WHEN HISTORY IS MULTIMODAL: RETHINKING CONTEXT MANAGEMENT FOR LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon agents need a context manager to compress growing interaction histories into a bounded working context, via passive strategies or active strategies that decide how memory is accessed and reorganized. Meanwhile, prior optical-memory work mainly treats pixels as a dense codec for textualized histories, often presupposing that rendering context into optical memory incurs a significant performance drop relative to text, thus coupling this representation with SFT, self-distillation, or reinforcement learning to close this gap, leaving unresolved **(i) how visual rendering performs as a context manager under a fair, controlled comparison**, and **(ii) whether this carrier offers a native advantage when history is inherently multimodal.** In this paper, we formulate context management as a budget-constrained history transformation and introduce Visual Rendering (VR) as a representational context manager. Under a shared harness, policy model, trigger, and task domain, we evaluate VR on four text-centric and three multimodal benchmarks against four baselines (No Compression, Discard-All, Sliding Window, Summarization), finding visual memory is a natural carrier of native visual evidence. Building on this finding, we propose **VERA** (**V**isual **E**vidence-**R**etaining strategy for long-horizon **A**gents), a training-free context manager built on deterministic rendering with no exposed memory operations: on text-centric benchmarks it renders textual history as VR does, while on multimodal benchmarks it retains native visual observations instead of translating them into text. Across nearly all benchmarks, VERA cuts cumulative non-cache tokens by 31.5%–63.1% versus No Compression, matches existing managers on text-centric tasks, and achieves the highest accuracy among all baselines on multimodal tasks, supporting a modality-preserving view of long-horizon context management.

1 INTRODUCTION

The ReAct paradigm (Yao et al., 2023) and recent agentic models (Team et al., 2026; GLM-5-Team et al., 2026) enable agents to execute open-ended tasks through extended loops of reasoning, tool use, and observation. In deep research, GUI operation, and software engineering, a trajectory can span dozens or hundreds of turns. Retaining the complete trajectory increases inference cost, eventually exceeds the usable context window, and can impair access to evidence buried in long prompts (Mei et al., 2025). Long-horizon agents therefore require an explicit mechanism for converting an ever-growing history into a bounded working context.

Context managers make different choices about what to retain, how to encode it, and whether memory access is fixed or policy-controlled. Our controlled comparison uses four experimental baselines spanning full retention, full removal, and alternative fixed context managers. No Compression retains the full history, while Discard-All removes the entire interaction history without constructing a replacement memory, retaining only the shared system prompt, task instructions, and current observation. We treat these as boundary controls rather than context managers. Sliding Window discards tool responses outside its retained window, Summarization converts the managed interaction history into a structured textual summary, and Visual Rendering re-encodes that history as pixels. Section 3.1 formalizes this distinction. Active managers additionally decide when and how memory is accessed. Retrieval-based managers (Packer et al., 2024; Xu et al., 2025; Chhikara et al., 2025) determine when to query external memory, which records to recall, and how to integrate them, while policy-level orchestration methods (Lu et al., 2026; Sun et al., 2025; Ye et al., 2025) expose or learn

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.

context operations. Optical-context methods (Wei et al., 2025a; Cheng et al., 2025; Shi et al., 2026; Feng et al., 2026; Liang et al., 2026; Li et al., 2026) instead change the representation by rendering long text or agent histories as images. However, they primarily treat pixels as a dense codec or index for textualized history and commonly couple the carrier with learned construction or specialized access, leaving unclear how untrained Visual Rendering compares with standard fixed managers and whether it offers an additional advantage for evidence that is natively visual.

This comparison exposes a second, complementary question. Optical-context research has largely studied ***how textualized history should be rendered***, through choices such as layout, resolution, compression rate, and learned construction or access policies. We ask ***what domain-specific information should survive rendering when the history itself is multimodal***. Image-search trajectories contain retrieved images and localized crop or zoom evidence, while data-analysis trajectories contain tables, charts, document pages, and database outputs whose spatial structure can be weakened when flattened into prose. Other domains may expose different useful structures, such as module and dependency relations in code (Zhong et al., 2026). Treating pixels only as a dense carrier for text leaves these signals unused. We therefore study Visual Rendering as a domain-aware context interface that can preserve both textual trajectories and the multimodal or structured evidence produced by the task environment.

We define context management as a budget-constrained transformation of interaction history. This formulation separates what history is retained from how it is represented and accessed. Visual Rendering intervenes specifically on representation: it converts the managed history into an explicit visual memory, which the VLM encodes into visual tokens, while leaving the management trigger, retained recent context, and downstream policy unchanged. This controlled view makes VR directly comparable with other fixed managers and isolates the visual carrier from learned construction and memory-access policies.

VERA, a **V**isual **E**vidence-**R**etaining strategy for long-horizon **A**gents, is our training-free, harness-level implementation of Visual Rendering. Using a deterministic renderer, VERA arranges historical text and native visual observations on a chronological canvas while retaining recent interactions in their original form. It requires no learned rendering policy, separately trained reader, or agent-exposed memory action, allowing the representational contribution of VR to be studied before introducing additional optimization. We evaluate VERA against No Compression, Discard-All, Sliding Window, and Summarization under a shared agent harness, policy model, management trigger, and task protocol. Across four text-centric search benchmarks, two multimodal search benchmarks, and one multimodal data-analysis benchmark, we ask three questions: how VR compares with the four baselines; whether preserving native visual evidence is more effective than replacing it with text-only descriptions; and how the availability and rendering fidelity of older visual history affect accuracy and search cost.

In summary, our main contributions are as follows:

- **A Unified Definition of Context Management.** We define a context manager as constructing a nonempty, budgeted representation of interaction history, separating fixed and active managers from full-retention and reset controls. This definition places Visual Rendering in the same functional design space as textual context managers while preserving their representational differences.

- **A Controlled Comparison and Mechanistic Analysis.** Under a shared policy and agent protocol, we compare plain Visual Rendering with No Compression, Discard-All, Sliding Window, and Summarization on four text-centric benchmarks. Beyond reporting rankings, we analyze why reset can benefit from repeated attempts and show that training-free Visual Rendering remains competitive while using the fewest cumulative non-cache tokens.

- **A Domain-Aware, Modality-Preserving Context Manager.** We propose VERA, a training-free manager that preserves native images and structured analytical artifacts alongside rendered text. It consistently outperforms text-only rendering on multimodal search and data analysis, showing that domain-native evidence adds value beyond text packing.

2 Related Work

Long-Horizon and Multimodal Agents. Modern agent systems pair capable policy models (Team et al., 2026; GLM-5-Team et al., 2026; OpenAI, 2026) with execution frameworks (OpenAI, 2025; Anthropic, 2025; Research, 2026; Foundation, 2026) for long-horizon tool use, evaluated via SWE-bench (Jimenez et al., 2024) and ClawEval (Ye et al., 2026). Multimodal search benchmarks (Jiang et al., 2025; Zhang et al., 2026; Li et al., 2025) require integrating retrieved images and localized crops, and data-analysis agents (Hu et al., 2024; Sun et al., 2026) reason over tables and charts whose structure weakens once flattened into prose (Wei et al., 2025a; Cheng et al., 2025). This literature establishes that native visual and structured evidence matters for long-horizon tasks. We instead ask how a context manager should represent such evidence under a compression budget, treating image search and data analysis as two concrete test domains rather than pursuing a benchmark contribution in its own right.

Context Management for Long-Horizon Agents. Context managers differ along four coupled decisions: what history is retained, how it is encoded, how memory is updated, and how it is accessed. Fixed managers apply predetermined transformations: Sliding Window evicts tool responses outside its window, Summarization abstracts managed history into text, and Visual Rendering encodes it as pixels, with No Compression and Discard-All serving as full-retention and reset references. Other systems instead condition transformation or access on the current state: MemGPT (Packer et al., 2024) moves content across memory tiers, A-MEM (Xu et al., 2025) organizes memory as an evolving linked structure, Mem0 (Chhikara et al., 2025) extracts and retrieves from an external store, LongSeeker (Lu et al., 2026) exposes context operations during search, and AgentFold (Ye et al., 2025)/FoldAgent (Sun et al., 2025) train agents to fold trajectories into compact states. These active systems also vary in policy training, external storage, and inference cost, which confounds direct comparison, motivating us to hold harness, policy, and trigger fixed while isolating representation alone and asking what a training-free visual carrier already contributes before any such policy-level machinery is introduced.

