RETAIN AND REVEAL: MULTI-CONTINUATION TRAINING THROUGH AN OPTICAL MEMORY BOTTLENECK

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language model agents must preserve evidence for decisions whose requirements become clear only as new observations arrive. Long-context question answering provides a concrete setting: later passages can determine which earlier relations are needed. Under limited memory, these relations must survive both selective retention and compression into a readable representation. We introduce Retain and Reveal (R&R), which trains one committed memory to support multiple source-grounded continuations. The writer commits without access to the continuation probes. Its memory is rendered once, and the reader answers the same question using the identical image paired with different continuations. Continuation rewards encourage retention of unresolved alternatives, while reader supervision trains evidence recovery through the optical channel. Paired memory skills coordinate evidence organization and interpretation, with failure-driven revisions validated on held-out histories under frozen model weights. Experiments on long-context QA benchmarks demonstrate competitive accuracy across settings under matched answer-time memory-input token allowances. Gains over textual memory approaches are more pronounced at smaller budgets, supporting the value of training bounded memory to preserve and recover evidence for subsequent reasoning. Controlled comparisons show mean-accuracy gains from continuation diversity, joint probe supervision, and paired skill revision.

1 INTRODUCTION

Large language model (LLM) agents rely on memory to carry information across observations, interactions, and reasoning steps. As their histories grow, retaining the entire context becomes increasingly expensive, motivating systems that manage a compact working context alongside persistent information (Packer et al., 2023; Chhikara et al., 2025). The central challenge extends beyond storing past observations: an agent must preserve evidence for decisions whose requirements may become clear only later. Long-context question answering offers a concrete instance of this challenge, where later passages determine which earlier relations are needed to answer a question.

Existing approaches address this challenge through memory construction and context compression. MemAgent and Mem-α learn to update memory from sequential observations using downstream task rewards (Yu et al., 2026; Wang et al., 2025b). AutoCompressors and ICAE encode context into compact learned representations (Chevalier et al., 2023; Ge et al., 2024). More recently, DeepSeek-OCR and Glyph have explored representing text through visual tokens, while AgentOCR applies optical compression to agent histories (Wei et al., 2025; Cheng et al., 2025; Feng et al., 2026). MemOCR further exploits layout to allocate visual space according to evidence importance (Shi et al., 2026). Together, these developments make memory construction a problem of allocating limited representational capacity: useful evidence must be selected, organized, and preserved in a form that supports subsequent reasoning (Figure 1).

However, allocating this capacity requires anticipating what will matter. Consider a prefix stating that Lin was advised by Xu and Su by Tang, followed later by a passage identifying the winning author. To answer "Who advised the winning author?", the agent must preserve both relations until the winner is known. We call this *delayed relevance*: the question is available, but the evidence

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.

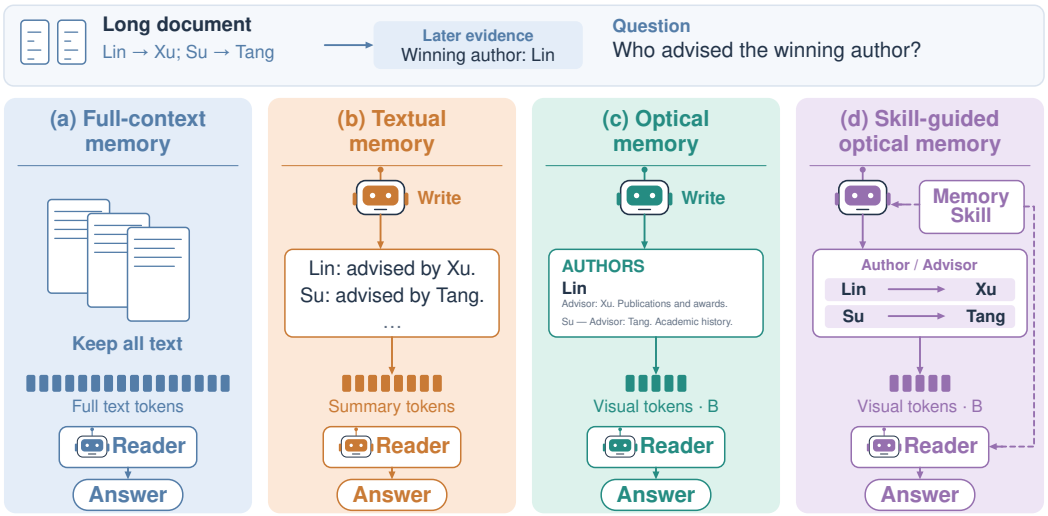


Figure 1: **Allocating memory capacity for later use.** Textual memory selects a compact state; optical memory exposes it through a budgeted image. R&R tests the same rendered memory under multiple hidden continuations to train retention and visual recovery. Paired skills coordinate writing and reading.

selecting the relevant earlier relation arrives later. Final-answer training can encourage useful retention, yet it evaluates only the realized continuation. It does not directly test whether the same intermediate memory would support a different later selection. Optical compression makes this allocation problem especially explicit: keeping an alternative in the textual state is insufficient if its relation becomes unreadable after rendering.

We introduce Retain and Reveal (R&R), which trains memory to support multiple later uses through a budgeted optical channel. At an intermediate document boundary, the writer commits to a memory before seeing the continuation probes used to evaluate it. The memory is rendered once, and the identical image is tested with several source-grounded continuations. Each continuation supplies evidence or a controlled selector requiring a different prefix relation, without revealing the answer itself. These tests reward preservation of unresolved alternatives. Because answers are evaluated after rendering, they also require the retained relations to remain visually accessible. Writer rewards and reader answer supervision thus address two connected requirements: retaining potentially useful evidence and recovering it through the deployed representation.

Coordinating these requirements also calls for shared conventions between writer and reader. Prior work demonstrates the value of reusable procedures and lessons derived from failures, including Reflexion, ExpeL, Agent Workflow Memory, and MemSkill (Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2024; Zhang et al., 2026). R&R maintains a *Paired Memory Skill* containing instructions for organizing relations and interpreting their rendered layout. Continuation failures motivate revisions, which are accepted through held-out comparisons with model weights frozen. The skill provides a procedural interface between what the writer preserves and how the reader accesses it.

Our contributions are threefold. First, we introduce *multi-continuation training* to test a committed intermediate memory against alternative later uses. Second, we evaluate this objective through the *optical channel*, coupling evidence retention with visual recoverability through complementary writer and reader updates. Third, we develop validated revision of *paired memory skills* to coordinate organization and interpretation. Together, these mechanisms train bounded memory around the future use of evidence: preserving what may matter and making it accessible when needed.

2 Related Work

Agent memory. MemGPT manages active context and external storage, while Mem0 consolidates persistent records (Packer et al., 2023; Chhikara et al., 2025). MemAgent and Mem-α learn memory

updates through reinforcement learning (Yu et al., 2026; Wang et al., 2025b). LightMem, SimpleMem, and HyMem study efficient storage, consolidation, and retrieval (Fang et al., 2026; Liu et al., 2026; Zhao et al., 2026). R&R builds on recurrent document reading and targets premature deletion by evaluating the same intermediate memory under several continuations.

Context compression. AutoCompressors and ICAE learn compact representations (Chevalier et al., 2023; Ge et al., 2024). DeepSeek-OCR and Glyph encode text visually; VTC-R1 applies this channel to reasoning; AgentOCR compresses agent histories; and VTCBench evaluates compressed-context understanding (Wei et al., 2025; Cheng et al., 2025; Wang et al., 2026; Feng et al., 2026; Zhao et al., 2025). MemOCR learns layout-aware visual memory (Shi et al., 2026). R&R maintains a textual recurrent state and applies optical compression when reading, training both evidence retention and recovery through the image. Its continuation probes also serve a different purpose from ICAE's next-text prediction: they supply later evidence and ask questions requiring earlier relations, testing the same pixels under alternative uses.

