MEMORY CONTROL SIGNALS EMERGE BEFORE ACTION IN LONG HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long horizon language model agents continuously accumulate interaction history, increasing computational cost while making relevant information harder to preserve and reuse. Existing context management methods mainly focus on how to compress or retrieve history, but largely leave open whether the model itself already represents the need for these memory operations before they occur. We study the hidden state immediately before each agent action and find that compression and recall needs are already encoded in the model's internal representations. These signals cannot be explained by simple context length or interaction progress, and they exhibit distinct formation patterns across model depth. We further show that most memory decision information is preserved in a compact recent context, while selectively restored historical evidence complements the long range dependencies that recent context misses. Based on these findings, we propose **Preaction Memory with Evidence Retrieval(PaMER)**, which combines state guided compression with external evidence retrieval. PaMER+ further introduces step level evidence selection to recover only the historical information required by the current task. Experiments on WorkBuddyBench, across multiple context management baselines and model backbones, show that our framework substantially reduces context consumption while maintaining competitive task performance.

1 INTRODUCTION

Long horizon language model agents face a direct challenge: interaction history continuously grows, while longer context increases computation, latency, and memory cost and may also reduce the model's ability to use relevant information effectively (Liu et al., 2024; Hsieh et al., 2024; Laban et al., 2026). A common strategy is to compress history when the context approaches its window limit, or to trigger compression periodically according to token count or interaction length. However, context length is not equivalent to compression need. Two agent trajectories with the same number of tokens may represent very different task states. One may be long because the agent has just assembled critical evidence, while another may already contain repeated searches, irrelevant tool outputs, and outdated intermediate results. Recent work therefore begins to let agents manage context according to the current task state rather than relying only on fixed length rules (Li et al., 2026c;b; Liu et al., 2026; Verma, 2026; Wang et al., 2026a).

Compression introduces a complementary problem: information removed from the active context may become important again later. When subsequent reasoning depends on earlier evidence, the agent must recover or retrieve historical information (Hu et al., 2026b; Zhang et al., 2026b; Li et al., 2026d;b; Yu et al., 2026; Sun et al., 2026a; Maharana et al., 2024). Long horizon memory management therefore involves two dynamic decisions: whether the current history should be compressed, and whether earlier information is needed again. Existing methods implement such operations through rules, prompting, additional controllers, or dedicated training (Kang et al., 2025; Li et al., 2026c;b; Hu et al., 2026b; Yan et al., 2026; Shen et al., 2026; Xiong et al., 2026; Huang et al., 2026). This leaves a more fundamental issue unresolved: whether the language model already represents its memory need internally before compression or historical recall is performed.

We study the hidden state immediately before the model produces its next action at each agent decision point. Our hypothesis is that this preaction hidden state has already integrated the task objective, recent evidence, and historical dependencies, and can therefore reveal whether the current

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.

history should be compressed or whether earlier evidence is needed again. The final preaction hidden state predicts compression and recall with AUROC values of **0.831** and **0.765**, respectively, substantially outperforming observable controls based on context length, turn count, and tool type. More importantly, we further investigate **how this information forms inside the model**. Layerwise analysis shows that compression information generally becomes stronger with model depth, whereas recall information emerges earlier and reaches its highest measured performance at an intermediate layer. The representation at the last input token also consistently outperforms mean pooling. These results suggest that the model contains more than a simple signal that the context has become long. Instead, memory needs are reflected in internal representations that depend on the current task state and develop differently during model computation.

We next study how much history is required to preserve this decision information. Retaining only the task prefix and the two most recent complete interaction blocks uses approximately 26% of the Full Context input, yet preserves most of the predictive performance for both compression and recall. This indicates that the recent working state already contains most of the dynamic information required for memory decisions. Recent context alone, however, is not sufficient. When long range dependencies become relevant again, restoring a small amount of selected raw historical evidence recovers additional predictive information and outperforms the hidden state memory alternatives that we evaluate. This leads to a natural design principle: recent context describes the current working state, while external history restores the evidence that is currently missing.

Based on these findings, we propose **PaMER**, Preaction Memory with Evidence Retrieval. PaMER uses the preaction representation extracted by a shared frozen Qwen3.5-9B model to decide when to compress, removes older history from the active context, and stores it in retrievable external memory for later task-specific evidence reuse. PaMER+ further introduces step level evidence selection so that only the raw historical steps required by the current task are restored.

Our contributions are threefold. **First**, we systematically study whether compression and recall needs are already present in the model's internal representations before the corresponding memory operations occur. We show that preaction hidden states contain memory decision information that cannot be explained by simple context length or interaction progress, and we further characterize how these signals form across model depth. **Second**, we investigate which parts of history support these decisions. Approximately 26% of Full Context is sufficient to preserve most predictive performance, while selective historical evidence recovers long range dependencies that are missing from recent context. **Third**, we translate these findings into PaMER and PaMER+, and evaluate them on WorkBuddyBench (Team et al., 2026) against multiple context management baselines and across several model backbones. The resulting system substantially reduces context consumption while maintaining competitive task performance.

2 Related Work

2.1 Context Compression for Long Horizon Agents

Long context processing has motivated a broad class of methods that reduce the amount of history presented to a language model while preserving task relevant information. LLMLingua 2 learns token importance through distilled supervision, while LongLLMLingua extends prompt compression to long context settings (Pan et al., 2024; Jiang et al., 2024). As language models are increasingly used as agents, recent work has moved from static prompt compression to the management of continuously growing interaction histories. ACON (Kang et al., 2025) optimizes compression for long horizon agent trajectories, SWE Pruner (Wang et al., 2026b) adaptively removes redundant context in coding agents, and PACE (Wei et al., 2026) selects historical information according to its predicted relevance to future actions.