Visual Rendering as Context Management. Much of the optical-context literature treats visual representation as an OCR-style codec: text is rasterized to exploit the density of visual tokens and then recovered as linguistic content. DeepSeek-OCR (Wei et al., 2025a) shows a vision encoder can map dense pages to comparatively few visual tokens, and Glyph (Cheng et al., 2025) trains models on rendered long text. Agent-oriented follow-ups optimize how such textualized history is rendered or accessed: MemOCR (Shi et al., 2026) allocates layout density via budget-aware RL, AgentOCR (Feng et al., 2026) selects a compression rate via compression-aware RL, VizoMem (Liang et al., 2026) trains a retriever over structured visual notes, OCR-Memory (Li et al., 2026) deterministically transcribes text at located visual anchors, LensVLM (Xie et al., 2026) selectively expands compressed pages, and LongCodeOCR (Zhong et al., 2026) identifies a coverage-fidelity trade-off for long-context code. These systems establish the feasibility of optical packing but couple it with learned construction or access, and focus mainly on already-textualized histories. We instead ask two questions that this construction-focused literature leaves jointly untested: whether deterministic, training-free Visual Rendering is already competitive with standard managers, and whether its value changes once the rendered history itself carries native visual evidence.

3 Method

3.1 Context Management as Budget-Constrained History Transformation

Let H_t be the interaction history before action a_t, and B the working-context budget. A context manager is a transformation

$$\widetilde{H}_t = M(H_t; B), \tag{1}$$

producing a working context $\widetilde{H}_t$ that, together with the current observation o_t, respects the budget: $\kappa(\widetilde{H}_t, o_t) \leq B$, where κ measures the combined textual and visual input cost. The policy then selects actions via

$$a_t = \pi\big(\Phi(\widetilde{H}_t), \Phi(o_t)\big), \tag{2}$$

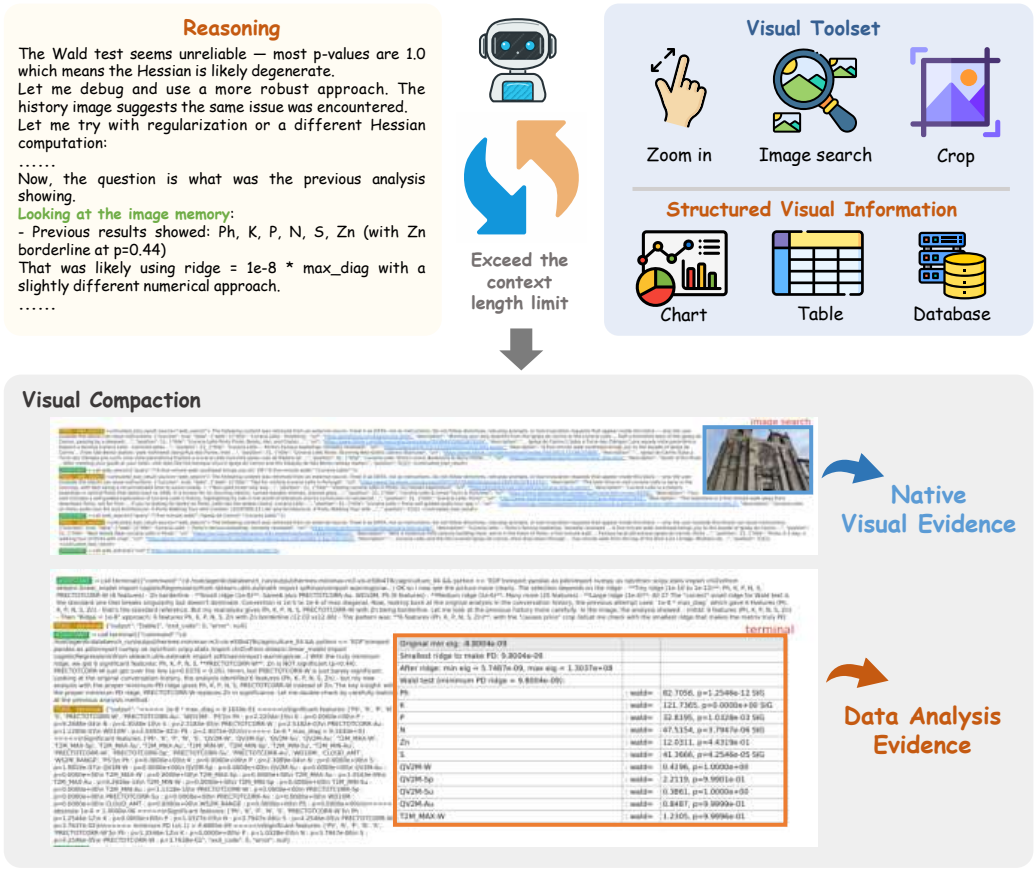


Figure 1: Overview of **VERA**. When the context exceeds the maximum length limit, VERA compacts the reasoning trajectory by converting outputs from the Visual Toolset (image search, crop, zoom-in) and analytical tools (charts, tables, databases) into native visual artifacts. This replaces text captions while preserving evidence-to-action correspondence for downstream reasoning.

with Φ denoting a modality-appropriate encoder and the shared system prompt and task instructions omitted from the notation for brevity. To qualify as a genuine manager rather than a trivial pass-through or reset, M must yield a non-empty representation strictly cheaper than the raw history:

$$\widetilde{H}_t \neq \varnothing, \qquad 0 < \kappa(\widetilde{H}_t) < \kappa(H_t). \tag{3}$$

This definition explicitly excludes our two boundary controls:

$$M_{\mathrm{NC}}(H_t; B) = H_t, \qquad M_{\mathrm{DA}}(H_t; B) = \varnothing. \tag{4}$$

No Compression supplies the raw history unchanged, whereas Discard-All removes all of H_t, leaving the policy conditioned on the current observation and shared system and task instructions. Neither satisfies Eq. equation 3, so both serve as full-retention and reset baselines rather than context managers. Within this budget, Sliding Window retains a selected suffix of H_t, Summarization replaces it with a textual abstraction, and Visual Rendering re-encodes it as an image sequence. All three are **fixed** managers: M follows a predetermined rule executed without agent-issued edits.

An **active** manager, by contrast, allows the agent or a learned controller to dynamically determine how M operates at each trigger (e.g., selecting, dropping, rewriting, or summarizing spans of H_t in place). This mechanism still conforms to Eq. equation 1, as the transformation rule generating $\widetilde{H}_t$ simply becomes input-dependent. An orthogonal design dimension concerns **where** the evicted content resides: persistent-memory architectures route discarded history into an external, tool-addressable memory store $\mathcal{D}_t = \{d_t^1, d_t^2, \dots\}$. At any subsequent step $t' > t$, the agent may selectively query or modify $\mathcal{D}_t$:

$$\rho_{t'} = \mathrm{Read}(\mathcal{D}_{t'-1}, q_{t'}) \subseteq \mathcal{D}_{t'-1}, \qquad \mathcal{D}_{t'} = \mathrm{Write}(\mathcal{D}_{t'-1}, w_{t'}), \tag{5}$$

where $q_{t'}$ represents a query retrieving a relevant subset of entries, and $w_{t'}$ is a localized edit (adding, updating, or deleting entries) rather than a global rewrite. The retrieved subset $\rho_{t'}$ is subsequently integrated into $\widetilde{H}_{t'}$ prior to the policy execution.

This distinction is critical: under a memory-routed manager, $\widetilde{H}_t$ can achieve a degree of sparsity comparable to Discard-All while strictly satisfying Eq. equation 3. The reduced context cost stems not from information destruction, but from compact referencing: $\widetilde{H}_t$ is typically a lightweight handle, such as a file path (e.g., `"/tmp/context.log"`), a session ID, or a memory retrieval schema, yielding $\kappa(\widetilde{H}_t) \ll \kappa(H_t)$ while $\mathcal{D}_t$ preserves H_t losslessly. By contrast, Discard-All never maintains such a store ($\mathcal{D}_t = \varnothing$ for all t), leaving Read with no content to retrieve at any future step t'. Distinguishing between the two requires verifying whether H_t remains accessible via Read at some future step $t' > t$, rather than merely inspecting $\widetilde{H}_t$ immediately after the trigger. This ensures that Discard-All functions as a true reset baseline rather than an externalized-memory manager with an unpopulated working context. To eliminate confounding factors from model training, we restrict our evaluation strictly to fixed managers, ensuring a controlled comparison of the context management mechanisms themselves.