Reusable skills. Reflexion and Self-Refine use feedback to improve later trials or outputs (Shinn et al., 2023; Madaan et al., 2023). Voyager accumulates skills, ExpeL extracts lessons, and Agent Workflow Memory stores task procedures (Wang et al., 2023; Zhao et al., 2024; Wang et al., 2024). SPIN studies iterative self-training, while SAGE combines reinforcement learning with a skill library (Chen et al., 2024; Wang et al., 2025a). MemSkill revises memory-management skills (Zhang et al., 2026). Our paired skills coordinate optical writing and reading; evaluating revisions under frozen model weights separates their contribution from parameter learning.

3 Method

3.1 Task and Inference

Let $D = (C_1, \ldots, C_T)$ be a document split into T chunks, with question q known before reading. We instantiate both the writer W_θ and reader R_ϕ with Qwen3-VL-8B-Instruct, a multimodal large language model (MLLM) (Bai et al., 2025). The writer uses its text interface to maintain memory; the reader takes a memory image together with the question and produces a textual answer. Skill $S = (S^{\mathrm{w}}, S^{\mathrm{r}})$ supplies their respective instructions (Appendix C).

The writer carries structured textual memory M_t of at most L_m tokens between chunks:

$$M_t \sim W_\theta(\cdot \mid M_{t-1}, C_t, q; S^{\mathrm{w}}), \qquad M_0 = \varnothing. \tag{1}$$

After the final chunk, a deterministic renderer converts M_T to HTML and rasterizes the HTML into an image. With fixed settings, the same memory and budget produce identical pixels. Resizing is chosen so that the reader's image processor encodes the image within visual-token budget B. Denoting this budgeted image by V_T, we have

$$V_T = \mathrm{Render}_B(M_T), \qquad \hat{a} \sim R_\phi(\cdot \mid V_T, q; S^{\mathrm{r}}). \tag{2}$$

The renderer has no trainable parameters. The reader recovers the answer from the image, without access to the source document or unrendered memory.

Recurrent capacity versus readout budget. L_m limits the *text tokens* carried between writer updates; B limits the *visual tokens* presented to the reader after image preprocessing and patch merging. We fix $L_m = 2{,}048$ and vary B independently. A 64-token optical read therefore does not truncate the writer's memory to 64 text tokens. Question, skill, and prompt tokens are additional reader inputs; answer tokens are outputs. Figure 2 shows this separation between textual recurrence and the final optical read.

3.2 Preserving Relations for Later Use

Delayed relevance requires retaining relations together with their arguments, such as an advisor and the corresponding author. For prefix $P_t = (C_1, \ldots, C_t)$ and later evidence u, we seek to preserve the answer distribution available from the full prefix:

$$p^*(a \mid P_t, q, u) \approx R_\phi(a \mid \mathrm{Render}_B(M_t), q, u; S^{\mathrm{r}}). \tag{3}$$

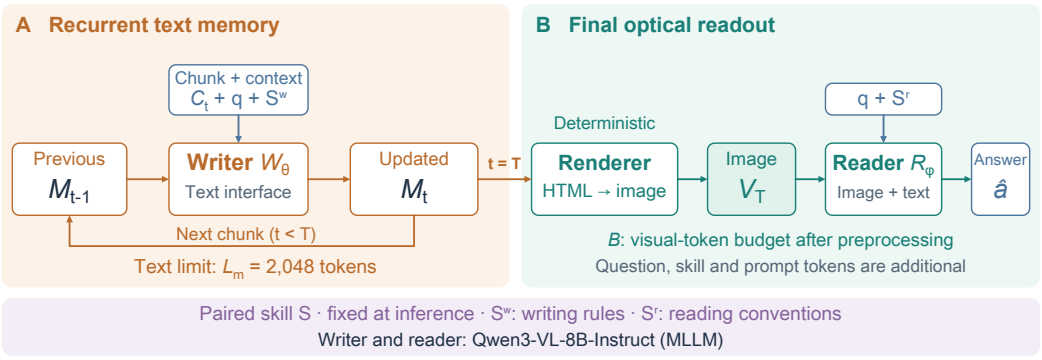


Figure 2: **Inference Pipeline.** The writer recurrently updates textual memory using the question and current chunk. The final memory is rendered for visual reading. A fixed shared skill pairs the writing rules with their reading conventions.

Here p^* denotes the full-prefix answer distribution. This *conditional sufficiency* is a training goal whose coverage depends on the continuations tested. In the author–advisor example, memory must retain both relations while omitting irrelevant biography. We train this behavior by holding the prefix and question fixed and varying the later evidence that selects a relation. Evaluating answers after rendering tests whether the relation was both retained and readable.

3.3 Multi-Continuation Training

Final-QA supervision evaluates one realized continuation; R&R additionally tests whether the same intermediate memory supports several later uses. The writer commits before the continuations are revealed, so it must preserve unresolved relations rather than tailor memory to a disclosed branch. All branches read identical pixels, exposing failures from both deletion and unreadable rendering. Their rewards train retention, while answer supervision trains visual recovery. These extra reads occur only during training; inference still uses one final render and read. Figure 3 separates model updates from skill revision.

Source-grounded probes. At boundary t, a valid set $\mathcal{U}_t = \{(u_j, a_j)\}_{j=1}^{K}$ contains $K \geq 2$ continuations and answers. Each answer requires a prefix relation together with the continuation's evidence or selector; the continuation alone cannot reveal it. Probes use suffix passages or controlled selectors over source-supported relations. Controlled alternatives may differ from the original suffix, but require a supported evidence path. In the default Multi condition, we exclude ambiguous probes and omit auxiliary terms when fewer than two valid probes exist.

For Lin–Xu and Su–Tang, two probes select Lin or Su and require Xu or Tang. The writer commits to the tested state before seeing either selector.

Shared-memory reward. One rendered image is reused across all branches:

$$V_t = \mathrm{Render}_B(M_t), \qquad \hat{a}_j \sim R_\phi(\cdot \mid V_t, q, u_j; S^r),$$
$$r_{\mathrm{cont}}(M_t) = \frac{1}{K}\sum_{j=1}^{K} \mathrm{SEM}(\hat{a}_j, a_j). \tag{4}$$

The normalized answer-matching score $\mathrm{SEM} \in [0, 1]$ is defined in the evaluation setup. With final gold answer a^* and $r_{\mathrm{QA}} = \mathrm{SEM}(\hat{a}, a^*)$, the writer maximizes

$$J_W(\theta; \phi, S) = \mathbb{E}[r_{\mathrm{QA}} + \lambda_W r_{\mathrm{cont}}], \qquad \lambda_W \geq 0. \tag{5}$$

The expectation covers training examples, boundaries, and sampled outputs, with $r_{\mathrm{cont}} = 0$ at ineligible boundaries. Weight λ_W balances final-QA and intermediate retention rewards.

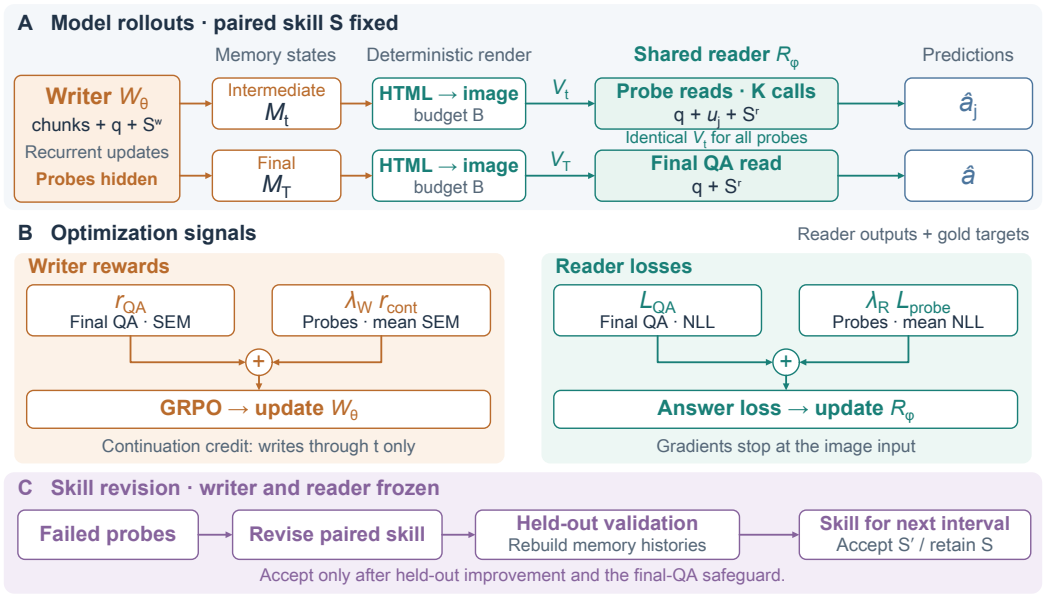


Figure 3: **Training Pipeline.** One memory is written before the probes are revealed, then read with different continuations using shared reader weights. Answers supply writer rewards and reader supervision. Failed probes guide skill revision, accepted on held-out histories with both models frozen.