Recent studies further suggest that compression timing should depend on the current task state rather than only on a fixed context threshold. Self Compact allows an agent to decide when to compact its trajectory, Self GC continuously reorganizes context during interaction, and Sculptor provides explicit operations for summarizing, hiding, restoring, and searching information (Li et al., 2026c; Hao et al., 2026; Li et al., 2026b). State Proprioception similarly exposes runtime information to support adaptive context management (Xu et al., 2026b). These methods show that context length alone does not fully characterize when memory intervention is useful. Our work studies an earlier

question in this process. Rather than first providing an explicit context state or training a new memory policy, we ask whether the model's own hidden representation already contains information about the need to compress.

2.2 Memory Retrieval and Historical Evidence

Removing information from active context creates a complementary problem: historical evidence may become useful again later (Ai et al., 2026; Zhu et al., 2026; Banerjee et al., 2026; Cao et al., 2026; Ge et al., 2025). MemGPT organizes information between active and external memory, SAM adapts memory access to the current reasoning state, CoMem introduces a separate long context component for context management, and ACM studies agentic context management for long horizon tasks (Packer et al., 2023; Hu et al., 2026b; Zhang et al., 2026b; Li et al., 2026d). Learning What Not to Forget further studies lightweight mechanisms for deciding which parts of long agent histories should remain available (Jahan Lia & Mazumder, 2026; Zhang et al., 2026a; Sun et al., 2026b).

Retrieval itself has also been studied as a dynamic decision. FLARE performs retrieval according to signals observed during generation, while Self RAG learns when to retrieve and how to use retrieved evidence through explicit reflection (Jiang et al., 2023; Asai et al., 2024; Li et al., 2026a; Hu et al., 2026a). Most closely related to our analysis, Probing RAG uses internal representations to determine whether additional document retrieval is necessary (Baek et al., 2025). Our setting differs in both the target and the interaction structure. Probing RAG studies selective retrieval of external documents for question answering, whereas we study two memory needs inside long agent trajectories: whether accumulated interaction should be compressed and whether evidence that previously left the active context is needed again. We further examine how these two signals develop across model depth and how much of the trajectory history is required to preserve them.

2.3 Probing Internal Representations

Our analysis is also related to work that uses probes to study information encoded in neural representations. Linear probes provide a simple way to measure whether a target property can be decoded from intermediate model states (Alain & Bengio, 2016). At the same time, prior work emphasizes that probe accuracy should be interpreted carefully and does not by itself identify the causal mechanism used by the underlying model (Hewitt & Liang, 2019; Belinkov, 2022).

Related studies have shown that hidden representations can expose information that is not directly available from model outputs. Such signals include latent knowledge (Burns et al., 2022), confidence about whether an answer is known (Kadavath et al., 2022; Mahaut et al., 2024), and information related to truthfulness (Azaria & Mitchell, 2023). We use probing for a more specific systems question: whether the model state immediately before an agent action already contains information about an upcoming memory need. Accordingly, our analysis measures predictive information in preaction states. We compare hidden states with observable metadata and surface text controls, analyze how compression and recall signals evolve across layers, and study how these signals change when most historical context is removed. These findings then provide the empirical basis for the design of our online memory controller for long horizon agent interactions.

3 Understanding Memory Needs in Long Horizon Agents

We study whether compression and recall needs are represented internally before the model produces its next action. Our analysis defines these memory decisions from long agent trajectories, extracts the corresponding preaction hidden states, and measures how well the decisions can be read from different model layers across successive stages of computation.

We first compare hidden states with observable controls, then examine how the signal changes across layers. Finally, we vary the available history to test how much decision information remains under a smaller context budget and after historical evidence is restored.

3.1 Memory Decision Data and Probing

We collect 72,912 decision points from 2,521 long agent trajectories covering 600 questions. The source questions do not overlap with WorkBuddyBench and come from different task domains. An LLM annotator identifies states where compression or recall is needed. The annotation procedure and the use of multiple labeled points within one trajectory are described in Appendix A.1.

At decision step t, let o_t denote the context available before the next action. For trajectory τ, the annotator identifies compression and recall decision sets $\mathcal{A}^{\mathrm{c}}(\tau)$ and $\mathcal{A}^{\mathrm{r}}(\tau)$. We define

$$y_t^m = \mathbf{1}[t \in \mathcal{A}^m(\tau)], \qquad m \in \{\mathrm{c}, \mathrm{r}\}, \tag{1}$$

where c denotes compression and r denotes recall. Compression indicates that older context should be condensed while preserving task relevant information. Recall indicates that the current reasoning state needs earlier evidence that is no longer available in the active context. Recall is evaluated only when valid external memory exists. A trajectory may contain multiple positive decision points because the same information need can arise again during a long interaction. This formulation allows memory needs to be evaluated at the decision level while preserving their dependence on the surrounding trajectory context.

For each decision point, we extract the model state immediately before the next action. Let $\phi^{(L)}(o_t)$ denote the final hidden sequence after the output RMSNorm. Our main representation is

$$\mathbf{h}_t = \phi^{(L)}(o_t)[-1] \in \mathbb{R}^d, \qquad d = 4096. \tag{2}$$

The input uses the same conversation format and tool descriptions as the corresponding agent state. Neither the annotation nor the future action is included when extracting $\mathbf{h}_t$.