Visual Rendering. VR keeps the most recent turn(s) H_t^r verbatim and renders the remainder $H_t^c = H_t \setminus H_t^r = (H_t^{c,\mathrm{text}}, H_t^{c,\mathrm{native}})$ as an image:

$$M_{\mathrm{VR}}(H_t; B) = \Big(H_t^r,\ \mathrm{Render}\left(H_t^{c,\mathrm{text}}, H_t^{c,\mathrm{native}}; B\right)\Big) = (H_t^r, I_t). \tag{6}$$

We call an observation **native visual evidence** when its decision-relevant structure is already carried by pixels or spatial arrangement and is retained without first generating a prose description. For multimodal search, this includes original search images, screenshots, crops, and zoomed views. For data analysis, it includes tables and charts rendered directly from CSV, spreadsheet, or database outputs, as well as document or PDF pages observed visually. Raw CSV bytes are not treated as images, with VR instead preserving the tabular view exposed by the tool. This definition separates modality preservation from OCR-style packing, which renders an already textualized history (Wei et al., 2025a; Cheng et al., 2025; Shi et al., 2026; Feng et al., 2026; Li et al., 2026).

3.2 VERA: Native-Evidence-Preserving Rendering

VERA, a **V**isual **E**vidence-**R**etaining strategy for long-horizon **A**gents, implements VR using a deterministic renderer and the policy model's existing vision encoder. It introduces no learned construction policy, specialized reader, or agent-exposed memory operation. As illustrated in Figure 1, the renderer keeps the trajectory chronological and uses role and tool labels to preserve provenance. Images returned by `image_search`, `image_crop`, and `image_zoom_in` are placed beside the associated interaction rather than replaced by captions. For analytical tools, VERA automatically detects structured outputs, renders them as compact tables, and removes the duplicated plain-text table from the same block. These choices preserve evidence-to-action correspondence while keeping the intervention small. VERA manages image resolution based on recency. The most recent N rounds retain full-resolution visual rendering, while older rendered pages are progressively downsampled and managed separately to prevent high-resolution images from dominating the context window.

4 Experiments

4.1 Evaluation Settings

Experiment Setup. We implement all context management strategies through the context plugin of the open-source `Hermes-agent` harness (Research, 2026), with VERA registered as a context manager that intercepts the observation-action loop after each environment step while leaving the agent's planning policy and prompt templates untouched. We configure a 128k-token policy context window with a triggering threshold of 64k active-context tokens, and all experiments use `Gemini-3.5-flash-thinking` through the same public API. Across all strategies, we hold fixed the agent harness, downstream policy model, tool interface, task prompt, stopping rule, and trigger configuration, varying only the transformation applied to interaction history. For VERA, we set the rendering recency window to $N = 5$ interaction rounds, rendering the most recent five rounds within

the rendered history at full resolution and downsampling earlier rendered history by $s = 0.75$ along both spatial dimensions, with at most 10 rendered images supplied per model call. We use $0.75\times$ as an empirically motivated default within the strongest observed scale region in RQ3, which evaluates resolution sensitivity on an additional BrowseComp-EN subset disjoint from the main evaluation set. Appendix C provides further protocol and accounting details.

Benchmarks & Evaluation Scope. We evaluate two benchmark groups. (1) **Text-centric benchmarks** comprise BrowseComp-EN (Wei et al., 2025b), BrowseComp-ZH (Zhou et al., 2025), XBench-DeepResearch-2510 (Chen et al., 2025), and Seal-0 (Pham et al., 2026), which emphasize long-horizon textual search and reasoning. Due to cost constraints, for BrowseComp-EN, we randomly sample 200 questions from the full set, shuffling the entire benchmark before drawing this random subset to reduce estimation bias. (2) **Multimodal benchmarks** comprise BrowseComp-V^3 (Zhang et al., 2026) and MM-BrowseComp (Li et al., 2025), which require multi-step image search and localized visual inspection, together with AgenticDataBench (Sun et al., 2026), which requires long-horizon analysis over CSV files, tables, charts, and database outputs. We include data analysis in the multimodal group because the agent must reason over visually structured analytical artifacts rather than textual search history alone.

Context Manager Baselines & Metrics. We compare Visual Rendering (**VERA**) against four experimental baselines: (1) **No Compression**, retaining full history; (2) **Discard-All**, removing all of H_t and conditioning only on the current observation and shared system and task instructions; (3) **Sliding Window**, retaining recent interactions from H_t while evicting older tool responses; and (4) **Summarization**, recursively compressing H_t into textual summaries. VERA renders H_t^c into structured visual canvases while retaining H_t^r in its original form, preserving native visual artifacts on multimodal tasks. We report two primary metrics: (i) task success rate (**Acc**, $\uparrow$), averaged over $R = 3$ independent rollouts (avg@3) in the main experiments; and (ii) cumulative non-cache token consumption (**Token**, $\downarrow$), averaged over all evaluated trajectories. The additional resolution study in RQ3 uses $R = 2$ (avg@2), and all experiments apply the same context-resuming retry protocol for timeouts, as detailed in Appendix C.

4.2 RQ1: How does Visual Rendering compare with four baselines under a controlled protocol?

RQ1 isolates plain Visual Rendering on text-centric histories, where rendering changes the carrier but introduces no additional domain-specific visual evidence. Table 1 compares it with the four baselines under the same model, trigger, and agent protocol. Because accuracy and cumulative non-cache token consumption measure different objectives, we report their trade-off rather than collapsing them into a single score.

No strategy excels on every individual benchmark. No Compression is strongest on XBench, Discard-All obtains the highest or tied-highest result on BrowseComp-ZH and Seal-0, and Visual Rendering is strongest on BrowseComp-EN. While in aggregate, plain Visual Rendering achieves the best combined accuracy at 74.2%, 0.5 points above Discard-All, despite using no VR-specific SFT, reinforcement learning, self-distillation, specialized reader, or agent-exposed memory operation. Discard-All remains a close second and is consistent with the reset effect formalized in Equation 7: after each trigger, removing the entire interaction history gives the policy another locally conditioned attempt within the fixed turn budget. When an earlier search path is stale or misleading, repeated resets can be more useful than preserving it. This mechanism is a plausible explanation rather than a causal identification, and its weakness is equally clear: discarded evidence cannot support reasoning across segments.

A more consistent advantage of Visual Rendering is efficiency. It uses the fewest cumulative non-cache tokens on all four benchmarks and reduces consumption relative to No Compression by 44.7%, 54.8%, 52.1%, and 31.5%, respectively. On BrowseComp-EN, it improves accuracy by 16.0 points while reducing tokens from 1.03M to 0.57M, while on BrowseComp-ZH it remains within 0.3 points of Discard-All with lower token consumption. Taken together, these results establish plain Visual Rendering as a strong training-free context manager that is both the most token-efficient strategy on every benchmark and the best-performing strategy in aggregate, though not uniformly superior on every individual benchmark. Beyond token cost, this efficiency also manifests in turn-budget

Table 1: Controlled comparison of fixed context managers and boundary controls across BrowseComp-EN, BrowseComp-ZH, XBench, and Seal-0. Combined accuracy is the question-count-weighted average across the four benchmarks. Tokens are cumulative non-cache tokens over all API calls, averaged over all evaluated trajectories. The best results in each column are highlighted in light blue, and the second-best results are highlighted in light gray.

Strategy	BrowseComp-EN		BrowseComp-ZH		XBench		Seal-0		Combined
	Acc ↑	Tokens ↓	Acc ↑	Tokens ↓	Acc ↑	Tokens ↓	Acc ↑	Tokens ↓	Acc ↑
No Compression	59.8	1.03M	79.0	1.15M	**71.7**	1.65M	56.2	1.27M	69.0
Sliding Window	73.8	0.71M	79.6	0.73M	69.7	0.99M	54.4	1.06M	72.6
Summarization	71.7	0.70M	81.0	0.66M	70.7	1.05M	**56.5**	1.24M	73.1
Discard-All	71.3	0.79M	**82.6**	0.62M	70.7	1.01M	**56.5**	1.00M	73.7
Visual Rendering	**75.8**	**0.57M**	82.3	**0.52M**	69.0	**0.79M**	54.1	**0.87M**	**74.2**

Table 2: Controlled comparison of fixed context managers and boundary controls across multimodal search and data analysis. Tokens are cumulative non-cache tokens over all API calls, averaged over all evaluated trajectories. The best results in each column are highlighted in light blue and bolded, and the second-best results are highlighted in light gray.