Writer updates. GRPO (Shao et al., 2024) compares candidate trajectories for the same document, question, and skill. Each candidate's intermediate memory is scored before recurrence resumes. A probe at t credits only writes through t, while final-QA reward credits the full trajectory. Appendix A gives the step-dependent advantages and clipped objective. The updates require no gradient through generation or rendering.

Reader updates. The reader minimizes answer-sequence loss through the budgeted images:

$$\mathcal{L}_R = -\mathbb{E}\left[\log R_\phi(a^* \mid V_T, q; S^{\mathrm{r}}) + \frac{\lambda_R}{K}\sum_{j=1}^{K}\log R_\phi(a_j \mid V_t, q, u_j; S^{\mathrm{r}})\right]. \tag{6}$$

Weight $\lambda_R \geq 0$ controls probe supervision independently of the writer reward; its term is omitted without valid probes. Gold answers serve only as targets. Gradients update reader parameters and stop at the image input. All reader calls in Figure 3 share one model.

The independent ablations remove writer continuation reward ($\lambda_W = 0$), reader probe loss ($\lambda_R = 0$), or skill revision (fixed initial skill). Each retains the other updates, final-QA supervision, and the inference procedure in Equations 1–2.

3.4 Paired Skill Coordination

Paired writing and reading rules coordinate how relations are organized and interpreted. Periodically, we freeze both models, propose a bounded revision from failed probes, and rebuild held-out memories to evaluate its effect on both roles. A revision is accepted only when the combined validation score improves without exceeding a tolerance on final-QA loss. Revision and acceptance examples are disjoint and exclude test data. Skills remain fixed between revisions and at inference. Appendix C gives the instruction format and acceptance criterion.

3.5 Computation

For N document tokens and chunk size c, inference uses $T = \lceil N/c \rceil$ sequential writes. With skill limit L_s, each write has at most $L_m + c + |q| + L_s$ input tokens and L_m output tokens. Let $\mathcal{C}_W$ bound

this call's cost, $\mathcal{C}_R$ the final read with B visual tokens and bounded answer length, and $\mathcal{C}_{\text{render}}$ the final rendering cost. Then

$$\mathcal{C}_{\text{infer}} \leq T\mathcal{C}_W + \mathcal{C}_{\text{render}} + \mathcal{C}_R. \tag{7}$$

With fixed limits, work grows approximately linearly with document length. Reducing B changes the final read but leaves the T sequential writes. Training additionally renders each intermediate candidate once and reads that image under K continuations.

4 Experiments

We evaluate whether Retain and Reveal improves QA under limited memory, whether the gains depend on continuation diversity and the optical channel, and how the resulting accuracy compares with inference cost. The experiments separate the complete method from controls that hold the paired skill fixed.

4.1 Experimental Setup

Datasets and metric. Following the long-context QA setting of Shi et al. (2026), we train on HotpotQA with supporting and distractor documents totaling approximately 30K tokens. Evaluation covers multi-hop HotpotQA and 2WikiMultiHopQA (2Wiki), and single-hop Natural Questions (NQ) and TriviaQA, at 10K, 30K, and 100K context lengths (Yang et al., 2018; Ho et al., 2020; Kwiatkowski et al., 2019; Joshi et al., 2017). Each dataset contains 128 evaluation instances. We report SEM accuracy (%), averaged over three runs; the main tables average the four dataset scores with equal weight. SEM follows the normalized answer-matching rule used by Yu et al. (2026), detailed in Appendix D. Score differences are stated in percentage points.

Baselines and memory budgets. Our controlled comparisons use Qwen3-VL-8B-Instruct and the same evaluation instances. We include Mem0, Mem-α, and MemAgent with prefix readout, two budget-aware MemAgent readouts—question-aware rewriting and LLMLingua-2 compression—and the optical-memory baseline MemOCR (Chhikara et al., 2025; Wang et al., 2025b; Yu et al., 2026; Pan et al., 2024; Shi et al., 2026). The latter is evaluated under our common-backbone setup; its scores are distinct from the original paper's Qwen2.5-based evaluation. We match answer-time memory-input token allowances: B counts retained text tokens or processed visual tokens. Question and prompt overhead are additional inputs. These matched allowances do not imply equivalent overall resources, computation, or latency. Table 1 focuses on $B = 64$ and 256; Appendix F includes the 16- and 1024-token settings. Full configurations and comparison protocols appear in Appendices D and E.

4.2 Overall Performance

R&R improves on budget-aware textual memory. R&R achieves the highest average accuracy in five of the six settings in Table 1. Its gains over the strongest textual baseline, MemAgent + rewrite, range from 1.1 to 5.8 percentage points at $B = 256$ and from 5.2 to 9.4 points at $B = 64$. Active compression substantially strengthens the textual comparison: at 30K and $B = 64$, rewriting raises MemAgent from 50.3% to 59.1%, while R&R reaches 67.4%. The remaining advantage therefore extends beyond the loss caused by prefix truncation. Compared with MemOCR, R&R improves by 1.3–2.7 points across all three context lengths at $B = 64$. At $B = 256$, it gains 3.0 and 2.2 points at 10K and 30K, respectively, but scores 66.8% at 100K, slightly below MemOCR's 67.1%.

The advantage over text is larger at smaller budgets. Reducing B from 256 to 64 lowers R&R's average accuracy by 4.3–5.8 points, compared with 8.1–9.4 points for MemAgent + rewrite. At 30K, these declines correspond to 6.0% and 12.1% relative to each method's 256-token score. MemOCR loses 4.8–6.4 points over the same budget reduction. Both optical methods thus show smaller declines than budget-aware textual rewriting, suggesting that robustness to tighter budgets is partly shared by optical memory. The following controls examine the additional role of multi-continuation training. These comparisons describe mean scores; run-level dispersion remains incomplete, so we do not infer statistical significance from the observed gaps.

Table 1: **Long-context QA accuracy under matched answer-time memory-input token allowances.** SEM accuracy (%), averaged over three runs and four datasets. Columns denote source-context length; bold marks the highest mean at each budget and length. Prefix rows use truncation; the remaining textual readouts actively compress the same MemAgent state.

Method	Readout	10K	30K	100K
Answer-time memory allowance: 256 tokens				
Mem0 (prefix) (2025)	Text	61.3	58.0	55.0
Mem-α (prefix) (2025b)	Text	40.3	37.0	38.7
MemAgent (prefix) (2026)	Text	67.3	65.3	63.9
MemAgent + budget rewrite	Text	69.8	67.2	65.7
MemAgent + LLMLingua-2 (2024)	Text	69.0	66.4	64.8
MemOCR (2026)	Optical	72.6	69.5	**67.1**
R&R	Optical	**75.6**	**71.7**	66.8
Answer-time memory allowance: 64 tokens				
Mem0 (prefix) (2025)	Text	56.0	53.2	53.0
Mem-α (prefix) (2025b)	Text	25.2	26.4	26.4
MemAgent (prefix) (2026)	Text	50.7	50.3	48.7
MemAgent + budget rewrite	Text	60.4	59.1	56.8
MemAgent + LLMLingua-2 (2024)	Text	60.0	58.4	56.2
MemOCR (2026)	Optical	67.3	64.7	60.7
R&R	Optical	**69.8**	**67.4**	**62.0**

4.3 Continuation Diversity and Memory Representation

We cross text and optical memory with three training conditions: final QA only, repeated reads of one continuation (Single repeat), and distinct continuations (Multi). All six conditions use the same fixed initial skill. Single repeat and Multi both make K probe reads of one committed memory, with matched numbers of writer rollouts and optimization steps. Single repeat samples one valid continuation from the same eligible pool, then draws K independent reader responses; Multi uses K distinct continuations. Both average rewards and losses over the K reads and keep the committed memory unchanged. Their difference is which later evidence the reader receives, rather than the number of auxiliary reads.