We train an independent readout for each memory decision. The linear probe predicts

$$p_t^m = \sigma\big((\mathbf{w}^m)^\top \mathbf{h}_t + b^m\big), \qquad m \in \{\mathrm{c}, \mathrm{r}\}. \tag{3}$$

We additionally evaluate a one hidden layer MLP to test whether nonlinear readout provides additional predictive power. The language model itself is not updated during probe training. This separation tests what information is already accessible from the pretrained representation while keeping the underlying language model fixed.

Because both targets are imbalanced, we optimize class weighted binary cross entropy,

$$\mathcal{L}_m = -\frac{1}{N_m} \sum_{i=1}^{N_m} \left[\rho_m y_i^m \log p_i^m + (1 - y_i^m) \log(1 - p_i^m)\right], \tag{4}$$

where $\rho_m = N_m^- / N_m^+$ is computed from the training split only. All states from the same question remain in the same split. We use 70/15/15 training, validation, and test splits over five seeds. Decision thresholds are selected by validation F1 and then fixed for testing. AUROC is the primary metric. AUPRC, thresholded metrics, and the complete MLP definition are reported in Appendix A.2.

This evaluation protocol uses each question as the grouping unit, keeping its decision states together when partitioning the data. The test split therefore contains questions excluded from probe fitting, and performance is measured across held-out questions. This setting reflects the deployment requirement of predicting memory needs for unseen tasks while retaining the dependencies among states that originate from the same question or trajectory. The five split seeds vary question allocation and readout initialization.

To characterize how memory decision information develops across model depth, we train an independent linear probe at each evaluated layer. At layer ℓ, we compare the representation at the final input position with the mean representation over all input tokens:

$$\mathbf{h}_{t,\mathrm{last}}^{(\ell)} = \phi^{(\ell)}(o_t)[-1], \qquad \mathbf{h}_{t,\mathrm{mean}}^{(\ell)} = \frac{1}{n_t} \sum_{j=1}^{n_t} \phi^{(\ell)}(o_t)[j]. \tag{5}$$

For each layer, target, and pooling choice, we also measure the difference between the mean positive and negative probe logits,

$$\Delta_{\ell,\alpha}^m = \mathbb{E}\big[z_{t,\ell,\alpha}^m \mid y_t^m = 1\big] - \mathbb{E}\big[z_{t,\ell,\alpha}^m \mid y_t^m = 0\big], \tag{6}$$

where $\alpha \in \{\mathrm{last}, \mathrm{mean}\}$. AUROC remains the primary metric because the magnitude of the logit gap also depends on probe scale.

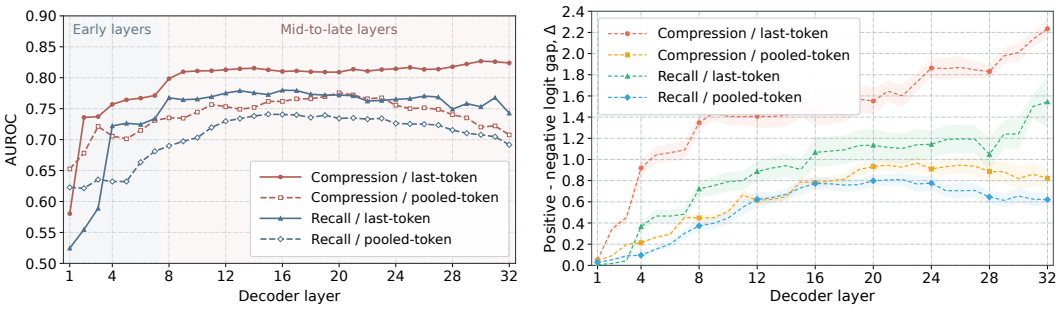


Figure 1: **Memory-decision signals across layers.** (a) AUROC of independently trained probes using the final input token or mean-pooled states. (b) Mean positive-minus-negative probe logit. Results are aggregated over five question-grouped splits.

3.2 Memory Signals beyond Observable Context

A simple explanation for memory decisions is that the agent compresses because the context is long, or recalls because the interaction has lasted many turns. Table 1 shows that these observable signals are informative but insufficient. Metadata reaches 0.758 AUROC for compression and 0.549 for recall, whereas the final hidden state reaches 0.831 and 0.765, respectively. A nonlinear MLP provides almost no additional gain, suggesting that much of this information is already accessible through a linear readout.

Table 1: Memory decision signals. Mean AUROC.

Signal	Comp. ↑	Recall ↑
Length / Turn	0.751	0.552
Metadata	0.758	0.549
Hidden (Linear)	**0.831**	**0.765**
Hidden (MLP)	0.827	0.763

These results show that preaction hidden states contain predictive information beyond the metadata controls, supporting state-based prediction of compression and recall needs before an action. Text controls, uncertainty estimates, and robustness results are reported in Appendix A.

3.3 How Memory Signals Form across Depth

Figure 1 shows that compression prediction becomes stronger with model depth. The earliest measured representation provides only a weak signal, with AUROC around 0.57, whereas the final normalized state reaches 0.831. Recall follows a different pattern and reaches its highest measured mean AUROC of 0.780 at block 16, compared with 0.765 at the final state. The two memory needs therefore develop differently across the network.

The final input position is consistently more informative than mean pooling. At the final state, last-token and mean-pooled AUROCs are 0.831 and 0.712 for compression, and 0.765 and 0.696 for recall. This suggests that information relevant to the next memory action is more accessible near the position used to produce that action than in an average over the full input.