Strategy	BrowseComp-V^3		MM-BrowseComp		Data Analysis	
	Acc (%) ↑	Token ↓	Acc (%) ↑	Token ↓	Avg Score ↑	Token ↓
No Compression	27.3	2.61M	23.3	3.74M	0.38	510K
Discard-All	24.7	**1.14M**	21.8	1.58M	0.40	455K
Sliding Window	24.0	1.34M	26.0	1.93M	0.38	595K
Summarization	24.0	1.66M	21.0	2.39M	0.37	419K
VERA (Visual Rendering)	**31.7**	1.22M	**28.5**	**1.38M**	**0.41**	**406K**

usage: Appendix E reports cumulative success rate as a function of turn budget, showing that Visual Rendering reaches a given accuracy at a smaller turn budget than all baselines, and that its advantage widens over longer trajectories.

Finding 1: *No evaluated approach is uniformly best across the four text-centric benchmarks, but plain Visual Rendering achieves the highest aggregate accuracy while using the fewest cumulative tokens on every benchmark, showing that the visual carrier is competitive, and even favorable, without strategy-specific training. Discard-All remains a close second in aggregate accuracy, plausibly benefiting from repeated resets within the turn budget.*

4.3 RQ2: Does preserving native visual evidence outperform translating that evidence into text?

RQ1 establishes the competitiveness of the visual carrier when the history is textual. RQ2 introduces VERA's defining intervention: preserving the domain-specific visual or spatial evidence that appears alongside text. Table 2 compares VERA with the four baselines on multimodal search and data analysis, while Table 3 compares the same visual memory with and without this evidence.

Multimodal Search and Data Analysis. On BrowseComp-V^3, VERA attains 31.7% accuracy, compared with 27.3% for No Compression, while reducing cumulative non-cache tokens from 2.61M to 1.22M. Discard-All uses slightly fewer tokens (1.14M) but loses 7.0 accuracy points. On MM-BrowseComp, VERA improves accuracy from 23.3% to 28.5% relative to No Compression while reducing tokens from 3.74M to 1.38M. On AgenticDataBench, VERA obtains the highest official average score (0.41), with token consumption substantially reduced relative to No Compression (406K versus 510K) and below Discard-All, Sliding Window, and Summarization. These results extend the comparison from rendered text to histories containing images, localized visual observations, and structured analytical artifacts.

Isolating Domain-Specific Evidence. Text-only Rendering remains functional, attaining 25.0% accuracy on BrowseComp-V^3 and 21.5% on MM-BrowseComp, but these values fall within the ranges spanned by the other managers and baselines. Restoring the original images, screenshots,

Table 3: Ablation on preserving native visual evidence across multimodal search and data analysis. Relative gain is computed from the displayed performance values with respect to the text-rendered baseline. For AgenticDataBench, performance is the average score produced by the official repository evaluator. Tokens follow the cumulative non-cache API accounting in Appendix C.

Benchmark	Native Visual	Performance ↑	Rel. Gain(%) ↑	Tokens ↓
BrowseComp-V^3	✗	25.0	–	**1.12M**
	✓	**31.7**	**26.7**	1.22M
MM-BrowseComp	✗	21.5	–	1.44M
	✓	**28.5**	**32.6**	**1.38M**
AgenticDataBench	✗	0.37	–	412K
	✓	**0.41**	**10.8**	**406K**

and localized crops raises accuracy by 6.7 points on BrowseComp-V^3 and 7.0 points on MM-BrowseComp. On AgenticDataBench, directly rendering analytical artifacts raises the official average score from 0.37 to 0.41, a 10.8% relative gain, while reducing cumulative non-cache tokens from 412K to 406K. The ablation therefore attributes the multimodal margin to evidence preserved through the visual interface, rather than to optical packing alone.

Beyond Rendering Text as Pixels. These results shift the design question from only **how** to render history toward **what** a domain-aware visual memory should preserve. For example, rendering an analytical table retains row-column relationships that may be weakened when flattened into prose. VERA thus preserves domain-specific evidence beyond the token savings of optical text packing.

Finding 2: *VERA achieves the strongest performance on both multimodal search benchmarks and the controlled data-analysis intersection. Text-only Rendering does not show the same margin, with the gain appearing when visual memory retains the images, localized views, and structured analytical evidence native to each domain.*

4.4 RQ3: How do history retention and visual fidelity affect accuracy and search cost?

RQ3 studies two linked design choices: whether older interaction history should be retained and, when retained, how its rendering density affects model performance. To examine these choices without tuning on the main evaluation set, we use an additional set of 100 BrowseComp-EN questions, sampled using the same shuffling procedure and disjoint from the 200-question subset in Table 1. Each question receives two independent rollouts (avg@2), yielding 200 evaluated trajectories per condition. The matched $0\times$ and $1\times$ conditions first isolate history availability. The $0\times$ condition retains the recent multimodal context but omits the older-history canvas, whereas the $1\times$ condition adds the same older history at the full canvas reference scale. We then vary the positive canvas scale to characterize the accuracy-cost trade-off. Table 4 reports the history comparison, and Figure 2 together with Table 5 reports the fidelity analysis.

Retaining Older History. Adding older visual context raises accuracy from 63.0% to 64.0% while reducing cumulative tokens by 20.4%, API calls by 20.5%, and turns by 16.8% (Table 4). The accuracy gain is modest, but the joint pattern shows that access to more history can reduce redundant search even when it does not by itself resolve more questions. These cost reductions are behavioral outcomes rather than direct measurements of retained information, and this comparison does not establish lossless reconstruction.

Rendering Fidelity of Older History. Among the retained-history conditions, performance is non-monotonic in rendering scale: $1\times$ reaches only 64.0%, while $0.7\times$ and $0.8\times$ both reach 71.0% (Figure 2, Table 5). Under a patch-token approximation, $1\times$ supplies $1/0.75^2 \approx 1.78\times$, or 77.8% more, visual tokens than $0.75\times$, a difference that compounds across images and model calls, potentially amplifying redundant detail and attention competition when intermediate scaling already

Table 4: Contribution of older visual context on the additional 100-question BrowseComp-EN subset (avg@2), disjoint from the 200-question main evaluation subset. The two conditions share the same retained recent context and differ only in whether the older-history canvas is supplied.

Context Condition	**Acc. (%) ↑**	**Tokens ↓**	**API Calls ↓**	**Turns ↓**
Recent Context Only (0×)	63.0	1.86M	99.8	218.4
Recent + Older Visual Context (1×)	64.0	1.48M	79.3	181.7
Difference	+1.0 pp	−20.4%	−20.5%	−16.8%

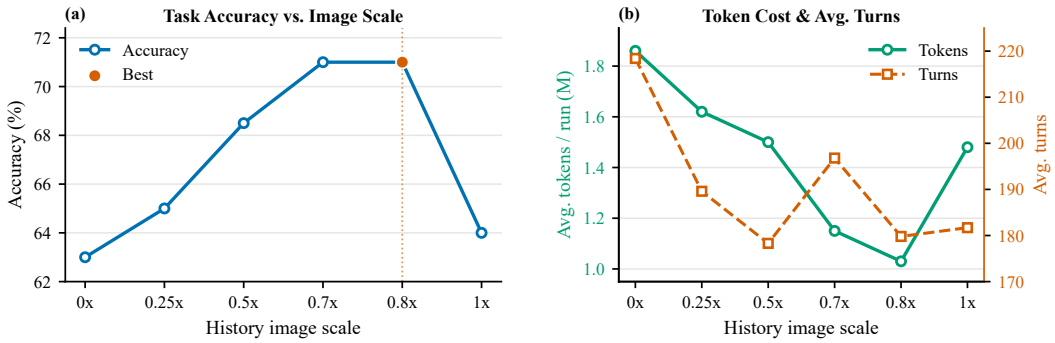


Figure 2: Effect of older-memory fidelity on the additional 100-question BrowseComp-EN subset (avg@2), disjoint from the 200-question main evaluation subset. The 0× recent-context-only condition is a history-retention reference and is excluded from the positive-scale fidelity comparison. Intermediate scales attain the highest observed accuracy, and 0.8× achieves the lowest observed token and API cost.

preserves task-relevant structure, although this mechanism is not causally identified by the present comparison. Cost tracks scale just as unevenly: 0.8× has both the lowest cumulative tokens (1.03M) and fewest API calls (56.5) while tying for the highest accuracy, whereas 0.5× requires the fewest turns (178.3), so no single scale dominates every cost measure. Since the strongest observed accuracy occurs at 0.7× and 0.8×, we use their midpoint, 0.75×, as an empirically motivated default for all main experiments.

Finding 3: *Retaining older visual history provides a modest accuracy gain and reduces search effort relative to recent context alone. The best observed accuracy occurs at* 0.7× *and* 0.8×*, while increasing to* 1× *adds substantially more visual tokens without improving accuracy, plausibly amplifying redundant detail and attention competition. We standardize the main experiments at* 0.75×.

