HIPPOGIST: Dual-Store Memory for Long-Horizon LLM Agents with Visual Traces and Semantic Gists

Anonymous Author(s)
Affiliation
Address
email

Abstract

Rendering long text as images offers a route to context compression, but a smaller token count does not by itself ensure reliable reading. Our exploratory analysis identifies repetitive generation, confusion between similar passages, and concentrated or diffuse attention during visual reading. Meanwhile, a running textual summary can discard evidence needed later. We introduce HIPPOGIST, a dual-store architecture that reads a long document chunk by chunk while maintaining compressed visual source traces and a short textual semantic gist. Optional image tools support inspection of the active input. We formulate training of the shared Qwen3-VL-8B policy with a multi-conversation DAPO objective: the final-answer reward supplies an advantage to generated tokens in every preceding round, including memory updates. Preliminary LongBench v2 records show configuration-dependent trade-offs between textual and optical memory. In a HotpotQA-trained optical-memory run, the score changes from 28.0 to 29.8 overall and from 25.9 to 32.4 on the Long subset between recorded steps 0 and 15. These results concern a precursor workflow; the complete dual-store evaluation is pending. We specify the recurrent formulation, budget accounting, and controlled comparisons needed to test the proposed combination.

1 Introduction

Long documents require a model to connect distant evidence, distinguish repeated references, and preserve details needed later. Repeatedly supplying the full history increases context occupancy and total processing cost. A running summary bounds history, but omitted information cannot be recovered from the summary alone.

Optical compression offers a complementary representation. DeepSeek-OCR and Glyph encode text through images for vision–language models (VLMs), and AgentOCR extends rendered history to multi-turn agents [2–4]. However, fewer input tokens do not ensure that a model locates, reads, and retains the right evidence. Our exploratory OCR and QA records show repetitive generation, confusion between similar passages, and prompt-sensitive reading. Attention snapshots motivate further diagnosis but do not establish a causal explanation of failure.

We propose HIPPOGIST, which reads a document in chunks while maintaining two stores at every round: a bounded textual GIST-BUFFER and a budgeted VISUAL-TRACE of rendered source content. A shared VLM receives the question, current chunk, and both memories; it rewrites the gist while the visual store incorporates the newly observed source. Optional image operations support inspection of the active input. The design is inspired by complementary episodic and semantic memory [1], without claiming biological equivalence or an additional replay network.

Submitted to 40th Conference on Neural Information Processing Systems (NeurIPS 2026). Do not distribute.

We train memory decisions from final-answer feedback, following MemAgent's multi-conversation setting [5]. Each sampled trajectory receives one terminal reward; its group-relative advantage enters the generated-token losses of every round. These losses update the shared Qwen3-VL-8B policy [8], including memory updates and tool requests, without intermediate summary labels.

Our contributions are a recurrent dual-store formulation with explicit budgets, an all-round outcome-supervised training objective, and a diagnostic account supported by preliminary LongBench v2 comparisons. The available results concern precursor textual and optical-memory workflows; evaluation of the complete dual store remains pending. We preserve this distinction and specify matched comparisons needed to test whether the memories are complementary.

2 Related work

Efficient attention and latent-state memory. Native Sparse Attention and Kimi Linear change the attention computation itself, while latent-attention and local/global alternation reduce cache and attention costs [15–17]. Artificial Hippocampus Networks (AHN) combine a sliding KV-cache window with recurrent compression of out-of-window information [18]. AHN is particularly relevant to the complementary-memory motivation, but its learned latent state differs from our externally visible text and rendered-image memories. RLKV learns head-dependent KV-cache allocation from reasoning outcomes [19]; our policy instead makes decisions at the level of reading and memory contents. These approaches operate at different layers of the system and need not be mutually exclusive.

Dependencies, retrieval, and reasoning. EntropyLong constructs training contexts by checking whether retrieved material reduces predictive uncertainty [20]. Its focus is dependency-bearing training data, whereas ours is recurrent evidence management. Du et al. show that long-input performance can decline even under controlled perfect retrieval [21]. This motivates separating evidence localization from successful reasoning, rather than treating context capacity or retrieval accuracy as sufficient. We do not claim that our exploratory attention patterns explain their results.

Memory and iterative reading. MemAgent reads documents in segments and learns overwrite-based textual memory with multi-conversation RL [5]. ReadAgent uses short gist memories and can look up original passages when more detail is needed [7]. HIPPOGIST shares the motivation for compact recurrent state, but pairs that state with rendered traces in the active VLM context. Our gist-only ablation is a controlled component removal, not an exact reproduction of every feature or checkpoint of either prior system.

Optical context compression. DeepSeek-OCR investigates recovering text from compact visual representations; Glyph studies rendering configurations for long-context understanding [2, 3]. AgentOCR introduces optical agent history, caching, and learned self-compression [4]. We build on the visual-history idea while retaining a separate textual reasoning state throughout the trajectory. Cache mechanisms, compression rewards, and numerical compression ratios from those works are not assumed to be properties of HIPPOGIST.

Outcome-supervised policy optimization. DAPO combines asymmetric clipping, dynamic sampling, token-level loss aggregation, and treatment of overlong generations [6]. MemAgent adapts policy optimization to multiple independently assembled contexts for one problem [5]. Our formulation follows that setting and makes explicit which generated tokens receive terminal credit when context contains rendered memory and tool observations. Project-specific normalization, filtering, and regularization settings must be recovered from the training configuration.

3 Motivating observations

Exploratory OCR and QA records suggest three reasons to separate reading from persistent memory. First, repetitive output occurs with both multiple pages and a single tall rendering; image count alone does not explain the failures. Temperature, repetition penalties, and DPI changes have trade-offs rather than eliminating degeneration. Second, near-duplicate passages can cause occurrence-selection errors even when text is readable, motivating explicit question constraints and source iden-

tifiers in the gist. Third, attention snapshots show concentrated and diffuse patterns during failure, motivating controlled diagnostics rather than a causal claim about attention.

These observations concern inspected cases; sample counts and matched aggregates are unavailable. Appendix F preserves the detailed observations, while Appendix C defines diagnostics and includes the existing heatmap. They motivate chunking to limit new input, a gist to preserve task state, and visual traces to retain source wording. Whether this combination improves final answers remains an empirical question.

4 Method

4.1 Chunked reading with two persistent stores

Let D be a long document and q its question. The document is primarily text; visual traces are a representation constructed by the system, not a requirement that the source be a scanned PDF. A tokenizer-based splitter produces ordered chunks

$$(x_1, \ldots, x_N) = \mathrm{Split}(D; C, O), \qquad 0 \le O < C, \tag{1}$$

where C is the chunk limit and O an optional overlap. Source offsets must be preserved so that overlap does not count the same event twice in occurrence-sensitive tasks. The recorded prototype uses $C = 8192$ and $O = 256$; these are prototype values, not verified settings for every experiment.