Although recall peaks at block 16, we use the final state for the online controller, whose learned decision is compression. Compression improves from 0.810 at block 16 to 0.831 at the final state. The recall advantage at block 16 is 0.015, while the reported variation across splits is larger than this difference. Across the cross-model evaluations, a shared frozen Qwen3.5-9B feature extractor produces this preaction representation from the task prefix and Recent2. The task-executing model can change while the controller continues to operate in the same representation space. We therefore align the readout with the online compression target and keep its representation interface fixed across model evaluations. Exact layer-level results are reported in Appendix Table 9.

3.4 What Historical Information Is Needed

We next reduce the amount of trajectory history available to the model while keeping the prediction targets unchanged. Let P_t denote the system and task prefix and let $\mathcal{R}_t$ contain the two most recent complete interaction blocks. The Recent2 condition uses

$$C_t^{\text{recent}} = P_t \oplus \mathcal{R}_t. \tag{7}$$

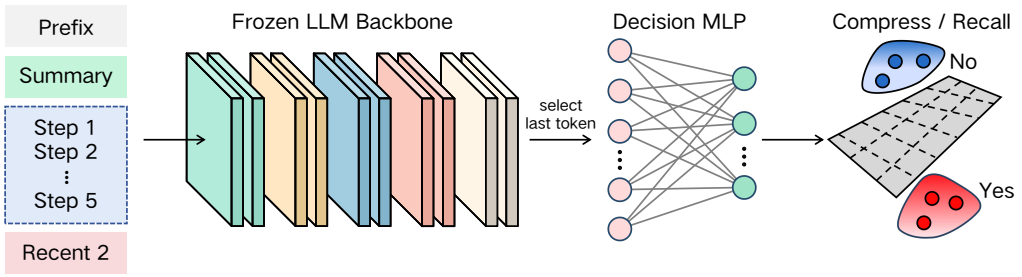


Figure 2: **Reading memory needs before an action.** The diagram illustrates compression and recall as prediction targets. In the deployed controller, the compression heads read the final state from the task prefix and Recent2. The recall probe supports offline analysis; online retrieval operates independently of its predictions.

This construction preserves the task objective together with the current working state while removing most earlier interaction. Recent2 uses only 26.0% of the Full Context token budget, yet retains compression and recall AUROCs of 0.776 and 0.761, compared with 0.829 and 0.780 under Full Context. Decision distillation keeps the same context budget. It changes compression AUROC only from 0.776 to 0.781, but increases compression F1 from 0.541 to 0.598, suggesting that Recent2 already contains much of the required information while the Full Context teacher mainly improves the thresholded decision.

The remaining gap shows that recency alone is not sufficient. Some dependencies needed for a memory decision lie outside the recent window. If $\mathcal{E}_t$ denotes recovered historical evidence, the reconstructed context is

$$C_t^{\text{recover}} = P_t \oplus \mathcal{E}_t \oplus \mathcal{R}_t. \tag{8}$$

The task and system prefix is retained in every condition, so all comparisons preserve the goal against which relevance is judged. The prefix-only Current condition is reported in Appendix B. These comparisons vary historical context while retaining the same system and task prefix.

Table 2: Compact context and historical evidence. Tokens are relative to Full Context.

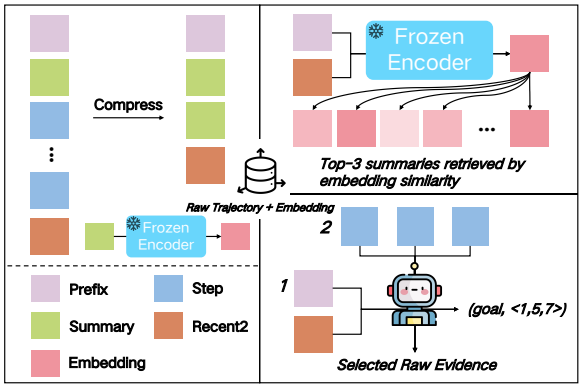
History	Tokens ↓	Comp. ↑	Recall ↑
Full Context	100.0	0.829	**0.780**
Compact context			
Recent2	26.0	0.776	0.761
+ Decision KD	26.0	0.781	0.753
Recovered history			
Hidden Bank	16.8	0.769	0.730
Selective Raw	66.2	**0.832**	0.769
Qwen Retrieval	53.9	0.804	0.764

Table 2 shows that a short recent history preserves most of the recall signal and a large fraction of the compression signal. However, the form of older information also matters. Hidden Bank Recent5 uses the smallest token budget in the table but loses more decision information. Restoring selected raw evidence is more effective in this setting: Selective Raw Top5 reaches 0.832 compression AUROC and 0.769 recall AUROC. Qwen Top3 provides a practical embedding-based alternative, reaching 0.804 and 0.764 with 53.9% of the Full Context token budget, with a moderate token budget.

Taken together, the results suggest three complementary sources of information. The task prefix establishes what is relevant, recent interaction describes the current working state, and selected historical evidence restores dependencies that have left the recent window. This organization of task information, recent interaction, and historical evidence provides the empirical design pattern used by PAMER in the next section. Full context sweeps and retrieval controls are reported in Appendix B.

4 PAMER: PREACTION MEMORY WITH EVIDENCE RETRIEVAL

PAMER uses the current model state to decide when to compress history and an external memory to retain information for later access. PAMER$^+$ adds selection at the level of individual historical steps. The design follows the preceding analysis: a short recent context can support compression decisions, while selected older evidence remains available when information has left the active history.



