answer grader. Output ONLY valid JSON.
Determine if the generated answer is CORRECT or WRONG compared to the gold answer.
Be generous: if the generated answer mentions the same core topic/fact, mark CORRECT.
For time questions: same date/period in any format counts as CORRECT.
For preferences: if the core preference is captured, mark CORRECT.
For knowledge updates: the response should reflect the LATEST information.

Question: {question}
Gold answer: {gold}
Generated answer: {generated}

Output ONLY: {{"label": "CORRECT"}} or {{"label": "WRONG"}}
```

Figure 11: LLM-as-a-Judge prompt for LongMemEval-S.

Prompt Template for Results Generation (LongMemEval-S)

```
You are a personal assistant with access to extracted memories from past conversations with a user. The
memories are shown in the IMAGE above — read them carefully before answering the question below.

Today's Date: {question_date}

{question_date} = today (question-asking date). [DD Month, YYYY] before each memory = the date that
chat session happened. They are different dates.
For any question involving time or quantity calculations, perform the computation first, then give the final
answer.
For conflicting values of the same fact, use the most recent memory.
If the question leans toward retrieving assistant-provided information, such as "Can you remind me...",
prioritize facts where the assistant recommended or suggested something.

Question: {question}

Give your final answer after "ANSWER:".
```

Figure 12: Answer prompt for LongMemEval-S.

Prompt Template for Results Generation (LoCoMo)

```
You are a personal assistant with access to extracted memories from past conversations with a user. The
memories are shown in the IMAGE above — read them carefully before answering the question below.

{memories}
Question: {question}
Give your final answer after "ANSWER:".
```

Figure 13: Answer prompt for LoCoMo.

Prompt Template for Results Generation (HotpotQA)

```
Based only on the document in the image, answer the question with a short phrase (a name, entity, number, or
yes/no). Do not explain.
<problem>
{question}
</problem>

Your answer:
```

Figure 14: Answer prompt for HotpotQA.

Prompt Template for Extracting Facts (HotpotQA)

You are presented with a problem and {n_docs} article sections that may contain the answer to the problem. Please read the articles carefully and extract the crucial information that helps to answer the problem.

<problem>
{question}
</problem>

<articles>
{articles}
</articles>

You are given a problem and several article sections.
You should draft the extracted key information in markdown format with the crucial information in it that helps to answer the problem.
In your markdown draft, you may use different headings to arrange the font sizes and styles of the information.
E.g., more important information should be emphasized and more visible (larger font size, bolder, colored red, etc.), in case the rendered image can be clearly read.
No need to wrap with ```markdown or ```, just write the markdown draft directly.

EXTRACTION RULES
- Extract ONLY information explicitly stated in the articles — never infer, guess, or add information that is not written.
- Be GENEROUS with relevance: if a statement shares even one keyword with the problem, keep it. Skip only completely unrelated background.
- Preserve specific details EXACTLY: proper nouns, titles, numbers, dates, quantities — never generalize ("416 pages" not "about 400", "assistant manager" not "manager").
- Each piece of information should be self-contained — replace pronouns with the specific reference.
- For comparison/multi-hop questions, keep facts from ALL relevant articles so the answerer can connect them.

The draft key information, in markdown format:

Figure 15: Fact extraction prompt for HotpotQA.