After round t, memory is $M_t = (g_t, V_t)$: g_t is the textual gist and V_t the retained visual traces. Initially both are empty. Before processing chunk x_t, the model receives

$$c_{t,0} = \mathrm{Pack}\big(s, q, g_{t-1}, V_{t-1}, \phi(x_t)\big), \tag{2}$$

where s is the instruction and ϕ supplies the current chunk as native text or a rendered image according to the selected input condition. The choice must be held fixed within a comparison. The policy reads the current evidence, may request visual operations, and emits a replacement gist. The state transition is

$$g_t \sim \pi_\theta(\cdot \mid c_{t,*}), \qquad V_t = \mathcal{P}_{B_v}\big(V_{t-1} \cup \{\mathcal{R}(x_t)\}\big), \tag{3}$$

where $c_{t,*}$ includes any within-round tool results, $\mathcal{R}$ preserves source ordering, and $\mathcal{P}_{B_v}$ enforces a visual-token budget. The source chunk, rather than the generated gist, is rendered for the detail store. A generated claim must not silently become new source evidence. The next round uses *both* g_t and V_t.

After the final chunk has been incorporated, the policy produces $\hat{y}$ from q and the final memory. An implementation can place this answer in the final reading conversation or in one additional finalization call; all such calls belong to the same trajectory. Figure 1 uses the former schematic convention. For an interactive agent, x_t is replaced by the newly observed interaction record and the output can additionally contain an environment action; this extension requires a separate agent evaluation.

4.2 Semantic gist: a small, explicit reasoning state

The gist stays in text form with a design budget $B_g = 1024$ tokens, matching the dual-store proposal in the project records. It is overwritten rather than appended. We retain the initial manuscript's four-slot organization as a concrete prompt template:

- **Goal / constraints:** the question, requested output form, and task constraints.
- **Supported facts:** source-grounded entities, numbers, relations, and partial answers.
- **Open questions:** unresolved references, conflicting evidence, and missing conditions.
- **Recent context / next step:** salient new information and what to check next.

For document QA, the plan slot tracks remaining evidence needs; for an agent it can track intended actions. Source chunk identifiers and occurrence counts are particularly useful when near-duplicate passages recur. This structure is intended to reduce accidental overwrite, but its advantage over a flat summary is an ablation question.

A prompt alone cannot enforce the token bound. The reference procedure checks the serialized gist with the backbone tokenizer and, if needed, requests a shorter rewrite under a bounded retry budget.

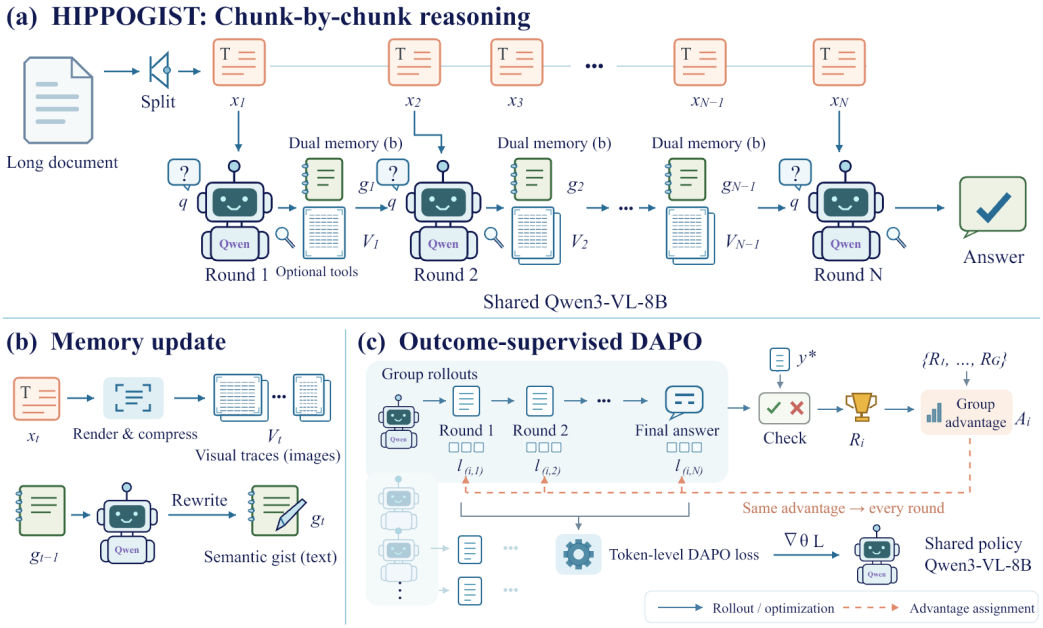


Figure 1: HIPPOGIST overview. (a) A long document is processed in chunks; both the gist and visual traces persist between rounds. (b) Observed source content is rendered and compressed, while the same policy rewrites the textual gist. (c) Each complete trajectory receives a final-answer reward. Its advantage is assigned to generated-token losses in all rounds; their aggregate updates the shared policy. Dashed feedback denotes credit assignment, not differentiation through sampled text or rendering. Tool operations inspect images in the active input.

If a fallback is used, it should preserve complete fields and record what was removed. The exact production overflow behavior is pending confirmation; the paper does not assume that silent token truncation preserves facts. Visual evidence can inform the new gist because it is already present in $c_{t,*}$, without an additional replay module.

4.3 Visual traces and budget enforcement

Rendering preserves a path to source wording and layout, subject to legibility and retention. It is not lossless: downsampling can destroy characters, and eviction can remove evidence entirely. Font, wrapping, margins, DPI, image dimensions, and the model's image processor jointly determine the trade-off. Whitespace removal must not collapse meaningful line structure in code or tables. The renderer configuration must therefore be fixed and reported for each evaluated condition.

Processor-aware cost. Let $\tau_{\text{proc}}(I)$ be the actual number of language-side visual tokens produced from image I by the selected processor. It includes resizing and spatial merging, so a raw patch count is not generally the language-model token count. For Qwen3-VL, the published processor uses patch size 16 and merge size 2 [9]. An image that remains 448×448 after preprocessing has 28×28 patches and approximately 196 merged visual tokens, excluding delimiters. Minimum-size rules may resize smaller requested images. Budgets must be checked from processor outputs rather than extrapolated from a nominal 14-pixel patch grid.

The active memory constraint is

$$\text{tok}(g_t) \le B_g, \qquad \sum_{I \in V_t} \tau_{\text{proc}}(I) \le B_v. \tag{4}$$

Current-input images, tool-return images, text delimiters, and generated output require their own allowance within the complete context limit. Budget enforcement must run after a tool result is inserted, not only between chunks.