Variant	Reward ↑	Tokens ↓	Peak Ctx. ↓
Base agent	68.68	3.066M	280.1K
PaMER	**70.00**	**0.899M**	**97.5K**
w/o Recall	63.70	0.924M	107.1K
w/o Compression	68.87	3.180M	253.3K

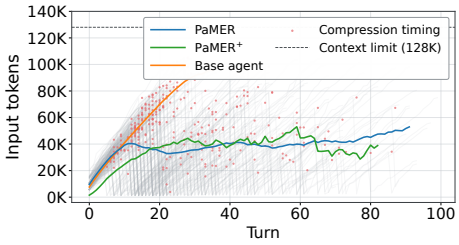


Figure 3: **Online behavior and component comparisons.** (a) Compression and evidence recall. (b) Component results. (c) Input context across interaction turns. The horizontal line marks 128K as a context-growth analysis reference; controller rules operate independently of this reference value.

4.1 State Guided Compression and Retrieval

PAMER retains the system and task prefix P_t and the two most recent complete interaction blocks, $\mathcal{R}_t$. A block contains an assistant message and its subsequent tool results and other messages, up to the next assistant message. Earlier complete blocks form the compression candidate. Operating on complete blocks preserves the assistant and tool-call structure.

For a candidate of at least 1,024 tokens, five independently trained compression heads read the final state from $P_t \oplus \mathcal{R}_t$. Compression is triggered when at least three heads exceed their validation-selected thresholds. The heads are trained on Recent2 using both the compression annotation and the corresponding Full Context teacher prediction:

$$\mathcal{L}_s = \mathrm{WBCE}(z_{t,s}, y_t^{\mathrm{c}}) + 0.5\,\mathrm{BCEWithLogits}(z_{t,s}, p_{t,s}^{\mathrm{full}}). \tag{9}$$

Here s indexes the independently trained head. The Full Context teacher is used during training, not during online decisions.

When compression is triggered, we summarize the older blocks. Each summary is encoded with Qwen3-Embedding-8B and stored with links to the covered history. The raw blocks are removed only after storage succeeds. The current request then contains

$$C_t' = P_t \oplus \mathcal{R}_t. \tag{10}$$

The new summary is not inserted into this same request, and no recall is performed on it. This prevents immediate retrieval from undoing the context reduction.

On a later request with available memory, the task and Recent2 form an embedding query. The controller retrieves the Top3 summaries by cosine similarity and inserts them between the prefix and recent blocks. Online retrieval operates independently of the learned recall probe. The probe characterizes recall signals in the offline analysis, while retrieval uses memory available during agent execution. If a memory operation fails, the controller keeps the original history rather than applying an incomplete update. Implementation details are in Appendix C.

4.2 PAMER+: Step Level Evidence Selection

A summary can cover many steps even when the current request needs only a small part of them. PAMER+ records `step_details` for each summary, linking its contents to the original historical steps. A selected summary identifies candidate memory, while the step selector determines which of its historical blocks are restored. The selector specifies `raw_steps_needed`, and only those complete historical blocks enter the recall path. The summary therefore helps locate the relevant memory, while selected raw steps provide the supporting evidence for the current reasoning state. The selected steps must belong to the chosen memory, and invalid selections are rejected. Detailed validation, cache reuse, and fallback rules are given in Appendix C.3.

5 EXPERIMENTS

We evaluate task performance and token consumption on the full WorkBuddyBench benchmark, compare context-management methods on Pilot40, and examine results across several model backbones. We then inspect context growth and the separate roles of compression and recall.

5.1 EXPERIMENTAL SETUP

WorkBuddyBench contains 260 tasks: 80 Code, 50 Office, 60 Security, and 70 Web. Pilot40 contains 10 fixed tasks from each domain. Methods within a comparison use the same task set. The memory-decision corpus is collected separately from these evaluation tasks.

We report mean task reward multiplied by 100. Overall scores average over tasks, rather than giving equal weight to domains of different sizes. Tokens denote mean total input and output tokens per task, including the main agent and auxiliary memory calls. Cache tokens are not added separately because they are already included in input usage. Detailed evaluation protocols, token accounting, and system results are provided in Appendix D.

Table 3: **Full benchmark system performance and token usage.** Scores are multiplied by 100. Green and red indicate favorable and unfavorable changes from the base system.

Model	Code	Office	Sec.	Web	Avg.	Tokens ↓
DeepSeek-V4-Flash	**77.0**	**81.8**	47.8	72.1	69.9	2.69M
+ PAMER	75.3 (-1.7)	79.8 (-2.0)	**52.3** (+4.5)	70.0 (-2.1)	69.4 (-0.4)	**0.72M** (-73.2%)
+ PAMER$^+$	76.7 (-0.3)	80.9 (-0.9)	51.6 (+3.8)	**76.3** (+4.2)	**71.6** (+1.7)	0.76M (-71.7%)

Table 4: **Comparison with context management methods on WorkBuddyBench Pilot40.** All methods use the same fixed 40 tasks. Scores are multiplied by 100. Parenthesized values show changes from the DeepSeek-V4-Flash base agent.