Table 2: **Continuation diversity and memory representation.** SEM accuracy (%), averaged over three runs and four datasets at 30K source length. The initial skill is fixed, revision is disabled, and final-QA supervision remains active. $K \geq 2$; B is the memory-input allowance.

Form	Training	Distinct branches	Probe reads per memory	SEM $B = 64$	SEM $B = 256$
Text	Final QA only	0	0	56.2	62.4
Text	Single repeat	1	K	58.1	64.0
Text	Multi	K	K	**60.0**	**65.4**
Optical	Final QA only	0	0	57.1	63.3
Optical	Single repeat	1	K	59.6	65.2
Optical	Multi	K	K	**62.8**	**67.8**

Distinct continuations improve both representations. At $B = 64$, replacing Single repeat with Multi raises text accuracy from 58.1% to 60.0% and optical accuracy from 59.6% to 62.8% (Table 2); the corresponding gains at $B = 256$ are 1.4 and 2.6 points. Distinct later uses thus benefit a committed memory beyond repeated testing of one continuation. The optical–text gap also grows: from 0.9 points at either budget under final QA alone, to 1.5/1.2 with Single repeat and 2.8/2.4 with Multi ($B = 64/256$). This pattern is consistent with learning to preserve relations through rendering, but does not establish a statistical interaction. Reader-call matching does not establish equal computation; processed-token and GPU-hour comparisons are not available (Appendix E).

4.4 Method Ablations

Table 3: **Method ablations under optical inference.** Four-task mean SEM (%) at 30K. Writer reward is absent (Off), image-scored, or text-scored; Reader indicates the image-based probe loss. Final-QA supervision is retained. Values are means without run-level SD.

Variant	Writer reward	Reader probe	Skill	SEM $B = 64$	SEM $B = 256$
Final QA only	Off	Off	Fixed	57.1	63.3
Writer reward only	Image	Off	Fixed	58.2	65.4
Reader loss only	Off	On	Fixed	57.4	64.6
Joint probes	Image	On	Fixed	62.8	67.8
Text-scored writer reward	Text	On	Fixed	64.1	68.5
Full R&R (matched revision)	Image	On	Revised	**67.4**	**71.7**

Final QA only and Joint probes reuse the optical endpoints of Table 2. Full R&R enables skill revision under matched training and evaluation settings.

Probe supervision and skill revision. At $B = 64/256$, writer reward alone improves over final QA by 1.1/2.1 points, and reader loss alone by 0.3/1.3 (Table 3). Joint probes gain 5.7/4.5 points, exceeding either term alone. With image-scored joint probes, matched skill revision adds 4.6/3.9 points, reaching 67.4/71.7% as in the 30K main comparison.

Feedback before versus after rendering. With the skill fixed, text-scored writer reward exceeds image-scored Joint probes by 1.3/0.7 points; reader supervision and inference remain optical. These means support continuation training but do not establish an advantage for scoring writer rewards after rendering. The text-scored reward uses up to L_m text tokens rather than B visual tokens; training-token costs are not matched (Appendix E).

4.5 Evidence Placement and Visual Access

At 30K context, we place the same ground-truth evidence in prominent (Crucial) or dense (Detailed) regions on HotpotQA, 2Wiki, and NQ. These earlier-series oracle interventions test visual access and are excluded from the main QA comparison.

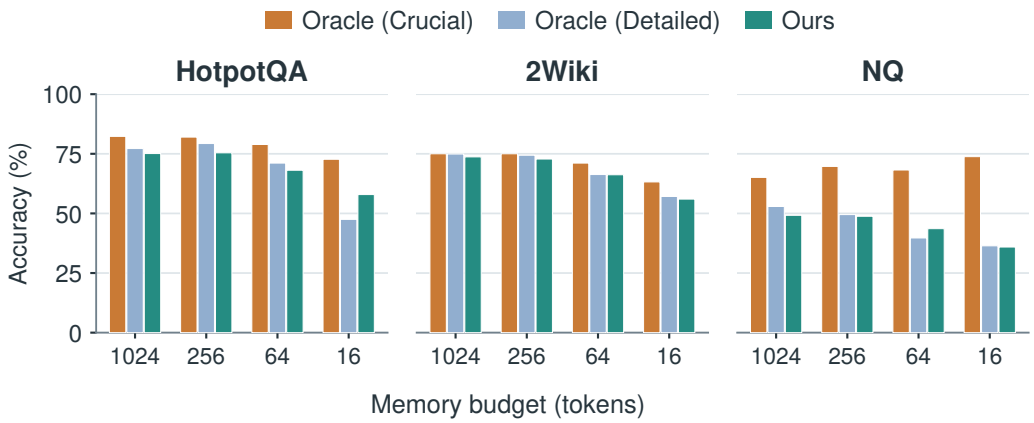


Figure 4: **Effect of evidence placement at 30K context.** SEM (%) in the earlier diagnostic series. Crucial and Detailed insert identical evidence into different regions. Ours uses ordinary memory from Appendix F, separate from the updated Table 1 aggregates. Budgets 16 and 1024 are nominal.

Prominent placement generally improves accuracy, especially at the smallest budget (Figure 4). At nominal $B = 16$ on HotpotQA, Detailed scores 47.5%, below ordinary memory's 57.9%, while

Crucial reaches 72.7%. Adding evidence therefore does not ensure recovery after compression. This diagnostic concerns visual access, not continuation-training attribution. Appendix H traces a concrete case through memory, compression, and answering: truncation deletes the relevant name, while uniform rendering retains it but makes it unreadable.

4.6 Inference Cost

Table 4: **Realized input and inference cost.** Measurements at 30K context. Memory tokens are shown as mean/max; total reader input and image size are means. Image dimensions are in pixels and latency is in seconds. Equal nominal B does not establish equal cost.

(a) Actual memory and reader input

Method	B	Memory tokens mean / max	Total reader input tokens	Image $W \times H$	Image KiB
MemAgent + rewrite	64	(62.1/64)	214.1	N/A	N/A
MemAgent + LLMLingua-2	64	(63.0/64)	215.0	N/A	N/A
MemOCR	64	(64/64)	232.0	(256×256)	16.8
R&R	64	(64/64)	232.0	(256×256)	17.6
MemAgent + rewrite	256	(250.8/256)	402.8	N/A	N/A
MemAgent + LLMLingua-2	256	(253.2/256)	405.2	N/A	N/A
MemOCR	256	(256/256)	424.0	(512×512)	55.2
R&R	256	(256/256)	424.0	(512×512)	57.9

(b) End-to-end inference latency (seconds)

Method	B	Write	Compress/ render	Read	Total
MemAgent + rewrite	64	18.40	1.20	0.70	20.30
MemAgent + LLMLingua-2	64	18.40	0.25	0.70	19.35
MemOCR	64	19.00	0.10	0.76	19.86
R&R	64	19.00	0.10	0.67	**19.77**
MemAgent + rewrite	256	18.40	1.75	0.91	21.06
MemAgent + LLMLingua-2	256	18.40	0.30	0.91	19.61
MemOCR	256	19.00	0.16	0.99	20.15
R&R	256	19.00	0.16	0.87	**20.03**

Memory is counted after preprocessing; total reader input includes prompt overhead. Write, compression/rendering, and read latency are reported separately. N/A denotes inapplicable quantities. Comparison protocols are described in Appendix E.

At 30K, R&R takes 19.77/20.03 s at $B = 64/256$ (Table 4), 0.45/0.60% less than MemOCR, with accuracy gains of 2.7/2.2 points. Read latency drops by 11.8/12.1%, from 0.76/0.99 s to 0.67/0.87 s.

Writing accounts for 94.9–96.1% of latency; rendering takes 0.10–0.16 s. Faster reading therefore yields modest overall savings. R&R is faster than MemAgent + rewrite and slower than MemAgent + LLMLingua-2; sequential writing remains the dominant cost.