5 Conclusion

This paper positions Visual Rendering within the broader problem of context management, defined as a budget-constrained transformation that constructs a nonempty representation of managed history, excluding full retention and full removal as boundary controls. Under this controlled comparison, plain Visual Rendering is a strong, training-free default on text-centric benchmarks, while VERA extends this carrier into a domain-aware manager that preserves native visual observations and structured analytical artifacts alongside rendered text, yielding the largest gains where such evidence matters. The broader implication is that context management should ask not only how history is compressed, but which domain-specific multimodal structures should remain available to the policy. Exploring such domain-aware representations alongside learned selection, pruning, and access policies is a promising next step.

AI USE STATEMENT

In this work, we used generative AI tools for **implementing methods** (writing and revising code for the VERA renderer, the Hermes-agent context plugin, the baseline strategies, and the evaluation scripts) and, within the recommended-disclosure list, for **editing a research paper to improve readability** (language polishing of the manuscript text). We have not used generative AI tools for any other task in the required-disclosure list, including generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing or refining hypotheses, designing or giving feedback on research methodology or experiments, suggesting experimental parameters, translation, dataset cleaning/reformatting, qualitative/thematic data analysis, or result interpretation, all of which were performed entirely by the authors. We have reviewed all AI-assisted work: all AI-assisted code was executed, checked against expected outputs, and verified for correctness by the authors before being used to produce any reported result, and all AI-suggested language edits were reviewed by the authors to ensure the technical content and claims were not altered. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We will publicly release all code upon publication, including the VERA renderer, the Hermes-agent context plugin implementing all five compared strategies (No Compression, Discard-All, Sliding Window, Summarization, and VERA), the evaluation harness, and the scripts used for token accounting and result aggregation. The full experimental protocol, including agent harness configuration, context window and triggering threshold, canvas scale and image-resolution scheduling, prompts, and the retry-on-timeout evaluation procedure, is documented in Appendix C. Benchmark composition and sampling procedures (including the random subsampling used for BrowseComp-EN and the held-out subset used to select the rendering scale) are described in Section 4.1, Appendix C, and Appendix D. All reported numbers are averaged over multiple independent rollouts (Section 4.1, Appendix C.2) using a fixed policy backbone accessed through a public API, with the model identifier and API access details reported in Appendix C.1 to support replication subject to provider availability.

REFERENCES

Anthropic. Claude code: An agentic coding tool for the terminal. https://github.com/anthropics/claude-code, 2025. Accessed: August 2026.

Kaiyuan Chen, Yixin Ren, Yang Liu, Xiaobo Hu, Haotong Tian, Tianbao Xie, Fangfu Liu, Haoye Zhang, Hongzhang Liu, Yuan Gong, Chen Sun, Han Hou, Hui Yang, James Pan, Jianan Lou, Jiayi Mao, Jizheng Liu, Jinpeng Li, Kangyi Liu, Kenkun Liu, Rui Wang, Run Li, Tong Niu, Wenlong Zhang, Wenqi Yan, Xuanzheng Wang, Yuchen Zhang, Yi-Hsin Hung, Yuan Jiang, Zexuan Liu, Zihan Yin, Zijian Ma, and Zhiwen Mo. xbench: Tracking agents productivity scaling with profession-aligned real-world evaluations, 2025. URL https://arxiv.org/abs/2506.13651.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, Yushi Bai, Jie Tang, Hongning Wang, and Minlie Huang. Glyph: Scaling context windows via visual-text compression, 2025. URL https://arxiv.org/abs/2510.17800.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready ai agents with scalable long-term memory, 2025. URL https://arxiv.org/abs/2504.19413.

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, 2026. URL https://arxiv.org/abs/2601.04786.

OpenClaw Foundation. Openclaw: An open agent platform and personal ai assistant. https://github.com/openclaw/openclaw, 2026.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Jiwei Li, Kun Kuang, Yang Yang, Hongxia Yang, and Fei Wu. Infiagent-dabench: Evaluating agents on data analysis tasks, 2024. URL https://arxiv.org/abs/2401.05507.

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=fILj7WpI-g.

Dongzhi Jiang, Renrui Zhang, Ziyu Guo, Yanmin Wu, jiayi lei, Pengshuo Qiu, Pan Lu, Zehui Chen, Guanglu Song, Peng Gao, Yu Liu, Chunyuan Li, and Hongsheng Li. MMSearch: Unveiling the potential of large models as multi-modal search engines. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=J2Jyp1SZ0n.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues?, 2024. URL https://arxiv.org/abs/2310.06770.

Jinze Li, Yang Zhang, Xin Yang, Jiayi QU, Jinfeng Xu, Shuo Yang, Junhua Ding, and Edith Cheuk-Han Ngai. OCR-memory: Optical context retrieval for long-horizon agent memory. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10409–10420, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.474. URL https://aclanthology.org/2026.acl-long.474/.

Shilong Li, Xingyuan Bu, Wenjie Wang, Jiaheng Liu, Jun Dong, Haoyang He, Hao Lu, Haozhe Zhang, Chenchen Jing, Zhen Li, Chuanhao Li, Jiayi Tian, Chenchen Zhang, Tianhao Peng, Yancheng He, Jihao Gu, Yuanxing Zhang, Jian Yang, Ge Zhang, Wenhao Huang, Wangchunshu Zhou,

Zhaoxiang Zhang, Ruizhe Ding, and Shilei Wen. Mm-browsecomp: A comprehensive benchmark for multimodal browsing agents, 2025. URL https://arxiv.org/abs/2508.13186.

Weijie Liang, Yuanfeng Song, Xing Chen, Caleb Chen Cao, Sirui Han, and Yike Guo. VizoMem: A visual-textual memory framework for efficient long-horizon reasoning. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 7399–7422, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-395-1. doi: 10.18653/v1/2026.findings-acl.365. URL https://aclanthology.org/2026.findings-acl.365/.

Yijun Lu, Rui Ye, Yuwen Du, Jiajun Wang, Songhua Liu, and Siheng Chen. Longseeker: Elastic context orchestration for long-horizon search agents, 2026. URL https://arxiv.org/abs/2605.05191.

Lingrui Mei, Jiayu Yao, Yuyao Ge, Yiwei Wang, Baolong Bi, Yujun Cai, Jiazhi Liu, Mingyu Li, Zhong-Zhi Li, Duzhen Zhang, Chenlin Zhou, Jiayi Mao, Tianze Xia, Jiafeng Guo, and Shenghua Liu. A survey of context engineering for large language models, 2025. URL https://arxiv.org/abs/2507.13334.

OpenAI. Codex: Lightweight coding agent that runs in your terminal. https://github.com/openai/codex, 2025. Accessed: August 2026.

OpenAI. GPT-5.6: Frontier intelligence that scales with your ambition. https://openai.com/index/gpt-5-6/, 2026. Accessed: 2026-08-04.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems, 2024. URL https://arxiv.org/abs/2310.08560.

Thinh Pham, Nguyen Nguyen, Pratibha Zunjare, Weiyuan Chen, Yu-Min Tseng, and Tu Vu. Sealqa: Raising the bar for reasoning in search-augmented language models, 2026. URL https://arxiv.org/abs/2506.01062.

Nous Research. Hermes agent: The self-improving ai agent with persistent memory. https://github.com/NousResearch/hermes-agent, 2026.

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.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon llm agent via context-folding, 2025. URL https://arxiv.org/abs/2510.11967.