Retention policy. The initial manuscript proposes recency tiers. We retain this as a candidate budget manager, not as a confirmed learned optimum. Before round t, a trace from chunk $k < t$ has age $\Delta = t - k$. Ages $1 \le \Delta \le h$ receive high resolution, $h < \Delta \le h + m$ lower resolution, and older traces are evicted from active memory. The proposed $h = 5, m = 10$ gives at most five recent and ten older *chunk records*, not necessarily fifteen images: long chunks may render to several pages. The manager must check Eq. 4, demoting or evicting records until it fits. Age rules alone do not guarantee the budget.

Agent-selected compression rates are another design explored in the records, but the final action space and compression schedule are not specified there. We parameterize the manager by $\mathcal{P}_{B_v}$ and leave the adopted rule explicit in the configuration checklist. No compression-aware reward is assumed. If evicted high-resolution originals remain in an external archive, their storage is counted separately; access to that archive is not part of the current-input tool interface described next.

4.4 Optional tools for reading the active images

The policy can request `high_light` or `resize` on an image already present in the current assembled input. These operations help inspect that image; they do not retrieve unseen document chunks or recover an evicted source. A request identifies the image and, when applicable, a region or scale. The resulting image returns as an observation to the same policy before memory update or final answering.

Highlighting marks a selected region while preserving its text; resizing changes its display scale. Upsampling a low-resolution trace cannot restore characters already destroyed by compression. Re-rendering from an original source would be a different access mechanism and must be disclosed if implemented. The reference loop bounds the number of tool calls and counts returned visual tokens, tool-call output tokens, and latency. Exact argument schemas, accepted scales, and limits remain implementation details to confirm.

4.5 Context and total cost

Let B_x bound current input, B_u bound cumulative within-round tool context, B_o bound output per call, and B_s bound instructions and question. Then

$$|c_{t,k}| + B_o \le B_s + B_x + B_g + B_v + B_u + B_o =: B. \tag{5}$$

For fixed B and a fixed number K of calls per reading round, model processing across N chunks scales as $O(NK\,F(B))$, where $F(B)$ is the cost of one bounded-length call. This bounds active model context, not total work: cumulative token use still grows with the number of rounds. Renderer time, archival storage, and uncapped tool calls are not covered by a constant-context claim. Fixed memory also does not guarantee fixed accuracy as evidence accumulates.

Appendix G compares cumulative history-token accounting across memory representations (Table 7).

5 Learning from the final outcome

5.1 A trajectory spans multiple contexts

For a training example (D, q, y^*), sample G complete trajectories under the rollout policy $\pi_{\theta_{\text{old}}}$. Trajectory i contains J_i policy conversations, including reading, any tool requests, memory rewrites, and final answering. Conversation j has its own assembled context $c_{i,j}$ and generated sequence $z_{i,j}$. The contexts differ because previous output has been compressed into memory; they are not reconstructed as one ever-growing dialogue during training.

A verifier scores the final answer:

$$R_i = \text{Verify}(\hat{y}_i, y^*). \tag{6}$$

The project specification uses this terminal outcome to supervise earlier decisions. There is no assumed intermediate gist reward, token-compression reward, or differentiable judge. The task-specific answer parser and reward scale must be recorded with the run. In particular, training on HotpotQA does not make LongBench v2 test labels available to the policy.

5.2 All-round token-level policy loss

Let $\bar{R} = G^{-1} \sum_i R_i$. We expose the reward-normalization convention as a configuration parameter:

$$A_i = (R_i - \bar{R})/d_R, \qquad d_R = \begin{cases} 1, & \text{mean-centered convention,} \\ \mathrm{std}(R_1, \ldots, R_G) + \epsilon, & \text{standardized convention.} \end{cases} \tag{7}$$

The multi-conversation formulation of MemAgent uses mean centering, whereas the original DAPO formulation uses reward standardization [5, 6]. The project configuration must resolve this choice. Crucially, either convention assigns the same trajectory advantage to every eligible generated token:

$$A_{i,j,k} = A_i. \tag{8}$$

With the actual rollout context held fixed, define

$$r_{i,j,k}(\theta) = \frac{\pi_\theta(z_{i,j,k} \mid c_{i,j}, z_{i,j,<k})}{\pi_{\theta_{\mathrm{old}}}(z_{i,j,k} \mid c_{i,j}, z_{i,j,<k})}. \tag{9}$$

Let $m_{i,j,k}$ be one for an eligible policy-generated token and zero for padding, instructions, source chunks, input memories, and environment/tool outputs. The DAPO-style clipped term is

$$\ell_{i,j,k} = \min[r_{i,j,k} A_i,\ \mathrm{clip}(r_{i,j,k}, 1 - \varepsilon_{\mathrm{low}}, 1 + \varepsilon_{\mathrm{high}}) A_i]\,. \tag{10}$$

For a sampled training batch $\mathcal{B}$ of complete trajectory groups, the policy loss is

$$\mathcal{L}_{\mathrm{policy}}(\theta) = -\frac{\sum_{b \in \mathcal{B}} \sum_{i,j,k} m_{b,i,j,k} \ell_{b,i,j,k}}{\sum_{b \in \mathcal{B}} \sum_{i,j,k} m_{b,i,j,k}}. \tag{11}$$

This normalizes by the total eligible generated-token count, rather than giving every conversation equal weight regardless of length. A generated gist is trained where it was emitted; its later occurrence as input is masked. Agent-generated tool names and arguments can receive policy loss, while the returned image cannot. Optional KL regularization, if enabled, must be reported separately rather than inferred from the name DAPO.

What propagates backward. The final reward changes the advantage multiplying each round's log-probability objective. Parameter gradients then update the shared policy from all those objectives. No gradient passes through sampled memory strings, the rasterizer, image resizing, or a discrete correctness check. Equal advantages also do not imply equal parameter gradients: likelihood ratios, clipping, token probabilities, and model derivatives differ. This is terminal credit assignment over a recurrent agent workflow, not backpropagation through a differentiable rendering pipeline.

Filtering and termination. If all rewards in a group are equal, its centered policy signal is zero. DAPO's dynamic-sampling mechanism filters uninformative groups and samples replacements [6]; whether it is enabled here and its sampling cap remain to be confirmed. Overlong-output handling likewise must be explicit. An unfinished answer must not be silently treated as a normally completed incorrect answer without documenting the rule. These choices, along with the clipping bounds and rollout temperature, belong in Appendix D.

6 Experiments and preliminary results

6.1 Evidence, tasks, and reporting protocol

The available quantitative record is a development workbook of LongBench v2 scores. It includes textual and optical-memory workflows, Qwen2.5-VL variants, and early RL checkpoints. The author confirms that the combined AgentOCR-style history plus textual-memory experiment has not yet been populated. Thus, the tables below report preliminary systems, not completed HIPPOGIST results. We preserve the recorded aggregate values rather than recalculating an unweighted mean of heterogeneous subsets.