5 Conclusion

Retain and Reveal trains one committed memory for several later uses through an optical bottleneck. Writer rewards encourage retention, reader supervision trains visual recovery, and paired skills coordinate the two roles. Experiments show larger gains over budget-aware text compression at small memory budgets and a benefit from continuation diversity in both representations. These improvements retain the cost of sequential memory construction, motivating future work on recoverable storage and reuse across readers.

AI USE STATEMENT

Generative AI tools (ChatGPT/Codex) assisted with language refinement, analysis of experimental results, LaTeX typesetting, table preparation, and supporting scripts. The authors retain responsibility for the accuracy, originality, and attribution of the paper's content, including all AI-assisted material.

ETHICS STATEMENT

Compressed memory can omit qualifications, lose evidence, or associate facts with the wrong entities. A reader may therefore produce a confident but unsupported answer, particularly when optical compression makes relevant text difficult to recover. Such errors warrant evaluation before the method is used in consequential decisions. The method may also preserve biases present in its source documents or underlying models. Applying it to personal histories or private documents raises privacy concerns: rendering text as an image does not anonymize its contents. Deployment in these settings requires appropriate access controls, retention and deletion policies, and evaluation of both factual accuracy and information exposure.

REPRODUCIBILITY STATEMENT

Section 3 defines the memory interface and training objectives. Appendix A provides the training algorithm and writer update, while Appendices B and C describe probe construction, information boundaries, and the skill interface. The evaluation setup and comparison protocols are documented in Appendices D and E; Table 4 reports realized inference inputs and latency. Appendix F retains the available task-level results. Table 3 reports the main method ablations. The reported means do not include run-level dispersion, and the available call-matching controls do not establish equal training computation.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL Technical Report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-Play Fine-Tuning Converts Weak Language Models to Strong Language Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 6621–6642, 2024. URL https://arxiv.org/abs/2401.01335.

Jiale Cheng, Yusen Liu, Xinyu Zhang, Yulin Fei, Wenyi Hong, Ruiliang Lyu, Weihan Wang, Zhe Su, Xiaotao Gu, Xiao Liu, et al. Glyph: Scaling Context Windows via Visual-Text Compression. *arXiv preprint arXiv:2510.17800*, 2025. URL https://arxiv.org/abs/2510.17800.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting Language Models to Compress Contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, 2023. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv preprint arXiv:2504.19413*, 2025. URL https://arxiv.org/abs/2504.19413.

DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. LightMem: Lightweight and Efficient Memory-Augmented Generation. In *International Conference on Learning Representations*, 2026. URL https://doi.org/10.48550/arXiv.2510.18866.

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 preprint arXiv:2601.04786*, 2026. URL https://arxiv.org/abs/2601.04786.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context Autoencoder for Context Compression in a Large Language Model. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2307.06945.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing A Multi-hop QA Dataset for Comprehensive Evaluation of Reasoning Steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020. URL https://aclanthology.org/2020.coling-main.580/.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017. URL https://aclanthology.org/P17-1147/.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural Questions: A Benchmark for Question Answering Research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. URL https://aclanthology.org/Q19-1026/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://arxiv.org/abs/2005.11401.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. SimpleMem: Efficient Lifelong Memory for LLM Agents. *arXiv preprint arXiv:2601.02553*, 2026. URL https://arxiv.org/abs/2601.02553.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-Refine: Iterative Refinement with Self-Feedback. *arXiv preprint arXiv:2303.17651*, 2023. URL https://arxiv.org/abs/2303.17651.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. MemGPT: Towards LLMs as Operating Systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. LLMLingua-2: Data Distillation for Efficient and Faithful Task-Agnostic Prompt Compression. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 963–981, 2024. doi: 10.18653/v1/2024.findings-acl.57. URL https://aclanthology.org/2024.findings-acl.57/.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 Technical Report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

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. *arXiv preprint arXiv:2601.21468*, 2026. URL https://arxiv.org/abs/2601.21468.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language Agents with Verbal Reinforcement Learning. *arXiv preprint arXiv:2303.11366*, 2023. URL https://arxiv.org/abs/2303.11366.

Fanqi Wan, Weizhou Shen, Shengyi Liao, Yingcheng Shi, Chenliang Li, Ziyi Yang, Ji Zhang, Fei Huang, Jingren Zhou, and Ming Yan. QwenLong-L1: Towards Long-Context Large Reasoning Models with Reinforcement Learning. *arXiv preprint arXiv:2505.17667*, 2025. URL https://arxiv.org/abs/2505.17667.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*, 2023. URL https://arxiv.org/abs/2305.16291.

Jiongxiao Wang, Qiaojing Yan, Yawei Wang, Yijun Tian, Soumya Smruti Mishra, Zhichao Xu, Megha Gandhi, Panpan Xu, and Lin Lee Cheong. Reinforcement Learning for Self-Improving Agent with Skill Library. *arXiv preprint arXiv:2512.17102*, 2025a. URL https://arxiv.org/abs/2512.17102.

Yibo Wang, Yongcheng Jing, Shunyu Liu, Hao Guan, Rong-cheng Tu, Chengyu Wang, Jun Huang, and Dacheng Tao. VTC-R1: Vision-Text Compression for Efficient Long-Context Reasoning. *arXiv preprint arXiv:2601.22069*, 2026. URL https://arxiv.org/abs/2601.22069.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. Mem-α: Learning Memory Construction via Reinforcement Learning. *arXiv preprint arXiv:2509.25911*, 2025b. URL https://arxiv.org/abs/2509.25911.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent Workflow Memory. *arXiv preprint arXiv:2409.07429*, 2024. URL https://arxiv.org/abs/2409.07429.

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

An Yang, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoyan Huang, Jiandong Jiang, Jianhong Tu, Jianwei Zhang, Jingren Zhou, et al. Qwen2.5-1M Technical Report. *arXiv preprint arXiv:2501.15383*, 2025. URL https://arxiv.org/abs/2501.15383.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018. URL https://aclanthology.org/D18-1259/.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. MemAgent: Reshaping Long-Context LLM with Multi-Conv RL-based Memory Agent. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.02259.

Haozhen Zhang, Quanyu Long, Jianzhu Bao, Tao Feng, Weizhi Zhang, Haodong Yue, and Wenya Wang. MemSkill: Learning and Evolving Memory Skills for Self-Evolving Agents. *arXiv preprint arXiv:2602.02474*, 2026. URL https://arxiv.org/abs/2602.02474.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM Agents Are Experiential Learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2024. URL https://arxiv.org/abs/2308.10144.

Hongbo Zhao, Meng Wang, Fei Zhu, Wenzhuo Liu, Bolin Ni, Fanhu Zeng, Gaofeng Meng, and Zhaoxiang Zhang. VTCBench: Can Vision-Language Models Understand Long Context with Vision-Text Compression? *arXiv preprint arXiv:2512.15649*, 2025. URL https://arxiv.org/abs/2512.15649.

Xiaochen Zhao, Kaikai Wang, Xiaowen Zhang, Chen Yao, and Aili Wang. HyMem: Hybrid Memory Architecture with Dynamic Retrieval Scheduling. *arXiv preprint arXiv:2602.13933*, 2026. URL https://arxiv.org/abs/2602.13933.

A Training Procedure

Algorithm 1 specifies when each input becomes available. All candidate trajectories use the same document, question, skill, and sampled boundary. The writer produces the tested state before the continuation probes are revealed to the reader. Their gold answers enter only the score and supervised loss.