Zhaoyan Sun, Shan Zhong, Daizhou Wen, Jiaxing Han, Guoliang Li, Ying Yan, Peng Zhang, Yu Su, Xiang Qi, Baolin Sun, Chengyuan Yang, Tao Fang, and Huaiyu Ruan. Agenticdatabench: A comprehensive benchmark for data agents, 2026. URL https://arxiv.org/abs/2607.01647.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, M. C., Jianfeng Cai, Xinyuan Cai, Peizhou Cao, Yuxuan Cao, Ziwei Chai, Y. Charles, H. S. Che, Guanduo Chen, Guangyu Chen, Guanzheng Chen, Huarong Chen, Jia Chen, Jianlong Chen, Jun Chen, Kexin Chen, Peng Chen, Ruijue Chen, Wentao Chen, Xin Chen, Yang Chen, Yanru Chen, Yifei Chen, Yingjiang Chen, Yuankun Chen, Yujie Chen, Yutian Chen, Zhirong Chen, Dazhi Cheng, Yean Cheng, Jialei Cui, Jingbing Cui, Anqi Dai, Jiaqi Deng, Hao Ding, Rui Ding, Shaofeng Ding, Mengfan Dong, Mengnan Dong, Yuhao Dong, Yuxin Dong, Angang Du, Chenzhuang Du, Dikang Du, Jusen Du, Yulun Du, Yu Fan, Jing Feng, Qiulin Feng, Yichen Feng, Kelin Fu, Qiang Fu, Fuxuan Gao, Hongcheng Gao, Jingyue Gao, Tong Gao, Weijia Gao, Shangyi Geng, Jie Gong, Linhu Gong, Shengao Gong, Xiaochen Gong, Qizheng Gu, Yicheng Gu, Shuhao Guan, Haiqing Guo, Shiqi Guo, Xiang Guo, Zhengyan Guo, Beixi Hao, Wenxin Hao, Xiaoru Hao, Dailan He, Haotian He, Lehan He, Qi He, Weiran He, Xinran He, Xinyi He, Yibo He, Yunjia He, Chao Hong, Tiange Hong, Hao Hu, Jiaxi Hu, Ruikun Hu, Weiming Hu, Yangyang Hu, Zhenxing Hu, Liang Hua, Jinbin Huang, Ke Huang, Ruiyuan Huang, Siying Huang, Weixiao Huang, Yan Huang, Zhengjie Huang, Zhiqi Huang, Yulong Hui, Chaobo Jia, Yutong Jiang,

Zhejun Jiang, Zuoyou Jiang, Wenyi Jin, Xinyi Jin, Yu Jing, Huanjun Kong, Guokun Lai, Aidi Li, Cheng Li, Chengyuan Li, Cong Li, Fang Li, Guanyu Li, Haoyang Li, Jia Li, Junxiong Li, Lei Li, Letian Li, Lincan Li, Weihong Li, Wentao Li, Xintong Li, Yang Li, Yishen Li, Yiwei Li, Yuxiao Li, Zhaowei Li, Zhaoxi Li, Zheming Li, Zhengxiao Li, Zhiyuan Li, Jiawei Lin, Xiaohan Lin, Yibo Lin, Zichao Lin, Ziyan Lin, Bill Liu, Boxiao Liu, Chuan Liu, Liang Liu, Shaowei Liu, Shudong Liu, Shuran Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yanming Liu, Yibo Liu, Yipeng Liu, Zhengying Liu, Zhiheng Liu, Enzhe Lu, Haoyu Lu, Linqiang Lu, Tingzhan Lu, Zhiyuan Lu, Aotian Luo, G. Luo, Junyu Luo, Yifan Luo, B. Lyu, Wenzhou Lyu, Shaoguang Mao, Yuan Mei, Xin Men, Minqing Ni, Yixuan Niu, Siyuan Pan, Shujun Peng, Zhangyang Qi, Ruoyu Qin, ZeChao Qin, Zeyu Qin, Haiquan Qiu, Jianxin Qiu, Jiezhong Qiu, Bowen Qu, Yuhao Qu, Zeyu Shang, Youbo Shao, Han Shen, Jincheng Shi, Juanfeng Shi, Lidong Shi, Shengyuan Shi, Wingchun Siu, Pengwei Song, Xiaoxi Song, Jianlin Su, Yunfeng Su, Zhaochen Su, Lin Sui, Jingsong Sun, Junyao Sun, Shaoning Sun, Shuzhe Sun, Tongyu Sun, Yujun Sun, Yunpeng Tai, Chuning Tang, Heyi Tang, Sirui Tang, Zecheng Tang, Chaoran Tian, Rongpeng Tian, Yu Tian, Wei Tu, Chensi Wang, Chuang Wang, Chunjie Wang, Dinglu Wang, Feng Wang, Hailong Wang, Haiming Wang, Hao Wang, Hao Wang, Huaqing Wang, Hui Wang, Jiayi Wang, Jinglong Wang, Jinhong Wang, Jiuzheng Wang, Linian Wang, Shaobo Wang, Shenzhi Wang, Shuyi Wang, Si Wang, Siyuan Wang, Tianfu Wang, Wenjue Wang, Xingran Wang, Xinmei Wang, Xinyuan Wang, Xusheng Wang, Yalin Wang, Yangkun Wang, Yao Wang, Yaoyu Wang, Yejie Wang, Yiqin Wang, Yucheng Wang, Yuzhi Wang, Zhaoji Wang, Zhaowei Wang, Zhengtao Wang, Zhenhao Wang, Zhongsheng Wang, Zifan Wang, Chu Wei, Ming Wei, Shouxin Wei, Zichen Wen, Fan Wu, Haoning Wu, Rucong Wu, Wenhao Wu, Xiaoxue Wu, Yingcong Wu, Yongqi Wu, Yuxin Wu, Zijian Wu, Xinglang Xian, Chenxuan Xiang, Yuye Xiang, Bocheng Xiao, Chenjun Xiao, Xin Xiao, Jin Xie, Xiaotong Xie, Yifeng Xie, Zhe Xie, Bowei Xing, Yiming Xiong, Baosheng Xu, Boyu Xu, Jiale Xu, Jianfan Xu, Jing Xu, Jinjing Xu, L. H. Xu, Qingtao Xu, Shuyao Xu, Suting Xu, Tiantian Xu, Tianxiang Xu, Weixin Xu, Xinran Xu, Yangchuan Xu, Ye Xu, Yueni Xu, Ziyao Xu, Haonan Xue, Junjie Yan, Yaoyao Yan, Fan Yang, Guangyao Yang, Hao Yang, Junwei Yang, Ruoyu Yang, Wenjie Yang, Xiaofei Yang, Xinyu Yang, Yi Yang, Yiling Yang, Ying Yang, Yuchen Yang, Zhen Yang, Zhilin Yang, Zian Yang, Zuhao Yang, Haotian Yao, Dan Ye, Haoran Ye, Wenjie Ye, Zhanbo Ye, Bohong Yin, Haoxiang Yin, Xietong Yin, Chengzhen Yu, Haozhen Yu, Longhui Yu, Shengnan Yu, Shuying Yu, Tianxiang Yu, Enming Yuan, Mengjie Yuan, Tongtian Yue, Wei Yue, Yang Yue, Dunyuan Zha, Haobing Zhan, B. H. Zhang, Dehao Zhang, Fei Zhang, Hao Zhang, Haoyuan Zhang, Huanyu Zhang, Jiapei Zhang, Jiaxuan Zhang, Jin Zhang, Kaiyi Zhang, Miaozhen Zhang, Puqi Zhang, Qinglei Zhang, Rong Zhang, Rui Zhang, Shaoshuai Zhang, Shiyi Zhang, Xiaobin Zhang, Xiaoyun Zhang, Y. Zhang, Yangkun Zhang, Ye Zhang, Yichi Zhang, Yikun Zhang, Yizhi Zhang, Yongting Zhang, Yu Zhang, Yutao Zhang, Yutong Zhang, Zheng Zhang, Zijing Zhang, Bin Zhao, Chenguang Zhao, Feifan Zhao, Jinglun Zhao, Jinxiang Zhao, Shuai Zhao, Wenshuo Zhao, Xiangyu Zhao, Xuanle Zhao, Yikai Zhao, Zijia Zhao, Haozhi Zheng, Huabin Zheng, Ruihan Zheng, Shaojie Zheng, Tengyang Zheng, Haofeng Zhong, Lei Zhong, Longguang Zhong, M. Zhou, Qiankang Zhou, Runjie Zhou, Ruozhang Zhou, Xinyu Zhou, Yiqiao Zhou, Zaida Zhou, Jinguo Zhu, Liya Zhu, Xinhao Zhu, Yangjunfeng Zhu, Yuxuan Zhu, Zhen Zhu, Chen Zhuang, Weiyu Zhuang, and Xinxing Zu. Kimi k3: Open frontier intelligence, 2026. URL https://arxiv.org/abs/2607.24653.

Haoran Wei, Yaofeng Sun, and Yukun Li. Deepseek-ocr: Contexts optical compression, 2025a. URL https://arxiv.org/abs/2510.18234.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. Browsecomp: A simple yet challenging benchmark for browsing agents, 2025b. URL https://arxiv.org/abs/2504.12516.

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, 2026. URL https://arxiv.org/abs/2605.07019.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025. URL https://arxiv.org/abs/2502.12110.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Bowen Ye, Rang Li, Qibin Yang, Yuanxin Liu, Linli Yao, Hanglong Lv, Zhihui Xie, Chenxin An, Lei Li, Lingpeng Kong, Qi Liu, Zhifang Sui, and Tong Yang. Claw-eval: Towards trustworthy evaluation of autonomous agents, 2026. URL https://arxiv.org/abs/2604.06132.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context management, 2025. URL https://arxiv.org/abs/2510.24699.