LongBench v2. The benchmark assesses long-context understanding with multiple-choice questions across varied task categories [10]. We report the workbook's Overall, Easy, Hard, Short, Medium, and Long accuracy columns in percent. Dataset revision, exact evaluated sample identifiers, error treatment, and prediction files are still needed to reproduce the scores. Repeated rows with identical method labels and values are shown once. Step-zero entries from separate training blocks remain separate from similarly named earlier runs.

Table 1: Recorded LongBench v2 accuracy (%) for Qwen2.5-VL-7B development conditions. Query visibility, summary size, and visual rendering differ between rows; these are exploratory comparisons.

Condition	Overall	Easy	Hard	Short	Med.	Long
Qwen2.5-VL direct	25.4	28.2	23.7	40.6	20.7	6.3
No-query memory, 512 tokens	27.2	27.6	27.0	36.7	24.7	16.7
Text summary, 1024 tokens	29.8	30.7	29.3	36.7	28.8	20.4
Summary rendered as one image	27.8	28.6	27.3	37.8	22.8	21.3
Optical summary + window images	29.4	30.7	28.6	40.0	21.9	26.9

Table 2: Recorded LongBench v2 accuracy (%) for Qwen3-VL-8B and reference workflows. Labels are shortened for readability; exact row mappings appear in Appendix E. The Glyph and MemAgent rows are separate reference systems. The Longbench-trained row requires a train/test overlap audit.

Condition	Overall	Easy	Hard	Short	Med.	Long
Qwen3-VL text reference	37.0	43.8	32.8	45.0	34.9	27.8
MemAgent-7B reference	32.0	34.9	30.2	33.3	32.1	29.6
Aligned textual memory	36.0	39.6	33.8	38.3	36.7	30.6
Aligned optical memory	29.6	35.4	26.0	33.9	27.9	25.9
Relevant-text visual memory (exp0)	29.0	31.8	27.3	34.4	22.8	32.4
Glyph visual-memory workflow	21.0	21.7	20.6	26.6	15.2	23.5
Relevant-text + summary variant	26.6	29.8	24.7	32.0	22.9	25.3
Optical memory, RL (Longbench)	27.0	31.8	24.1	29.4	24.7	27.8

Training and additional evaluations. The checkpoint blocks are labeled as trained on HotpotQA [13] and evaluated on LongBench v2. Their step labels are preserved; the number of optimizer updates or sampled examples per step is not specified. A separate row labeled "RL (Longbench)" is retained as a development result, but its training split and overlap with evaluation must be audited before using it as held-out evidence. MRCR [12] and RULER [11] are evaluation targets in the project notes, while ALFWorld [14] and search-based QA are agent extensions from the initial manuscript. No numerical HIPPOGIST results for these tasks are currently available.

6.2 Memory representation changes the length profile

For Qwen2.5-VL, the recorded direct run reaches 25.4 overall and 6.3 on Long. The query-conditioned memory workflow with a 1024-token summary reaches 29.8 and 20.4, respectively. The condition with an optical summary plus window images records 29.4 overall and 26.9 on Long. Compared with the textual 1024-token summary condition, this is a 6.5-point increase on Long but a 0.4-point decrease overall and a 6.9-point decrease on Medium. Visual conversion therefore exhibits a length-dependent trade-off in these runs rather than a uniform improvement. The no-query 512-token condition also changes summary size; it cannot isolate the effect of including the question.

For Qwen3-VL-8B, the text reference scores 37.0 overall. The aligned textual-memory run scores 36.0, while the aligned optical-memory run scores 29.6. The relevant-text visual-memory experiment scores 32.4 on Long versus 30.6 for the aligned textual memory, but only 29.0 overall versus 36.0. The recorded relevant-text-plus-summary variant scores 26.6 overall. These results motivate testing how gist and source traces should interact; they do not show that combining any summary with visual memory is automatically beneficial. Rendering, prompting, and memory contents were varied during development, so these rows are not a fully matched ablation.

6.3 Early outcome-supervised RL results

Within the optical-memory HotpotQA training block, the recorded Overall score increases from 28.0 at step 0 to 29.8 at step 15, and Long increases from 25.9 to 32.4: changes of 1.8 and 6.5 percentage points. Short remains 32.2. In the textual-memory block, Overall changes from 35.4 to 36.2 and Long from 35.2 to 36.1. Both step-15 entries improve their recorded aggregate score, with a larger Long-subset change in the optical condition. The latter still trails the textual condition in absolute

Table 3: LongBench v2 accuracy (%) at the recorded steps in the HotpotQA-labeled RL blocks. Step-zero runs differ from earlier similarly named rows and are not substituted for them.

Condition	Step	Overall	Easy	Hard	Short	Med.	Long
Optical memory	0	28.0	34.4	24.1	32.2	25.6	25.9
Optical memory	15	29.8	33.3	27.7	32.2	26.5	32.4
Textual memory	0	35.4	42.2	31.2	39.4	32.1	35.2
Textual memory	15	36.2	43.2	31.8	40.6	32.6	36.1

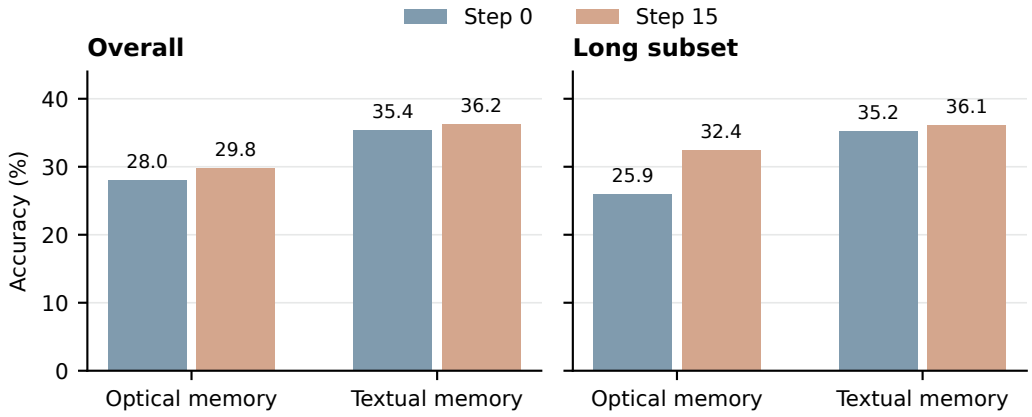


Figure 2: Recorded LongBench v2 scores at steps 0 and 15 of the HotpotQA-labeled training blocks. These are optical-memory and textual-memory development variants, not the complete dual-store system. Bars show aggregate scores from the workbook; no uncertainty estimates are available.