Algorithm 1 Model training and periodic skill revision

Require: Training data; disjoint revision and validation data; initial skill S; writer W_θ; reader R_ϕ

for each model-training interval **do**
  Fix S; sample document D, question q, gold answer a^*, and prefix boundary t.
  Construct valid source-grounded probes $\{(u_j, a_j)\}_{j=1}^{K}$; withhold them from the writer.
  Sample G recurrent writing trajectories from the empty initial state.
  for each trajectory g **do**
    if at least one probe read is enabled **then**
      Render $M_t^{(g)}$ once; reuse the image for the K continuation reads.
      Score the continuation answers to obtain $C_g = r_{\text{cont}}(M_t^{(g)})$.
    else
      Set $C_g = 0$ and omit the reader's auxiliary loss.
    end if
    Read the final rendered memory and score the answer to obtain $F_g = r_{\text{QA}}^{(g)}$.
  end for
  Update the writer with the step-dependent GRPO objective in Equation 11.
  Update the reader with Equation 6.
  if a skill-revision boundary is reached **then**
    Freeze the models; propose a bounded skill edit S' from failed probes.
    Rebuild held-out histories under S and S' with matched evaluation controls.
    Apply Equation 12; retain S if the revision is rejected.
  end if
end for

A.1 Local Credit for the Writer

The final answer depends on the full writing trajectory, but a probe at boundary t depends only on writes through t. We make that distinction explicit in the return used for each writer update. Let $g \in \{1, \dots, G\}$ index candidate trajectories, and let $s \in \{1, \dots, T\}$ index their writing steps. Write F_g for final-QA reward and C_g for the continuation reward at the sampled boundary. The step-dependent return is

$$b_{g,s} = F_g + \lambda_W \mathbf{1}\{s \le t\}\, C_g. \tag{8}$$

Equation 8 assigns final-QA credit to every writing step. It assigns continuation credit only to steps that can affect the tested state. Setting $\lambda_W = 0$ removes this auxiliary writer reward without removing the reader probe loss. Ineligible boundaries have $C_g = 0$.

We center and normalize these returns separately at each writing step:

$$\begin{aligned} \bar{b}_s &= \frac{1}{G}\sum_{g=1}^{G} b_{g,s}, \qquad \sigma_s = \sqrt{\frac{1}{G}\sum_{g=1}^{G}(b_{g,s} - \bar{b}_s)^2}, \\ A_{g,s} &= \frac{b_{g,s} - \bar{b}_s}{\sigma_s + \delta}, \qquad \delta > 0. \end{aligned} \tag{9}$$

In Equation 9, $A_{g,s}$ is the advantage for tokens generated at step s, and δ is a numerical stabilizer. The grouping compares trajectories for the same training example and boundary. If every candidate has the same return at a step, its centered policy signal is zero. A single advantage broadcast across the entire trajectory would incorrectly send an intermediate probe's credit to later writes.

Let $z_{g,s,i}$ be token i of the memory generated at step s, and let $n_{g,s}$ be the length of that output. Its conditioning input is $h_{g,s,i} = (M_{s-1}^{(g)}, C_s, q, S^{\mathrm{w}}, z_{g,s,<i})$. For the rollout policy $W_{\theta_{\mathrm{old}}}$, define

$$\rho_{g,s,i}(\theta) = \frac{W_\theta(z_{g,s,i} \mid h_{g,s,i})}{W_{\theta_{\mathrm{old}}}(z_{g,s,i} \mid h_{g,s,i})}. \tag{10}$$

Equation 10 measures how the updated writer changes the probability of a sampled output token. Input chunks, questions, and skill tokens are conditioning information, not sampled writer actions.

With $N_g = \sum_{s=1}^{T} n_{g,s}$, the writer maximizes the clipped GRPO surrogate

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{G}\sum_{g=1}^{G}\frac{1}{N_g}\sum_{s=1}^{T}\sum_{i=1}^{n_{g,s}} \Big\{\min\big(\rho_{g,s,i}A_{g,s},\ \mathrm{clip}(\rho_{g,s,i}, 1-\eta, 1+\eta)A_{g,s}\big) - \beta\mathcal{K}_{g,s,i}(\theta)\Big\}\Bigg]. \tag{11}$$

where $\eta > 0$ is the clipping parameter, $\beta \geq 0$ weights the reference-policy penalty, and $\mathcal{K}_{g,s,i}(\theta) = D_{\mathrm{KL}}(W_\theta(\cdot \mid h_{g,s,i}) \| W_{\mathrm{ref}}(\cdot \mid h_{g,s,i}))$. Equation 11 increases the probability of actions with positive relative advantage while limiting large policy changes. The returns and advantages are held fixed during this update. The expression uses conditional KL notation; it does not require differentiating through the score, the reader output, or rendering.

B Probe Construction and Information Boundaries

During a probe read, the continuation is supplied directly to the reader. Deployment instead resumes recurrent writing over the remaining source chunks before the final read. The auxiliary task tests the usefulness of an intermediate state, while final-QA supervision evaluates the deployed writing trajectory.

A probe associates a prefix boundary with a continuation, answer aliases, and source spans establishing the answer. An explicit evidence path links a prefix relation to the selector or later evidence in the continuation. The following conditions define valid tests:

1. The writer has access only to observed chunks, previous memory, the question, and the writing skill when producing the tested state.
2. The relevant prefix relations are supported by the original document. A controlled selector is distinguished from a naturally occurring suffix passage.
3. The continuation does not state the target answer or provide the complete reasoning chain without the prefix.
4. Different branches require different prefix relations; branches with distinct answers are especially useful for detecting premature selection.
5. Every branch reads the same memory bytes, rendered pixels, and visual budget. No branch-specific rewriting is allowed before that read.
6. Training probe construction and skill revision do not use final test instances. Revision examples and acceptance-validation examples are disjoint. Any held-out branch diagnostic is evaluation-only.

These controls make the auxiliary task a test of retained evidence. They cannot establish sufficiency outside the tested branches. A single-hop example whose answer is already resolved may not support two valid continuations; it still contributes to final-QA training, with the auxiliary terms omitted.

C Paired Memory Skill: Format and Revision

C.1 Instruction Format

The skill $S = (S^{\mathrm{w}}, S^{\mathrm{r}})$ consists of two bounded natural-language instruction blocks. The writer receives S^{w} with each memory-update call; the reader receives S^{r} with the rendered image and

question, together with a continuation during probe reads. The blocks specify complementary organization and reading rules. They contain reusable procedures rather than document-specific evidence or reference answers. A minimal example is:

> **Writing block (S^{w}).** Keep each evidence relation together with its arguments. Preserve unresolved alternatives relevant to the question. Use explicit labels and compact groups. Give visual prominence to complete relations rather than isolated names.
>
> **Reading block (S^{r}).** Identify the requested relation. Use the incoming evidence to select a record, then read its arguments together. Preserve the direction of the relation; proximity alone does not establish a link.

The example illustrates the interface; validated revisions may change the instructions. Memory itself may use records, aligned rows, or labeled groups, without a prescribed table layout. A revision edits the shared rules rather than inserting example-specific facts. The skill-length limit L_s is separate from the recurrent memory limit L_m and the visual-token budget B; skill tokens are counted as prompt overhead rather than memory content.

C.2 Revision and Acceptance

After a model-training interval, we freeze θ, ϕ and propose a bounded edit S' from failed probes. Losing the relation between two retained names, for example, can motivate a labeled record with a corresponding reading rule. Skills encode reusable procedures, excluding document-specific answers.

We evaluate each candidate by rebuilding held-out histories from empty memory under matched budgets, questions, and decoding controls. Rebuilding tests the combined change to writing and reading. Let $A_{\mathrm{QA,val}}$ and $A_{\mathrm{cont,val}}$ denote normalized final-QA and continuation scores, and define $J_{\mathrm{val}} = A_{\mathrm{QA,val}} + \lambda_{\mathrm{val}} A_{\mathrm{cont,val}}$. The validation weight $\lambda_{\mathrm{val}} \geq 0$ remains fixed across model-loss ablations, independently of λ_W, λ_R. We accept

$$S \leftarrow S' \quad \text{if} \quad \begin{aligned} & J_{\mathrm{val}}(S') > J_{\mathrm{val}}(S) \\ & \text{and} \quad A_{\mathrm{QA,val}}(S') \geq A_{\mathrm{QA,val}}(S) - \epsilon. \end{aligned} \tag{12}$$

An accepted edit improves the combined score while limiting final-QA loss to $\epsilon \geq 0$; otherwise we retain S. Revision and acceptance examples are disjoint and exclude test data. Accepted skills obey the length bound and remain fixed until the next revision stage and throughout inference.