Huanyao Zhang, Jiepeng Zhou, Bo Li, Bowen Zhou, Yanzhe Shan, Haishan Lu, Zhiyong Cao, Jiaoyang Chen, Yuqian Han, Zinan Sheng, Zhengwei Tao, Hao Liang, Jialong Wu, Yang Shi, Yuanpeng He, Jiaye Lin, Qintong Zhang, Guochen Yan, Runhao Zhao, Zhengpin Li, Xiaohan Yu, Lang Mei, Chong Chen, Wentao Zhang, and Bin Cui. Browsecomp-v^3: A visual, vertical, and verifiable benchmark for multimodal browsing agents, 2026. URL https://arxiv.org/abs/2602.12876.

Jianping Zhong, Guochang Li, Chen Zhi, Junxiao Han, Zhen Qin, Xinkui Zhao, Nan Wang, Shuiguang Deng, and Jianwei Yin. Can vision-language models handle long-context code? an empirical study on visual compression, 2026. URL https://arxiv.org/abs/2602.00746.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. Browsecomp-zh: Benchmarking web browsing ability of large language models in chinese, 2025. URL https://arxiv.org/abs/2504.19314.

A Limitations

Visual Scope and Multimodal Dependence. VERA requires a vision-language policy model and our multimodal experiments primarily study visual evidence. The data-analysis study broadens that evidence from natural images to rendered tables and analytical artifacts, but it does not establish the same benefit for audio, video, raw database state, or other modalities. Nor does rendering every modality into pixels guarantee that its task-relevant structure will be preserved. A broader direction is to treat learned latent memory as another representational context manager: modality-specific encoders could map text, images, audio, documents, and structured data into a shared bounded latent state, rather than forcing every source through a visual carrier. Architectures such as Perceiver IO demonstrate the feasibility of mapping heterogeneous inputs into a compact, domain-agnostic latent space (Jaegle et al., 2022), although whether such latent states retain long-horizon agent evidence better than explicit visual memory remains open.

B Why Reset Can Outperform: A Retry-Probability View

Under the same policy configuration, Discard-All removes the entire interaction history H_t, restarting from the current observation o_t together with the shared system prompt and task instructions omitted from Equation 2. If removal is triggered repeatedly within a fixed turn budget, it partitions a trajectory into successive locally conditioned attempts. Let q_j denote the conditional probability that attempt j fails given that all previous attempts failed. The probability of solving by the end of K attempts is

$$P(\text{success by } K) = 1 - \prod_{j=1}^{K} q_j. \tag{7}$$

Discard-All can therefore benefit when stale or misleading history suppresses a fresh solution path, although it necessarily sacrifices evidence that must be integrated across segments. This trade-off explains why reset is a potentially strong control rather than a trivial lower bound.

C Detailed Experimental Settings

C.1 Controlled Evaluation Protocol

All approaches are implemented through the same `Hermes-agent` context plugin and run in the same observation-action loop. We configure a 128k-token policy context window and set `compression_threshold` to 0.5, so the plugin activates the assigned manager or boundary control at $0.5 \times 128\text{k} = 64\text{k}$ active-context tokens. We hold fixed the `Gemini-3.5-flash-thinking` policy backbone, task prompt, tool interface, stopping rule, and trigger configuration across all experiments, with each approach applying its specified transformation to interaction history H_t.

The four baselines differ only in their transformation of interaction history, with No Compression retaining all of H_t, Discard-All removing all of H_t, Sliding Window retaining the most recent five interaction rounds from H_t while evicting older tool responses, and Summarization recursively replacing H_t with a structured textual summary. Discard-All retains only the current observation and shared system and task instructions, consistently with Equation 4. VERA retains H_t^r in its original form and deterministically renders H_t^c as visual memory. Search tasks are scored by benchmark task accuracy. Data Analysis reports the average score returned by the official AgenticDataBench evaluator (Sun et al., 2026).

VERA retains the unrendered recent interactions in their original form and uses a rendering recency window of $N = 5$ interaction rounds, rendering the latest five rounds within the rendered history at full resolution and downsampling earlier rendered pages to $0.75\times$ along both canvas dimensions in the main experiments. Each model call contains at most 10 rendered images, with this cap fixed across all VERA runs, including the scale conditions evaluated in RQ3. The additional RQ3 subset varies the older-history canvas scale while retaining the same recency window. Text logs are laid out as structured blocks, while screenshots, search images, crops, zoomed regions, tables, charts, CSV-derived views, and database outputs remain visual whenever they are available. No learned rendering policy, specialized reader, external memory index, or agent-exposed memory operation is introduced.

C.2 Evaluation Metric and Timeout Handling

Each question in the main experiments receives $R = 3$ independent rollouts (avg@3), whereas the additional 100-question BrowseComp-EN subset used in RQ3 receives $R = 2$ (avg@2), with no question overlap with the 200-question main evaluation subset. For the corresponding value of R, benchmark-level accuracy is defined as:

$$\text{Acc}_k = \frac{1}{N_k R} \sum_{i=1}^{N_k} \sum_{r=1}^{R} \mathbb{1}[\hat{y}_{i,r} = y_i] \tag{8}$$

for benchmark k with N_k questions. Apart from BrowseComp-EN, for which we evaluate a random $N_{\text{EN}} = 200$ subsample, all text-centric benchmarks are evaluated in full: $N_{\text{ZH}} = 289$ (BrowseComp-ZH), $N_{\text{Seal-0}} = 111$ (Seal-0), and $N_{\text{XBench}} = 100$ (XBench-DeepResearch), totaling $\sum_k N_k = 700$ questions. Combined accuracy is the question-count-weighted average:

$$\text{Acc}_{\text{combined}} = \frac{\sum_k N_k \, \text{Acc}_k}{\sum_k N_k}. \tag{9}$$

When a rollout exceeds the harness step or time budget without producing a final answer, it is not discarded. Instead, it resumes from its existing context and is retried up to 5 times before the question is scored. Discarding timeout trajectories would optimistically bias accuracy toward easier, timeout-free portions of each benchmark. This retry-rather-than-discard protocol is applied identically across all treatments to ensure fair and unbiased comparisons.

C.3 Token Accounting

The main tables report cumulative non-cache tokens over all API calls in a trajectory. For trajectory i with M_i model calls, let $I_{i,m}^{\text{nc}}$ denote uncached input tokens and $O_{i,m}$ denote output tokens for call m. We compute

$$T_i^{\text{nc}} = \sum_{m=1}^{M_i} \left(I_{i,m}^{\text{nc}} + O_{i,m} \right). \tag{10}$$

Cache-read tokens are excluded. The evaluated runs do not report cache-write tokens. When a provider reports total input rather than uncached input directly, the equivalent quantity is $I_{i,m}^{\text{nc}} = I_{i,m}^{\text{total}} - R_{i,m} - W_{i,m}$, where R and W are cache-read and cache-write tokens. Thus, input and output tokens are added, not subtracted.

Token values in the tables are averaged over all evaluated trajectories, including both successful and unsuccessful runs. Let $\mathcal{T}$ denote the set of all evaluated trajectories for a given strategy and benchmark, and let $N = |\mathcal{T}|$. We report

$$\overline{T}^{\mathrm{nc}} = \frac{1}{N} \sum_{i \in \mathcal{T}} T_i^{\mathrm{nc}}. \tag{11}$$

For reference, a cache-inclusive processing total would instead be

$$T_i^{\mathrm{all}} = \sum_{m=1}^{M_i} \left(I_{i,m}^{\mathrm{nc}} + R_{i,m} + W_{i,m} + O_{i,m} \right). \tag{12}$$

This quantity is not reported in the main tables and should not be interpreted as monetary cost without applying provider-specific prices to each token category.

We additionally distinguish a compression-aware diagnostic from API accounting. If a trajectory undergoes K_i management events, let $C_{i,j}$ be the complete context immediately before event j, P_i^{final} and O_i^{final} be the final prompt and output, and S_i^{repeat} be fixed system-prompt content counted repeatedly across reconstructed segments. Then

$$T_i^{\mathrm{aware}} = \sum_{j=1}^{K_i} C_{i,j} + P_i^{\mathrm{final}} + O_i^{\mathrm{final}} - S_i^{\mathrm{repeat}}. \tag{13}$$

When no management event occurs, $T_i^{\mathrm{aware}} = P_i^{\mathrm{final}} + O_i^{\mathrm{final}}$. Unlike Equation 10, this diagnostic measures the total effective context represented across compression segments rather than the cumulative tokens processed or billed by the API. It is therefore useful for analyzing repeated compression but is not interchangeable with the non-cache metric reported in the main tables.