Method	Code	Office	Sec.	Web	Avg.	Tokens ↓
Base and Ours						
DeepSeek-V4-Flash (Xu et al., 2026a)	72.9	81.6	44.6	71.0	67.5	1.89M
+ PAMER	76.3 (+3.4)	79.9 (-1.7)	43.9 (-0.7)	72.0 (+1.0)	68.0 (+0.5)	0.90M (-52.4%)
+ PAMER$^+$	71.3 (-1.6)	**86.6** (+5.0)	47.5 (+2.9)	70.0 (-1.0)	68.9 (+1.3)	0.98M (-48.1%)
Simple Context Baselines						
Sliding Window ($K = 5$)	66.8 (-6.1)	67.4 (-14.2)	32.3 (-12.3)	59.0 (-12.0)	56.4 (-11.2)	2.95M (+56.3%)
Sliding Window ($K = 10$)	74.4 (+1.4)	73.1 (-8.6)	40.1 (-4.5)	61.0 (-10.0)	62.1 (-5.4)	1.98M (+5.1%)
Sliding Window ($K = 20$)	79.3 (+6.4)	85.1 (+3.5)	40.3 (-4.3)	70.0 (-1.0)	68.7 (+1.1)	2.02M (+7.0%)
Periodic Summary ($n = 3$)	65.4 (-7.5)	67.7 (-14.0)	26.3 (-18.3)	**82.7** (+11.7)	60.5 (-7.0)	2.21M (+17.2%)
Periodic Summary ($n = 5$)	**83.6** (+10.7)	75.7 (-5.9)	40.0 (-4.5)	81.3 (+10.3)	**70.2** (+2.6)	2.37M (+25.5%)
API-based Methods						
PACE (Wei et al., 2026)	70.4 (-2.6)	55.8 (-25.9)	43.5 (-1.0)	64.0 (-7.0)	58.4 (-9.1)	1.10M (-41.6%)
LLMLingua-2 (Pan et al., 2024)	64.6 (-8.3)	73.3 (-8.3)	40.1 (-4.5)	70.0 (-1.0)	62.0 (-5.5)	2.40M (+27.3%)
SelfCompact (Li et al., 2026c)	76.8 (+3.9)	79.6 (-2.0)	36.2 (-8.4)	71.0 (0.0)	65.9 (-1.6)	1.57M (-17.0%)
ACON-Core (Kang et al., 2025)	71.8 (-1.1)	68.6 (-13.1)	**47.9** (+3.4)	65.0 (-6.0)	63.3 (-4.2)	1.40M (-25.7%)
Self-GC (Hao et al., 2026)	63.5 (-9.5)	71.7 (-10.0)	43.7 (-0.9)	71.0 (0.0)	62.5 (-5.1)	1.66M (-12.4%)
LRE (Jahan Lia & Mazumder, 2026)	62.6 (-10.3)	63.4 (-18.2)	27.1 (-17.4)	68.0 (-3.0)	55.3 (-12.2)	2.29M (+21.3%)
CoMem (Zhang et al., 2026b)	56.4 (-16.5)	59.9 (-21.8)	2.5 (-42.1)	58.0 (-13.0)	44.2 (-23.3)	**0.77M** (-59.5%)
SAM (Hu et al., 2026b)	67.6 (-5.3)	81.6 (0.0)	47.2 (+2.7)	67.0 (-4.0)	65.9 (-1.7)	1.35M (-28.7%)
SWE-Pruner (Wang et al., 2026b)	63.5 (-9.5)	72.6 (-9.0)	40.9 (-3.7)	65.0 (-6.0)	60.5 (-7.1)	1.50M (-20.5%)
Sculptor (Li et al., 2026b)	51.2 (-21.7)	81.6 (0.0)	35.2 (-9.3)	69.0 (-2.0)	59.3 (-8.3)	1.29M (-31.5%)
Released-policy Method						
ACM (Li et al., 2026d)	38.7 (-34.2)	55.9 (-25.7)	6.2 (-38.4)	34.0 (-37.0)	33.7 (-33.8)	2.36M

5.2 MAIN RESULTS

Full benchmark. Table 3 shows the performance and token tradeoff on all 260 tasks. PAMER reduces token usage while retaining a similar overall score. PAMER$^+$ improves the overall score at a slightly higher token budget than PAMER. Its domain-level gains occur in Security and Web, while Code and Office scores remain slightly below the base system.

Table 5: **Cross-model evaluation on WorkBuddyBench Pilot40.** Scores are multiplied by 100. All models are evaluated on the same fixed 40 tasks, with 10 tasks per domain. Parenthesized values show changes from the corresponding base agent. Bold indicates the best overall score and lowest token consumption within each backbone.

Model	Code	Office	Sec.	Web	Avg.	Tokens ↓
GPT-5.6-Luna	44.8	72.1	55.7	56.0	**57.2**	676K
+ PAMER	54.4 (+9.6)	57.8 (-14.3)	52.2 (-3.5)	55.0 (-1.0)	54.9 (-2.3)	**81K** (-88.0%)
+ PAMER$^+$	41.4 (-3.4)	57.7 (-14.4)	55.3 (-0.4)	66.0 (+10.0)	55.1 (-2.1)	97K (-85.7%)
GLM-5.3-Flash (Zeng et al., 2026)	43.2	85.0	66.1	61.7	64.0	762K
+ PAMER	44.4 (+1.2)	86.5 (+1.5)	70.7 (+4.6)	66.7 (+5.0)	**67.1** (+3.1)	318K (-58.3%)
+ PAMER$^+$	39.6 (-3.6)	84.4 (-0.6)	70.7 (+4.6)	60.0 (-1.7)	63.7 (-0.3)	**310K** (-59.3%)
HY-3	49.8	80.6	59.5	46.0	59.0	1441K
+ PAMER	53.6 (+3.8)	64.8 (-15.8)	64.2 (+4.7)	51.0 (+5.0)	58.4 (-0.6)	420K (-70.9%)
+ PAMER$^+$	43.6 (-6.2)	80.0 (-0.6)	67.7 (+8.2)	55.0 (+9.0)	**61.6** (+2.6)	**348K** (-75.9%)
MiMo-V2.5 (mim, 2026)	32.4	78.4	45.9	55.0	52.9	1114K
+ PAMER	44.8 (+12.4)	65.9 (-12.5)	75.0 (+29.1)	61.0 (+6.0)	61.7 (+8.8)	**276K** (-75.2%)
+ PAMER$^+$	59.0 (+26.6)	71.0 (-7.4)	75.0 (+29.1)	70.0 (+15.0)	**68.8** (+15.8)	282K (-74.7%)
Qwen3.8-Flash	76.5	85.0	62.9	85.0	77.4	1207K
+ PAMER	77.6 (+1.1)	74.8 (-10.2)	79.8 (+16.9)	80.0 (-5.0)	78.1 (+0.7)	**869K** (-28.0%)
+ PAMER$^+$	80.1 (+3.6)	83.9 (-1.1)	72.6 (+9.7)	100.0 (+15.0)	**84.2** (+6.8)	874K (-27.6%)