D Experimental Setup

Tasks and metric. Training uses HotpotQA examples with supporting and distractor documents totaling about 30K tokens. Evaluation covers HotpotQA, 2Wiki, NQ, and TriviaQA at 10K, 30K, and 100K tokens, with 128 instances per dataset and three runs (Yang et al., 2018; Ho et al., 2020; Kwiatkowski et al., 2019; Joshi et al., 2017). Following Shi et al. (2026), we report subword exact match (SEM) as accuracy in percent. The scoring rule follows the normalized answer matching (`sub_em`) described by Yu et al. (2026): the extracted response is checked for the required answer content after normalization of case, punctuation, and articles. For multipart answers requiring several elements, the score is the fraction recovered. Thus $\mathrm{SEM} \in [0, 1]$ and does not require an exact match of the entire response string. All methods use the same rule. Within each run, the macro score is the arithmetic mean of the four dataset scores; the reported accuracy averages these macro scores over runs. A change from a to b percent is $b - a$ percentage points, whereas the relative decrease from a reference score a is $100(a - b)/a$ percent.

Models and memory budgets. Our writer and reader use Qwen3-VL-8B-Instruct (Bai et al., 2025); new bounded textual baselines use its text input interface. The writer reads 5,000-token chunks and carries at most 2,048 text tokens. The answer-time memory-input allowance B applies after construction: prefix baselines keep the first B tokens, stronger textual readouts actively compress to B, and optical methods must satisfy the processed visual-token cap. Question, skill, template, and image-delimiter tokens are reported separately in total reader input; answer tokens are outputs. A nominal cap is not treated as realized until verified with the actual tokenizer or processor. Table 5 summarizes the model and optimization settings.

Table 5: **Model-training and evaluation settings.** These settings describe our optical-memory model. Table 11 uses a separate RULER-HQA protocol.

Setting	Value
Writer / reader backbone	Qwen3-VL-8B-Instruct
Bounded text-baseline backbone	Qwen3-VL-8B-Instruct (text-only)
Training dataset / context	HotpotQA / approximately 30K tokens
Chunk size / recurrent state cap	5,000 / 2,048 text tokens
Maximum answer length	2,048 tokens
GRPO candidates / clipping range	16 / 0.20
KL coefficient / learning rate	10^{-3} / 10^{-6}
Global / micro batch size	64 / 16
Warmup	20 steps
Rollout temperature / top-p	1.0 / 0.999
Evaluation temperature / top-p	0.7 / 0.95
Evaluation instances / runs	128 per dataset / 3
Answer-time memory budgets	16 / 64 / 256 / 1024 tokens

Comparison groups. The bounded-memory comparison includes Mem0, Mem-α, and MemAgent prefix readouts, two budget-aware MemAgent readouts, MemOCR, and R&R (Chhikara et al., 2025; Wang et al., 2025b; Yu et al., 2026; Pan et al., 2024; Shi et al., 2026). Direct-context and RULER-HQA references retain their separate checkpoints and access limits in Appendix G. Equal memory-token allowances do not imply equal computation: compression and optical encoding add cost, and recurrent writing still processes every chunk.

Budget realization. We count the processor's final image grid, not the source resolution: upscaling a small image can increase token use without recovering legibility. Optical variants use the same renderer and processor-level accounting.

Objective and revision controls. The independent weights λ_W and λ_R control the writer and reader auxiliary terms. Validation weight λ_{val} stays fixed across ablations. Probe limits, prefix sampling, revision interval, skill cap, and acceptance tolerance are fixed within a run and selected without test data. Models are frozen for skill acceptance; accepted rules are frozen during model training. Removal definitions follow Equation 6.

E Additional Experimental Details

This section describes the comparison protocols used to interpret the main results.

Budget-aware textual and optical baselines. MemAgent + budget rewrite conditions the common backbone on the final MemAgent state and the question, with an output cap of B tokens including any termination token. MemAgent + LLMLingua-2 applies the released extractive compressor to the same state (Pan et al., 2024). Both outputs are measured with the reader's tokenizer; the compression protocol includes a fixed policy for enforcing the budget when tokenizers differ. These readouts test whether active compression improves on prefix truncation, without access to discarded source chunks. MemOCR supplies a layout-aware optical baseline (Shi et al., 2026). Its common-backbone evaluation concerns an adaptation of the method, whereas published checkpoint results use the original model and constitute a separate comparison.

Representation and continuation controls. The six conditions in Table 2 share training documents, eligible boundaries, model initialization per seed, writer state and chunk limits, and the fixed initial skill. Textual memory uses a question-aware rewrite with a frozen compressor and fixed decoding; optical memory uses the budgeted renderer. Each readout is constructed once per committed state, and its answer signal supervises training. Multi samples $K \geq 2$ distinct valid branches. Single repeat samples one branch uniformly from the same eligible pool and generates K independent reader rollouts for that branch. In both conditions, the writer commits before branch selection, and every read uses identical text bytes or image pixels. Both auxiliary updates remain active in Single

repeat. Rewards and losses are averaged over the K reads, preserving their scale across conditions. Thus, the contrast varies continuation diversity while holding the number of probe reads fixed.

Training cost. Single repeat and Multi share the number of eligible prefixes, writer candidates, optimization steps, reader calls, decoding caps, and seed set. This controls the number of training calls, although differences in input lengths may still affect computation. Call matching alone does not establish equal training computation, because processed input and output lengths may differ. Final-QA-only omits probe reads and uses less computation per step. Skill revision is disabled in the factorial controls, while full R&R additionally rebuilds and evaluates held-out memories during revision.

Uncertainty and paired comparisons. Accuracy is measured by SEM on the same evaluation instances for every method. Within each seed, the four task scores are averaged to obtain a macro score; uncertainty is the sample standard deviation of these seed-level macro scores. Averaging task-level standard deviations would describe a different quantity. The protocol specifies three independent training seeds for trainable controls, with decoding randomness tracked separately. Paired comparisons use within-seed differences and their dispersion or confidence interval. For fixed pretrained baselines, repeated stochastic decoding measures decoding variability rather than training variability. Run-level scores and seed specifications are needed to supplement the reported means with these uncertainty estimates.

Parameter sharing. Table 3 ablates supervision terms. With shared parameters, a remaining loss can still affect both roles. Removing a loss therefore isolates a source of supervision rather than an independent model module.

Pre-render versus post-render writer feedback. The text-scored control in Table 3 changes the writer's auxiliary reward channel. It uses the same reader checkpoint within each scoring step, continuation sampling, decoding caps, and reward aggregation. At a given scoring call, the writer has already committed to M_t: feedback is computed from either that unrendered state or $\mathrm{Render}_B(M_t)$, without a branch-specific rewrite. Learned memories may diverge across training runs. Reader probe loss and final-QA training remain image-based in both conditions, and skill revision is disabled. The text-scored condition exposes up to L_m text tokens rather than B visual tokens; it tests the role of rendering in feedback, not equal-token computation. Processed training tokens and GPU-hours must therefore be recorded separately for this comparison.

Realized budget and latency. The realized memory budget is the retained text-token count or the visual-token count after preprocessing, including all image crops or tiles. Both its mean and maximum are relevant because preprocessing can raise the token count above the nominal allowance. A condition is infeasible when the processor's minimum exceeds B. Total reader input additionally includes the question, skill, template, and image delimiters. Image dimensions and encoded size characterize the optical representation separately. End-to-end latency covers recurrent writing, compression or rendering and preprocessing, and final answer decoding. The timings describe the measured implementation; they should not be interpreted as hardware-independent complexity estimates.

F Additional Results

Tables 6, 7, 8, and 9 report dataset-level SEM accuracy from the earlier run series at four memory budgets. The R&R rows at 64 and 256 tokens are retained for that series and are not task-level breakdowns of the updated aggregates in Table 1. Corresponding updated task-level scores, the new baselines' breakdowns, and run-level dispersion remain unreported. The earlier 64-token/100K macro has been corrected arithmetically from its displayed task means, as noted in Table 7.

Table 6: **Earlier-run per-task results at 256 answer-time memory tokens.** Mean SEM accuracy (%). Context blocks use the same evaluation instances across methods; Macro is the equal-weight average over four datasets. Prefix denotes truncation.