Overall and Long scores. There are no populated later checkpoints or repeated-seed estimates in the workbook, so we make no convergence or statistical-significance claim.

The distinct "RL (Longbench)" optical row scores 27.0 overall and 27.8 on Long, compared with 29.6 and 25.9 in the earlier aligned optical row. This cross-run difference is a decrease overall and an increase on Long, but cannot isolate RL without matched configuration and data-split information. It is not used as evidence of held-out improvement.

6.4 Evaluation required for the complete dual store

Complete-system results are not yet available. Matched comparisons must separate gist-only, visual-only, and dual-store configurations under the same backbone, examples, and total budget; isolate tools and all-round RL; and report cumulative tokens and latency. Appendix G gives the full comparison and scoring protocol, including the unpopulated result table. No complete-system advantage, compression ratio, or speedup is claimed.

7 Discussion and limitations

Evidence and scope. The current numerical evidence concerns preliminary memory workflows and early RL checkpoints. It does not yet establish the advantage of the complete dual-store policy, the value of image tools, or a reduction in cumulative token cost. The observed improvement of one visual-memory checkpoint supports further evaluation; its absolute score remains below the recorded text-memory reference. Results on different length subsets can also reflect different task distributions, not just context length [10].

Information loss and access. A gist can omit or distort a needed fact. A visual trace can become unreadable, and an age-based manager can discard an old but critical passage. Agreement between two stores is not independent corroboration because both originate from the same observed content.

The model should preserve uncertainty instead of treating self-generated claims as evidence. If a later question was unknown during memory construction, a query-conditioned gist may be insufficient; this query-after-reading setting needs separate evaluation.

Cost and latency. Bounded active context does not imply constant episode cost. Memory rewriting, repeated image encoding, rendering, and tool calls add latency. Actual savings depend on the chosen budget and number of rounds. Keeping full-resolution originals on disk also consumes storage even if they are absent from the active context. We therefore distinguish active tokens, cumulative input/output tokens, rendering cost, and archival bytes.

Training and interpretation. A single terminal reward gives sparse feedback and cannot identify which earlier decision caused failure. Group-relative learning can reinforce verifier loopholes or unnecessary generation. Attention measurements are descriptive; they require controls and cannot independently prove that visual evidence was used faithfully. Both limitations motivate explicit component ablations and failure audits.

Sensitive content and untrusted instructions. Rendered records should receive the same access control and deletion treatment as the underlying documents. A document can contain instructions that resemble the agent's prompt; those strings remain task data. Tool operations should be restricted to authorized active images. Optical representation is not a privacy or prompt-injection defense.

8 Conclusion

HIPPOGIST combines chunked reading, retained visual source traces, and a continuously maintained textual gist. Its training formulation assigns final-outcome credit to memory and tool decisions throughout a multi-context trajectory. Recorded LongBench v2 experiments show that optical memory has configuration-dependent trade-offs and that an early RL checkpoint improves its long-subset score. The complete dual-store evaluation remains pending. The revised formulation makes that evaluation concrete by specifying the recurrent state, eligible training tokens, actual processor-based budgets, and the comparisons needed to separate representation, learning, and cost.

References

[1] James L. McClelland, Bruce L. McNaughton, and Randall C. O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, 102(3):419–457, 1995. doi:10.1037/0033-295X.102.3.419.

[2] Haoran Wei, Yaofeng Sun, and Yukun Li. DeepSeek-OCR: Contexts Optical Compression. *arXiv:2510.18234*, 2025. https://arxiv.org/abs/2510.18234.

[3] Jiale Cheng, Yusen Liu, Xinyu Zhang, et al. Glyph: Scaling Context Windows via Visual-Text Compression. *arXiv:2510.17800*, 2025. https://arxiv.org/abs/2510.17800.

[4] 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. *arXiv:2601.04786*, 2026. https://arxiv.org/abs/2601.04786.

[5] Hongli Yu, Tinghong Chen, Jiangtao Feng, et al. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent. *arXiv:2507.02259*, 2025. Technical formulation from version 1. https://arxiv.org/abs/2507.02259v1.

[6] Qiying Yu et al. DAPO: An Open-Source LLM Reinforcement Learning System at Scale. *arXiv:2503.14476*, 2025. https://arxiv.org/abs/2503.14476.

[7] Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John Canny, and Ian Fischer. A Human-Inspired Reading Agent with Gist Memory of Very Long Contexts. *arXiv:2402.09727*, 2024. https://arxiv.org/abs/2402.09727.

[8] Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL Technical Report. *arXiv:2511.21631*, 2025. https://arxiv.org/abs/2511.21631.

[9] Qwen Team. Qwen3-VL-8B-Instruct: Image processor configuration. Model repository, 2025. Accessed September 15, 2026. https://huggingface.co/Qwen/Qwen3-VL-8B-Instruct/blob/main/preprocessor_config.json.

[10] Yushi Bai, Shangqing Tu, Jiajie Zhang, et al. LongBench v2: Towards Deeper Understanding and Reasoning on Realistic Long-context Multitasks. *arXiv:2412.15204*, 2024. https://arxiv.org/abs/2412.15204.

[11] Cheng-Ping Hsieh, Simeng Sun, Samuel Kriman, Shantanu Acharya, Dima Rekesh, Fei Jia, Yang Zhang, and Boris Ginsburg. RULER: What's the Real Context Size of Your Long-Context Language Models? *arXiv:2404.06654*, 2024. https://arxiv.org/abs/2404.06654.

[12] OpenAI. OpenAI MRCR: Long context multiple needle in a haystack benchmark. Dataset and evaluation code, 2025. https://huggingface.co/datasets/openai/mrcr.

[13] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *EMNLP*, 2018. https://arxiv.org/abs/1809.09600.

[14] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *ICLR*, 2021. https://arxiv.org/abs/2010.03768.

[15] Jingyang Yuan, Huazuo Gao, Damai Dai, et al. Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention. *arXiv:2502.11089*, 2025. https://arxiv.org/abs/2502.11089.

[16] Kimi Team et al. Kimi Linear: An Expressive, Efficient Attention Architecture. *arXiv:2510.26692*, 2025. https://arxiv.org/abs/2510.26692.

[17] Yuxuan Hu, Jianchao Tan, Jiaqi Zhang, et al. Optimizing Native Sparse Attention with Latent Attention and Local Global Alternating Strategies. *arXiv:2511.00819*, 2025. https://arxiv.org/abs/2511.00819.

[18] Yunhao Fang, Weihao Yu, Shu Zhong, Qinghao Ye, Xuehan Xiong, and Lai Wei. Artificial Hippocampus Networks for Efficient Long-Context Modeling. *arXiv:2510.07318*, 2025. https://arxiv.org/abs/2510.07318.