5.3 Baselines and Cross Model Evaluation

Context-management baselines. Table 4 compares simple controls and published methods on Pilot40. Periodic Summary ($n = 5$) achieves a higher average score with more tokens, while CoMem consumes fewer tokens at a substantially lower score. PAMER and PAMER$^+$ improve the base agent's average score while reducing token usage, combining task performance and efficiency on the same Pilot40 tasks.

Different model backbones. Table 5 shows token reductions across all evaluated backbones. PAMER$^+$ improves the scores of MiMo-V2.5 and Qwen3.8-Flash, while GPT-5.6-Luna has lower scores under both variants. The same frozen Qwen3.5-9B feature extractor supports memory control across these task-executing backbones, with performance gains varying by model. Additional cross-model usage details are reported in Appendix D.3.

5.4 Ablation and Online Behavior

Average token consumption does not reveal how context changes within a long interaction. Figure 3 combines the memory operations, component results, and context trajectories. Compression repeatedly reduces the active history, after which context grows as new interaction occurs. Its benefit can therefore persist across later requests that would otherwise resend the removed history.

The component ablation shows that removing compression raises token usage, whereas removing recall lowers task reward. These results support complementary roles for the two components: compression controls accumulated history, and recall restores information that has left the active context. Appendix E.2 gives a trajectory. The component batch is separate from the Pilot40 baseline comparison. Additional figure interpretation and reproducibility details are provided in Appendix E.

6 Conclusion

Memory needs can be predicted from language model states before the next action. Hidden representations capture information beyond context length and interaction metadata, with compression and recall showing distinct depth patterns. Recent context preserves much of this signal, while selected historical evidence restores missing information. These findings motivate PAMER and PAMER$^+$, which combine state guided compression with external memory access. Both reduce context consumption, and PAMER$^+$ improves the overall full-benchmark score. Our results connect memory-need prediction with practical context management for long horizon agents.

AI USE STATEMENT

Generative AI tools were used for language editing and literature retrieval and discovery, including identifying potentially relevant papers, search keywords, and related work, as well as surveying the current state of research on relevant topics. An LLM was also used as an annotator to identify compression and recall needs in the memory-decision corpus, as described in the paper and appendix. The annotation procedure and resulting labels were reviewed by the authors. The final selection, interpretation, and presentation of the literature and experimental results were determined by the authors. All AI-assisted content was reviewed by the authors, who take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the data construction, question-grouped train/validation/test splits, probe architectures and evaluation protocols, context construction, retrieval procedures, controller configuration, token accounting, and benchmark task sets in the main paper and appendices. Hyperparameters and implementation-specific settings for PAMER and PAMER$^+$ are documented alongside the fixed Pilot40 task manifest and definitions of the evaluation metrics. These descriptions specify the data partitions, model configurations, controller operations, and accounting conventions used in the analyses and system evaluations.

REFERENCES

Mimo-v2.5. https://huggingface.co/collections/XiaomiMiMo/mimo-v25, 2026.

Qiuyuan Ai, Zenghuang Fu, Zhaoyang Li, Ping Jiang, Haoyu Wu, Jie Song, and Guannan He. Cognitive scaffold: From fluid context to crystallized memory for long-horizon deepresearch agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25526–25542, 2026.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avi Sil, and Hannaneh Hajishirzi. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *International conference on learning representations*, volume 2024, pp. 9112–9141, 2024.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Ingeol Baek, Hwan Chang, Byeongjeong Kim, Jimin Lee, and Hwanhee Lee. Probing-rag: Self-probing to guide language models in selective document retrieval. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3287–3304, 2025.

Pratyay Banerjee, Masud Moshtaghi, Shivashankar Subramanian, Amita Misra, and Ankit Chadha. Apex-mem: Agentic semi-structured memory with temporal reasoning for long-term conversational ai. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16470–16489, 2026.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. *arXiv preprint arXiv:2212.03827*, 2022.

Shuqi Cao, Jingyi He, and Fei Tan. Higmem: A hierarchical and llm-guided memory system for long-term conversational agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 33853–33862, 2026.

Yubin Ge, Salvatore Romeo, Jason Cai, Raphael Shu, Yassine Benajiba, Monica Sunkara, and Yi Zhang. Tremu: Towards neuro-symbolic temporal reasoning for llm-agents with memory in multi-session dialogues. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 18974–18988, 2025.

Xubin Hao, Hongjin Meng, Xin Yin, Jiawei Zhu, and Chenpeng Cao. Self-gc: Self-governing context for long-horizon llm agents. *arXiv preprint arXiv:2607.00692*, 2026.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

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 preprint arXiv:2404.06654*, 2024.

Xuxian Hu, Zhu Teng, Wei Zhang, Ming He, and Jianping Fan. Querylink: Leveraging query-memory alignment for long-term reasoning in llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 15608–15621, 2026a.