Method	HotpotQA	2Wiki	NQ	TriviaQA	Macro
Source context: 10K					
Mem0 (prefix)	71.1	57.8	46.9	69.5	61.3
Mem-α (prefix)	57.8	38.3	47.9	17.2	40.3
MemAgent (prefix)	82.3	66.1	51.3	69.5	67.3
R&R (earlier)	82.2	71.2	57.3	79.7	72.6
Source context: 30K					
Mem0 (prefix)	65.6	44.8	51.1	70.6	58.0
Mem-α (prefix)	57.3	32.8	33.6	24.2	37.0
MemAgent (prefix)	76.8	62.5	44.3	77.6	65.3
R&R (earlier)	75.4	72.8	48.8	80.9	69.5
Source context: 100K					
Mem0 (prefix)	63.8	43.5	49.7	62.8	55.0
Mem-α (prefix)	57.0	41.7	38.0	18.3	38.7
MemAgent (prefix)	76.6	58.1	53.6	67.2	63.9
R&R (earlier)	75.7	65.5	58.3	68.9	67.1

Table 7: **Earlier-run per-task results at 64 answer-time memory tokens.** Mean SEM accuracy (%). Context blocks use the same evaluation instances across methods; Macro is the equal-weight average over four datasets. Prefix denotes truncation.

Method	HotpotQA	2Wiki	NQ	TriviaQA	Macro
Source context: 10K					
Mem0 (prefix)	56.7	53.9	46.9	66.4	56.0
Mem-α (prefix)	25.8	34.9	27.1	13.0	25.2
MemAgent (prefix)	50.8	38.5	48.7	64.6	50.7
R&R (earlier)	77.6	62.9	51.0	77.6	67.3
Source context: 30K					
Mem0 (prefix)	60.9	42.2	43.8	65.9	53.2
Mem-α (prefix)	29.7	31.5	25.0	19.5	26.4
MemAgent (prefix)	52.3	42.7	36.5	69.5	50.3
R&R (earlier)	68.1	66.2	43.6	80.7	64.7
Source context: 100K					
Mem0 (prefix)	56.2	43.3	50.0	62.5	53.0
Mem-α (prefix)	28.9	39.1	24.0	13.8	26.4
MemAgent (prefix)	51.0	36.2	47.9	59.6	48.7
R&R (earlier)	67.2	62.4	47.5	70.2	61.8[†]

[†]Recomputed from the four displayed task means: $(67.2 + 62.4 + 47.5 + 70.2)/4 = 61.825$, rounded to 61.8. This earlier-series mean is separate from the updated 62.0 aggregate in Table 1.

Table 8: **Per-task results at a nominal 1024-token memory budget.** Mean SEM accuracy (%). Context blocks use the same evaluation instances across methods; Macro is the equal-weight average over four datasets. Prefix denotes truncation.

Method	HotpotQA	2Wiki	NQ	TriviaQA	Macro
Source context: 10K					
Mem0 (prefix)	70.3	60.4	47.7	69.5	62.0
Mem-α (prefix)	75.0	47.7	45.6	21.1	47.3
MemAgent (prefix)	82.3	65.4	53.4	70.1	67.8
R&R	84.8	72.2	61.8	79.6	74.6
Source context: 30K					
Mem0 (prefix)	66.1	47.7	48.8	70.6	58.3
Mem-α (prefix)	77.9	40.1	44.8	23.2	46.5
MemAgent (prefix)	78.9	63.8	46.1	78.1	66.7
R&R	75.1	73.7	49.2	81.3	69.8
Source context: 100K					
Mem0 (prefix)	64.1	40.9	48.2	61.4	53.6
Mem-α (prefix)	79.4	50.0	43.0	19.3	47.9
MemAgent (prefix)	79.4	61.2	55.5	66.7	65.7
R&R	78.3	62.7	55.4	69.8	66.6

Table 9: **Per-task results at a nominal 16-token memory budget.** Mean SEM accuracy (%). Context blocks use the same evaluation instances across methods; Macro is the equal-weight average over four datasets. Prefix denotes truncation.

Method	HotpotQA	2Wiki	NQ	TriviaQA	Macro
Source context: 10K					
Mem0 (prefix)	37.0	28.4	41.4	65.3	43.0
Mem-α (prefix)	24.0	35.4	23.7	11.7	23.7
MemAgent (prefix)	24.0	28.4	24.5	49.7	31.6
R&R	67.2	57.9	42.8	80.8	62.2
Source context: 30K					
Mem0 (prefix)	43.3	31.5	34.1	62.8	42.9
Mem-α (prefix)	30.2	32.3	26.3	19.0	27.0
MemAgent (prefix)	26.6	26.3	20.1	56.3	32.3
R&R	57.9	56.0	35.9	72.2	55.5
Source context: 100K					
Mem0 (prefix)	34.9	27.6	43.0	58.9	41.1
Mem-α (prefix)	26.3	40.1	24.2	14.3	26.2
MemAgent (prefix)	23.2	16.7	27.3	41.7	27.2
R&R	52.4	45.7	45.9	67.2	52.8

G External Reference Results

Tables 10 and 11 report published results from Shi et al. (2026) and Yu et al. (2026), respectively, to contextualize direct source access and length extrapolation. The former uses different 7B text checkpoints and input caps; the latter follows the external RULER-HQA construction and mixes 7B and 32B models. These evaluations differ from our common-backbone comparison in model, source access, and task construction. They provide context for those settings rather than an estimate of R&R's gains.

Table 10: **Direct-context references.** SEM scores (%) reported in Shi et al. (2026), at source lengths up to 100K. These checkpoints have different input caps and are 7B text models; they read the source directly rather than a budgeted memory. A dash denotes an unavailable result. Best values are bold.

Model	HotpotQA	2Wiki	NQ	TriviaQA	Average
Source context: 10K					
Qwen2.5 (2024)	70.3	53.9	50.8	**71.1**	61.5
R1-Distill Qwen (2025)	50.8	50.8	35.2	54.7	47.9
Qwen2.5-1M (2025)	**75.0**	**67.2**	**53.1**	**71.1**	**66.6**
Source context: 30K					
Qwen2.5 (2024)	–	61.1	–	–	–
R1-Distill Qwen (2025)	12.5	26.6	3.9	11.7	13.7
Qwen2.5-1M (2025)	**66.4**	**65.6**	**43.8**	**77.3**	**63.3**
Source context: 100K					
Qwen2.5 (2024)	–	–	–	–	–
R1-Distill Qwen (2025)	0.8	2.3	1.6	0.8	1.4
Qwen2.5-1M (2025)	**68.8**	**54.7**	**50.8**	**59.4**	**58.4**

Table 11: **RULER-HQA reference comparison** (Yu et al., 2026). SEM scores (%) across source lengths. RAG retrieves eight wiki items; RL-MemAgent retains 1024 tokens. The 32B QwenLong-L1 checkpoint has a 120K input limit. Best values per length are bold; dashes denote unavailable results.

Source length	RAG-7B (2020)	QwenLong-L1-32B (2025)	RL-MemAgent-7B (2026)
7K	75.00	72.66	**81.25**
14K	75.00	75.00	**81.25**
28K	75.78	72.66	**82.03**
56K	74.22	60.94	**80.47**
112K	74.22	31.25	**79.69**
224K	**77.34**	17.19	75.78
448K	72.66	13.28	**76.56**
896K	68.75	11.72	**74.22**
1.75M	64.06	–	**77.34**
3.5M	64.84	–	**71.88**

H Qualitative Memory Example



Figure 5: **Two points of failure at a nominal 16-token budget.** The schematic is organized by memory state, compression, and answer; it does not reproduce actual reader-input pixels. Truncation deletes evidence; uniform rendering obscures it; structured optical memory keeps the relevant relation readable.

Figure 5 illustrates these two failure modes in the nominal 16-token case asking who wrote *Put Your Hand in the Hand*. Textual memory initially contains Gene MacLellan, but final truncation excludes the name. Uniform optical memory retains the words yet makes them unreadable, leading to the answer Greg Brown. Structured memory gives the answer-bearing region enough visual space to preserve the correct name. The no-layout column is a rendering comparison, separate from the training ablations.