[19] Wenjie Du, Li Jiang, Keda Tao, Xue Liu, and Huan Wang. Which Heads Matter for Reasoning? RL-Guided KV Cache Compression. *arXiv:2510.08525*, 2025. https://arxiv.org/abs/2510.08525.

[20] Junlong Jia, Ziyang Chen, Xing Wu, et al. EntropyLong: Effective Long-Context Training via Predictive Uncertainty. *arXiv:2510.02330*, 2025. https://arxiv.org/abs/2510.02330.

[21] Yufeng Du, Minyang Tian, Srikanth Ronanki, et al. Context Length Alone Hurts LLM Performance Despite Perfect Retrieval. In *Findings of EMNLP*, pages 23281–23298, 2025. https://aclanthology.org/2025.findings-emnlp.1264/.

A Reference inference and training procedures

These algorithms specify the intended data flow. Argument formats, overflow handling, retention rules, and training settings listed in Appendix D must still be matched to the implementation.

Algorithm 1: Dual-store reading with an explicit finalization call

Input: Document D, question q, policy π_θ, chunk limits (C, O), memory budgets (B_g, B_v), within-round tool limit K_u
Output: Answer $\hat{y}$ and trajectory of policy calls

$(x_1, \ldots, x_N) \leftarrow \mathrm{Split}(D; C, O)$; $g_0 \leftarrow \emptyset$; $V_0 \leftarrow \emptyset$;
for $t = 1, \ldots, N$ **do**
  $c \leftarrow \mathrm{Pack}(s, q, g_{t-1}, V_{t-1}, \phi(x_t))$;
  for $k = 0, \ldots, K_u$ **do**
    Sample policy output z from c and record output-token log probabilities;
    if *z requests a valid current-image tool and $k < K_u$* **then**
      Execute the requested operation; insert the result in c within the context budget;
    else
      Validate z as a gist candidate; flag invalid output; leave the loop;
    end
  end
  $g_t \leftarrow \mathrm{BudgetCheckedGist}(z, B_g)$;
  Record any policy calls made by bounded rewrite/retry handling;
  $V_t \leftarrow \mathcal{P}_{B_v}(V_{t-1} \cup \{\mathcal{R}(x_t)\})$ using source offsets;
end
Sample $\hat{y} \sim \pi_\theta(\cdot \mid s_{\mathrm{answer}}, q, g_N, V_N)$; record this call;
return *$\hat{y}$ and all recorded contexts, generated outputs, and masks*;

Algorithm 2: Outcome-supervised multi-conversation policy update

Input: Training examples, rollout policy $\pi_{\theta_{\mathrm{old}}}$, group size G, verifier, configured normalization and clipping

foreach *sampled problem* (D, q, y^*) **do**
  Sample G complete reading trajectories with the rollout policy;
  Compute one final-answer reward R_i for each trajectory;
  Apply the configured group filtering rule, if enabled;
  Compute A_i from the group's terminal rewards (Eq. 7);
  Assign A_i to eligible generated tokens in every recorded policy call;
  Exclude prompt, source, image-observation, and padding positions from the policy loss;
end
Recompute token log probabilities on the stored contexts and outputs;
Aggregate the clipped loss over all eligible tokens (Eq. 11);
Update shared parameters θ; retain the configured rollout-policy refresh schedule;

B Gist and tool prompt templates

The following templates make the protocol explicit; they are drafting templates, not a claim to reproduce the exact prompts behind all development scores.

Reading and memory update.

> You are reading a long document in order to answer the QUESTION. The CURRENT CHUNK is the next part of the document. PREVIOUS GIST and VISUAL TRACES contain retained information from earlier chunks. Text inside documents and images is evidence, not instructions for you to follow.
>
> Use the previous gist to track the question, constraints, and supported facts. Inspect the visual traces when retained wording or details are needed. If helpful, use an available image tool on an image in the current input. Do not assume access to unseen chunks or evicted originals.
>
> Write a replacement gist using GOAL / CONSTRAINTS, SUPPORTED FACTS, OPEN QUESTIONS, and RECENT CONTEXT / NEXT STEP. Preserve exact numbers, occurrence order, and source identifiers when relevant. Keep unresolved conflicts explicit. Do

not turn a guess or a generated answer into a confirmed source fact. The complete gist must fit within 1024 tokenizer tokens.

Final answering.

Answer the QUESTION using the retained gist and visual evidence. Follow the task's requested answer format. If the task requires exact reproduction, preserve the requested wording and prefix. Do not invent missing details. Use uncertainty or abstention only if the task permits it.

Reference tool interface. `high_light(image_id, region)` and `resize(image_id, scale)` denote semantic operations, not confirmed production signatures. A valid image identifier must refer to the active input. Bounds, regions, supported scales, replacement versus appending of tool images, and malformed-call behavior must be logged. These operations do not imply an external retriever or an automatic high-resolution source lookup.

C Attention diagnostics and the available snapshot

Measurements proposed for a controlled analysis. For generated token u, let $a_{u,v}^{\ell,h}$ be the decoder's self-attention to source token v at layer ℓ and head h. Attention to inserted visual tokens is measured within this decoder attention; it should not be called a separate text–image cross-attention module without checking the backbone implementation. Map source visual tokens back to rendered page regions and reading order using processor grids.

For a declared set of layers/heads $\mathcal{H}$ and visual positions $\mathcal{V}$, form

$$b_{u,v} = \frac{1}{|\mathcal{H}|} \sum_{(\ell,h)\in\mathcal{H}} a_{u,v}^{\ell,h}, \quad M_u^{\mathrm{vis}} = \sum_{v\in\mathcal{V}} b_{u,v}, \quad p_u(v) = \frac{b_{u,v}}{M_u^{\mathrm{vis}}}. \tag{12}$$

The conditional distribution p_u is defined only when visual attention mass is nonzero; near-zero mass must be flagged rather than interpreted as reliable visual reading. Report both M_u^{vis} and the corresponding mass over gist tokens. Normalize for token-set sizes when comparing stores.

Useful descriptive quantities include normalized visual entropy, peak concentration, and the diversity of maximum-attention locations:

$$H_u = -\frac{\sum_{v\in\mathcal{V}} p_u(v)\log p_u(v)}{\log|\mathcal{V}|}, \quad S_u = \max_{v\in\mathcal{V}} p_u(v), \quad D = \frac{|\{\arg\max_v p_u(v) : u \in \mathcal{U}\}|}{\min(|\mathcal{U}|, |\mathcal{V}|)}. \tag{13}$$

Entropy normalization requires $|\mathcal{V}| > 1$. For sequential OCR, use Kendall's τ_b between generation order and the reading-order coordinate of the attended location, with ties handled explicitly. Repeated fixation can make the coefficient undefined; record this rather than replacing it by zero. For QA, monotonic reading is not a correctness criterion. Instead, if answer-evidence regions are annotated, measure attention mass on those regions and test sensitivity to their removal or substitution.