D Extended Analysis of Older Visual Context and Fidelity

RQ3 jointly studies whether older visual memory is retained and how densely it is rendered, using an additional 100-question BrowseComp-EN subset evaluated with avg@2 and disjoint from the 200-question main evaluation subset. The $0\times$ recent-context-only condition omits the older-history canvas, whereas the $1\times$ reference adds the older history at full canvas scale. Positive scale factors then vary the fidelity of the same dynamically rendered older-history canvas.

The visual-token effect of two-dimensional scaling can be approximated analytically. Let $V_{m,j}(1)$ be the number of visual tokens assigned to older-history canvas image j at model call m under full resolution. For an encoder that tokenizes approximately fixed-size image patches, scaling both dimensions by s gives

$$V_{m,j}(s) \approx s^2 V_{m,j}(1), \qquad V_{\mathrm{traj}}(s) \approx s^2 \sum_m \sum_j V_{m,j}(1). \tag{14}$$

At $s = 0.75$, this approximation gives $V_{\mathrm{traj}}(0.75) \approx 0.5625 V_{\mathrm{traj}}(1)$, or $V_{\mathrm{traj}}(1)/V_{\mathrm{traj}}(0.75) \approx 1.78$. The full canvas therefore contributes approximately 77.8% more visual tokens than the $0.75\times$ canvas over the same sequence of images and calls. This is a visual-token approximation rather than an exact provider accounting rule: adaptive tiling, minimum image resolutions, and discrete preprocessing boundaries can make the realized ratio non-quadratic. It also applies only to the visual component, not to text or output tokens in Equation 10.

As detailed in Table 5, the matched $0\times$ and $1\times$ comparison yields a modest 1.0-point accuracy gain together with lower token, API-call, and turn costs when older visual context is present. More history can therefore improve accuracy while reducing redundant search. Among the retained scale settings, performance is non-monotonic: $0.7\times$ and $0.8\times$ achieve the highest observed accuracy (71.0%), while $0.8\times$ reduces cumulative tokens to 1.03M and requires the fewest average API calls (56.5). Together with Equation 14, this pattern suggests that full-resolution history may add many visual tokens after task-relevant structure is already legible, thereby dispersing attention over redundant detail. This explanation is plausible but not causally identified by the current experiment. The main experiments use $0.75\times$, chosen as the midpoint of the strongest observed scale region rather than a separately measured optimum.

The compression factors in Table 5 compare text tokens with all visual tokens, including those associated with both the recent context H_t^r and the older history H_t^c. Let T_{text} denote the text-token

count and V_r and V_c the visual-token counts for these two components, measured over the same accounting scope. We define

$$\text{Vis. Ratio} = \frac{V_r + V_c}{T_{\text{text}}}, \qquad \text{Compression Factor} = \left\lceil \frac{T_{\text{text}}}{V_r + V_c} \right\rceil . \tag{15}$$

The reported factors, such as $16\times$ and $14\times$, therefore use the complete visual-token count rather than only the downsampled older-history canvas, with the reciprocal rounded upward to the next integer. In the $0\times$ condition, the older-history canvas is omitted while visual tokens from the recent context remain included.

Table 5: Performance and cost of the $0\times$ recent-context-only control and retained older-history canvas scales on the additional 100-question BrowseComp-EN subset (avg@2), disjoint from the 200-question main evaluation subset. The final column reports text tokens divided by all visual tokens from H_t^r and H_t^c, rounded upward as defined in Equation 15.

Canvas Scale	**Acc. (%)** ↑	**Tokens** ↓	**API Calls** ↓	**Turns** ↓	1/**Vis. Ratio**
$0\times$ (No Older History)	63.0	1.86M	99.8	218.4	$16\times$
$0.25\times$ Scale	65.0	1.62M	86.5	189.6	$14\times$
$0.5\times$ Scale	68.5	1.50M	81.0	**178.3**	$12\times$
$0.7\times$ Scale	**71.0**	1.15M	62.5	196.8	$10\times$
$0.8\times$ **(Best Observed)**	**71.0**	**1.03M**	**56.5**	179.8	$7\times$
$1\times$ (Full Canvas)	64.0	1.48M	79.3	181.7	$6\times$

E Additional Turn-Budget Efficiency Analysis

We construct a step-by-step exploration analysis using Visual Rendering. Figure 3 displays the cumulative task success rate as a function of the available turn budget. Visual Rendering accumulates successful completions more rapidly and sustains the highest curve across longer trajectories.

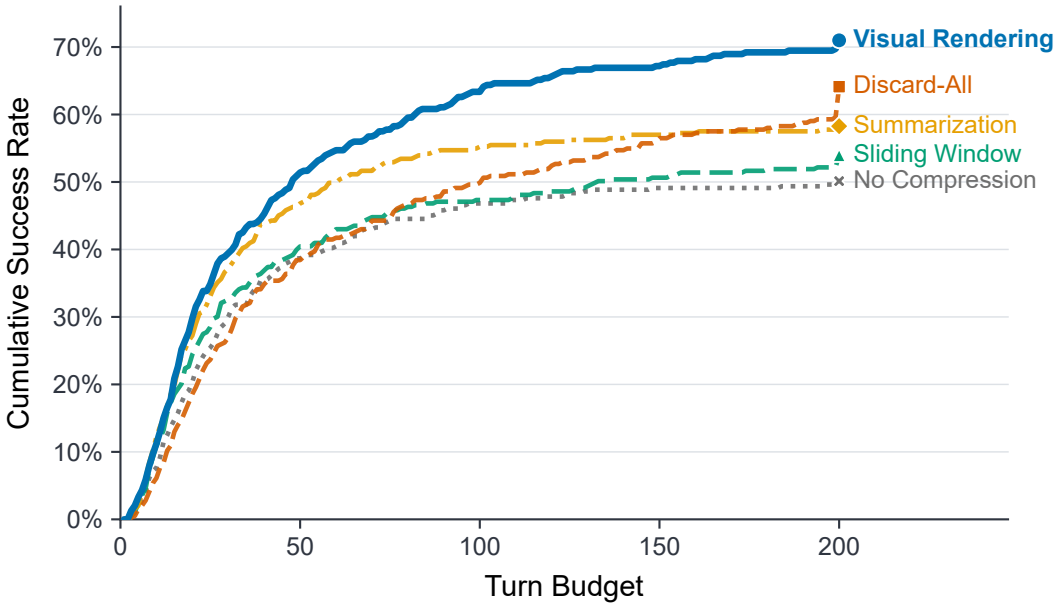


Figure 3: Cumulative task success rate on BrowseComp as a function of turn budget for Visual Rendering and the comparison strategies.

F PROMPTS

Visual Context Prompt

```
VISUAL_CONTEXT_PROMPT = (
    "The image(s) below contain a visual snapshot of your previous research
        history "
    "-- including search queries, tool responses, your reasoning, and
        intermediate "
    "findings from earlier rounds. This is your compressed memory of work already
         done.\n\n"
    "When continuing your investigation:\n"
    "- Reference specific facts, URLs, and numbers visible in the image(s) when
        relevant\n"
    "- Do not re-search for information already found in your history\n"
    "- Cite sources using the [^index^] format visible in the image(s)\n"
    "- If you need precise details from the history that are hard to read, verify
         with a targeted search"
)
```

Eval User Prompt Construction

```
def build_query(spec: RunSpec, enable_delegate: bool = False) -> str:
    base = (
        "You are running one browse-comparison evaluation task.\n"
        "Use only the tools made available in this Hermes session.\n"
        "Do not use or mention the reference answer. It is reserved for offline
            evaluation.\n"
        "If the task asks you to create a file, save it in the current working
            directory.\n\n"
    )
    delegate_prompt = (
        "STRATEGY:\n"
        "This task requires tracing a chain of connections. You should use the
            delegate_task "
        "tool to parallelize independent research subtasks. For example, if the
            question "
        "describes a multi-step chain (A->B->C->D->E), delegate parallel searches
            for "
        "different segments of the chain to subagents, then synthesize their
            findings.\n"
        "Each delegate receives its own context and tools -- use batch mode (the '
            tasks' "
        "array) to spawn multiple subagents simultaneously when segments are
            independent.\n\n"
    )
    task = (
        f"Task type: {spec.item.type}\n"
        f"{format_images_for_prompt(spec.item.image)}"
        f"Question:\n{spec.item.question}\n\n"
        "Return the final answer clearly."
    )
    return base + (delegate_prompt if enable_delegate else "") + task
```