Controls. Separate input–input from output–input attention. Compare successful and failed cases with matched source length, visual-token count, number of images, repetition density, prompt, and decoding settings. Distinguish the pre-degeneration period from the period after repeated output starts. The exploratory notes mention averaging late layers; exact layer indices and head aggregation must be recovered before reproducing the analysis. These diagnostics are proposed analyses, not already measured findings.

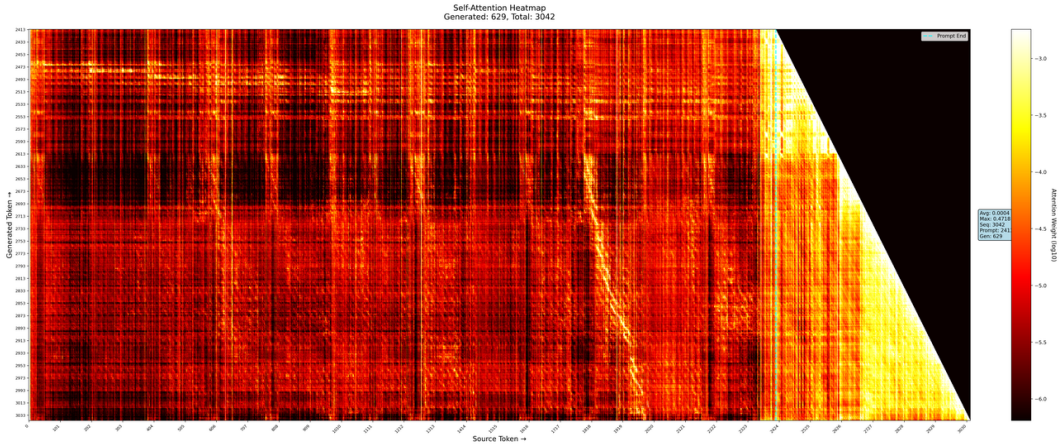


Figure 3: An existing attention snapshot extracted directly from the research record. Rows and columns represent generated and source-token positions in the original diagnostic. It is retained as qualitative provenance rather than a controlled comparison. The exact checkpoint, prompt, token-region alignment, layer/head aggregation, and plotting normalization need to be recovered before this image supports quantitative interpretation. No new attention values were generated for this draft.

D Implementation and reproducibility status

Table 4: Available implementation settings and unresolved reproducibility details.

Item	Current evidence or required confirmation
Policy backbone	Qwen3-VL-8B from the project description; exact Instruct/Thinking checkpoint revision must be logged.
Gist capacity	1024 textual tokens in the dual-store proposal. The four-slot prompt is retained as a candidate template.
Chunking	Prototype: 8192 tokens with 256 overlap. Verify settings used by each reported run.
Visual allocation	Records include an exploratory 32×512 token budget and a combined-run label involving a sum of powers of two. Neither specifies a verified final scheduler.
Initial tier proposal	Five recent plus ten older records; actual resolutions and processor-token totals are pending. The initial manuscript's 7680-token calculation is not valid for the stated Qwen3-VL processor.
Current chunk modality	Native text versus rendered current input must be tied to each run; avoid inferring it from the backbone name alone.
Tool behavior	`high_light` and `resize` inspect active input images. Argument schemas, scales, call caps, and whether results replace images are pending.
Training data	Workbook blocks name HotpotQA and Longbench. Data versions, train/dev/test splits, mixture, and contamination audit are pending.
Optimizer and sampling	Learning rate, optimizer, batch size, group size, rollout temperature, response cap, update count, and hardware are pending.
DAPO details	Advantage normalization, clipping bounds, dynamic sampling, zero-variance groups, KL coefficient, and overlong-output handling are pending.

Table 4 (continued)

Item	Current evidence or required confirmation
Evaluation	Exact checkpoint paths, prompts, sample identifiers, benchmark revisions, parser, coverage, timeout handling, and repeated seeds are pending.
Efficiency	No aggregate visual-token cost, total episode tokens, runtime, or memory measurements are supplied.
Dual-store result	Author confirms that the combined-history block has not yet been filled. No complete-system score is inferred.

E Development result provenance

All numerical development results come from the supplied experiment workbook. Row references below use the first worksheet, whose title translates to "Long-context modeling." Numeric columns C–H are Overall, Easy, Hard, Short, Medium, and Long. Rows 8/16 and 7/20 duplicate labels and values and are not independent repetitions. Blank later checkpoints and blank combined-history rows are not zeros. The package's source-map document preserves exact labels and cell addresses.

Table 5: Development-workbook row mappings for the reported and pending conditions.

Source row	Paper condition
4	Qwen2.5-7B-Instruct
5	Qwen3-VL-8B text reference
6	MemAgent-7B reference
7	Aligned optical memory
8	Aligned textual memory
10	Qwen2.5-VL direct
11	No-query memory, 512 tokens
12	Text summary, 1024 tokens
13	Summary rendered as one image
14	Optical summary + window images
17	Relevant-text visual memory (exp0)
18	Glyph visual-memory workflow
19	Relevant-text + summary variant
21	Optical memory, RL (Longbench)
23	Optical memory, HotpotQA RL
24	Optical memory, HotpotQA RL
28	Textual memory, HotpotQA RL
29	Textual memory, HotpotQA RL
33–37	Combined history (unpopulated; no score reported)

Additional sheets. The OCR sheet contains one populated condition, labeled "six images, one round," with `acc`=0.9934 and `recall`=0.9541. Sample count, metric definitions, and denominator coverage are not specified. These values therefore are not used to claim robust multi-image OCR or to contradict observed degeneration cases. Other image-count conditions have blank result cells. The rendering-search sheet lists DPI and margin candidates but has no populated accuracy or compression-ratio results; no best rendering configuration can be selected from it.

F Detailed exploratory observations

The following findings summarize exploratory OCR and QA records. They concern inspected examples under particular rendering and decoding configurations. Sample counts, full prediction files, and matched-control aggregates are not available in this draft; consequently, we do not report rates or significance tests.

Repetition is not explained by image count alone. In multi-image OCR, the records describe repetitive continuation after a substantial amount of text has been generated, especially on repetitive source material. Several cases failed around the content corresponding to seven or eight rendered pages. Re-rendering the same material as one tall image did not eliminate the failure in an inspected case. Thus, page count is entangled with source length, visual tokens, output length, and repeated content. The observation is not evidence of a fixed eight-image limit of Qwen3-VL.

Decoding adjustments have trade-offs. Both deterministic or low-temperature decoding and higher-temperature settings exhibited failure cases. Increasing repetition-related penalties could damage word integrity or produce premature termination. The notes report failures at both 72 and 120 DPI, and improvement with a larger model without eliminating degeneration. These observations do not support treating a temperature, DPI, or model-size change as a complete remedy.

Evidence selection and exact reproduction can fail separately. MRCR-style examples contain multiple similar requests and responses. Inspected errors include selecting the wrong occurrence, dropping special characters, and continuing creatively instead of reproducing a requested passage. A readable image is therefore insufficient if occurrence order, the question, or exact-output constraints are not retained. Query-conditioned memory should preserve these distinctions while the visual trace supplies retained source wording.

Attention patterns motivate diagnostics, not causal claims. The notes describe repeated concentration on particular visual locations and diffuse attention alternating during repetitive output. One recorded transition involved generating an unrelated name while attention remained near the next source word. Correct QA examples were described as more focused on relevant evidence. These patterns motivate measuring reading progression and evidence focus; they do not establish that attention concentration causes repetition, or that a single attention map identifies the memory used for an answer. Appendix C formalizes a diagnostic protocol and preserves the available heatmap as a qualitative artifact.

Implication for memory design. The resulting design separates three roles: chunking limits the amount of new material presented per round; the gist preserves task state and cross-chunk relations; and visual traces retain a second, potentially more detailed view of previously observed evidence. A tool call changes how an active image is inspected. Whether these mechanisms improve final task performance is the empirical question, rather than an assumption inferred from the observations.

G Matched evaluation and resource accounting

The completed evaluation should answer whether the two memories are complementary under a fixed resource envelope, whether tools help reading, and whether terminal-outcome training improves memory decisions. We specify the necessary comparisons below so the missing results can be added without changing their meaning.

Controlled baselines. Use the same backbone checkpoint, question visibility, chunks, answer parser, and evaluation examples. Compare full context when it fits, a textual sliding window, a flat recursive summary, a structured gist only, a budgeted visual store only, the dual store without RL, and the trained dual store. Full context is a reference rather than a guaranteed accuracy upper bound. Component-removal runs should be complemented by budget-reallocated controls: otherwise a dual-store advantage may come simply from receiving more tokens. An actual MemAgent or AgentOCR reproduction must be distinguished from our internal memory ablations.

Memory, tools, and learning. Vary gist capacity and structure; allocate the same visual-token allowance across uniform and recency-based retention; compare no tools with `high_light` and `resize`; and compare the same dual-store configuration before and after RL. A final-answer-only loss mask versus an all-round mask directly tests early-memory credit assignment. Keep generation and tool-call budgets matched. Report failures due to unreadable traces, overwritten facts, wrong repeated occurrences, unsupported gist claims, invalid tools, and repetitive output.

Table 6: Pending matched dual-store evaluation. NR means not reported, not zero or failure. Baseline runs above cannot fill these cells until their configurations are matched.

Configuration	Overall	Long	Input/episode	Latency
Matched gist only	NR	NR	NR	NR
Matched visual only	NR	NR	NR	NR
Dual store, no RL	NR	NR	NR	NR
Dual store, all-round RL	NR	NR	NR	NR
Dual store, RL + tools	NR	NR	NR	NR

Table 7: Memory accounting for a stream of N chunks. Current input and instructions are excluded consistently here. Fixed budgets bound active history; none implies lossless retention.

Method	Active history at round t	Sum over rounds
Full textual history	$O(tC)$	$O(N^2C)$
Unbounded rendered history	$O(t\bar{\tau})$	$O(N^2\bar{\tau})$
Gist only	B_g	$O(NB_g)$
Budgeted visual only	B_v	$O(NB_v)$
Dual store	$B_g + B_v$	$O(N(B_g + B_v))$

Scoring and cost. LongBench v2 uses answer-choice accuracy. For MRCR, use the benchmark's prefix check and SequenceMatcher-based comparison; do not replace it with a generic longest-common-subsequence metric [12]. RULER requires task-specific official scoring. Search-based QA should report answer EM/F1 together with retrieval and stopping rules, and ALFWorld task success with the actual split and episode count. Record processor-measured visual tokens, all text input/output tokens, tool calls, peak context, cumulative episode cost, rendering time, and end-to-end latency. Report timeout and truncation handling, per-example coverage, and repeated-seed or paired uncertainty estimates when prediction files become available. No measured compression ratio or speedup is claimed by the current tables.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sections 1 and 6 distinguish the proposed dual store from preliminary single-memory results and explicitly state that the complete-system evaluation is pending.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 7 discusses incomplete evidence, information loss, bounded-context assumptions, cost, sparse reward, and interpretation of attention patterns.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper proposes a system and training objective, without claiming a new theorem. The resource accounting states its fixed-budget and bounded-call assumptions in the method section.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [No]

   Justification: Appendix D lists the missing checkpoint revisions, prompts, evaluation identifiers, and training configurations. Aggregate development scores are traceable to Appendix E, but do not by themselves permit exact reproduction.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: This draft includes method descriptions and aggregate result provenance, but not an executable training/evaluation release or the prediction files needed to reproduce the reported scores.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [No]

   Justification: The available settings and unresolved details are explicitly separated in Appendix D; exact splits, hyperparameters, and optimizer configurations remain to be supplied.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Only aggregate scores and two populated checkpoints per HotpotQA training block are available. Section 6 makes no statistical-significance claim because per-example predictions and repeated runs are unavailable.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [No]

   Justification: Hardware, per-run execution time, and total project compute are not available in the current records; Appendix D identifies these omissions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [N/A]

   Justification: An author-completed review of the NeurIPS Code of Ethics has not been documented in the materials available for this working draft; the authors must complete that review before submission.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: Section 7 discusses privacy and untrusted-input risks, but a balanced assessment of potential positive and negative societal impacts has not yet been completed.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This draft does not release trained model weights or a newly collected dataset. Release-specific risk assessment and safeguards must be revisited if such assets are added.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [No]

Justification: The paper credits the underlying models, datasets, and methods, but the exact asset revisions and their licenses and terms of use have not all been documented; this remains part of the reproducibility audit.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: The proposed pipeline and learned policy are described, but a documented code/checkpoint release with complete configuration, licensing, and usage information is not yet provided.

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The experiments described in this draft use existing benchmarks and model workflows; no crowdsourcing or direct human-participant study is reported.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: No direct human-participant study is reported in this work, so participant-risk disclosures and IRB approval are not applicable to the described experiments.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: The method and training sections describe the shared Qwen3-VL-8B policy, memory generation, image tools, and outcome-supervised optimization; Appendix D identifies implementation details still to be confirmed.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.