Yuyang Hu, Hongjin Qian, Shuting Wang, Jiongnan Liu, Ziliang Zhao, Jiejun Tan, Zheng Liu, and Zhicheng Dou. Sam: State-adaptive memory for long-horizon reasoning agent. *arXiv preprint arXiv:2605.24468*, 2026b.

Weiquan Huang, Zixuan Wang, Hehai Lin, Sudong Wang, Bo Xu, Qian Li, Beier Zhu, Linyi Yang, and Chengwei Qin. Ama: Adaptive memory via multi-agent collaboration. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 3099–3120, 2026.

Nusrat Jahan Lia and Aritra Mazumder. Learning what not to forget: Long-horizon agent memory from a few kilobytes of learning. *arXiv e-prints*, pp. arXiv–2606, 2026.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, 2024.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7969–7992, 2023.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin A Inan, Lukas Wutschitz, Yanzhi Chen, Robert Sim, and Saravan Rajmohan. Acon: Optimizing context compression for long-horizon llm agents. *arXiv preprint arXiv:2510.00615*, 2025.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation. In *International Conference on Learning Representations*, volume 2026, pp. 54738–54778, 2026.

Jinze Li, Yang Zhang, Xin Yang, Jiayi Qu, Jinfeng Xu, Shuo Yang, Junhua Ding, and Edith Cheuk-Han Ngai. Ocr-memory: Optical context retrieval for long-horizon agent memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10409–10420, 2026a.

Mo Li, LH Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering llms with cognitive agency via active context management. In *International Conference on Learning Representations*, volume 2026, pp. 153411–153440, 2026b.

Tianjian Li, Jingyu Zhang, William Jurayj, Xi Wang, Chuanyang Jin, Mehrdad Farajtabar, Eric Nalisnick, and Daniel Khashabi. Self-compacting language model agents. *arXiv preprint arXiv:2606.23525*, 2026c.

Xiaochuan Li, Ryan Ming, Meng Chu, Shuai Shao, Rong Jin, and Chenyan Xiong. Acm: Agentic context management for long horizon tasks. *arXiv preprint arXiv:2607.23809*, 2026d.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the association for computational linguistics*, 12:157–173, 2024.

Shukai Liu, Bo Jiang, Jian Yang, Yizhi Li, Jinyang Guo, Xianglong Liu, and Bryan Dai. Context as a tool: Context management for long-horizon swe-agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20604–20617, 2026.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of llm agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13851–13870, 2024.

Matéo Mahaut, Laura Aina, Paula Czarnowska, Momchil Hardalov, Thomas Müller, and Lluís Màrquez. Factual confidence of llms: on reliability and robustness of current estimators. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4554–4570, 2024.

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.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, et al. 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.

Zhiyu Shen, Ziming Wu, Fuming Lai, Shaobing Lian, and Yanghui Rao. Membuilder: Reinforcing llms for long-term memory construction via attributed dense rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27868–27887, 2026.

Haoran Sun, Shaoning Zeng, and Bob Zhang. H-mem: Hierarchical memory for high-efficiency long-term reasoning in llm agents. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 341–350, 2026a.

Haoran Sun, Zekun Zhang, and Shaoning Zeng. Preference-aware memory update for long-term llm agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 783–793, 2026b.

Tencent WorkBuddy Bench Team, Siqi Cai, Shaopeng Chen, Xiang Fei, Yong Mao, Zihan Xu, Zhiheng Lyu, Zhijian Shao, Yuchen Shi, Shuwen Zhang, et al. Tencent workbuddy bench: A multi-domain coding-agent benchmark with contamination-resistant task construction. *arXiv preprint arXiv:2607.20911*, 2026.

Nikhil Verma. Active context compression: Autonomous memory management in llm agents. *arXiv preprint arXiv:2601.07190*, 2026.

XinQi Wang, Jinwei Xiao, Sijia Cui, Hongming Zhang, Yanna Wang, Qingyang Zhang, and Bo Xu. Hymem: Hierarchical context management for long-horizon agents via information isolation. *arXiv preprint arXiv:2608.15703*, 2026a.

Yuhang Wang, Yuling Shi, Mo Yang, Rongrui Zhang, Shilin He, Heng Lian, Yuting Chen, Siyu Ye, Kai Cai, and Xiaodong Gu. Swe-pruner: Self-adaptive context pruning for coding agents. *arXiv preprint arXiv:2601.16746*, 2026b.

Lei Wei, Xiao Peng, Guannan Zhang, Chenhao Jiang, Hongyu Li, Lanbo Lin, Yuanwu Xu, Jiayao Liu, Kesu Wang, Bin Wang, et al. Pace: Predictive adaptive context extraction for long-horizon llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199, 2026.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts llm agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026a.

Binyan Xu, Haitao Li, and Kehuan Zhang. Llm agents are latent context managers: Eliciting self-managed context via state proprioception. *arXiv preprint arXiv:2606.30005*, 2026b.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483, 2026.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Jiaquan Zhang, Chaoning Zhang, Shuxu Chen, Zhenzhen Huang, Pengcheng Zheng, Zhicheng Wang, Ping Guo, Fan Mo, Sung-Ho Bae, Jie Zou, et al. Lightweight llm agent memory with small language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12914–12929, 2026a.

Yuwei Zhang, Chengyu Dong, Shuowei Jin, Changlong Yu, Hejie Cui, Hongye Jin, Xinyang Zhang, Hamed Bonab, Colin Lockard, Jianshu Chen, et al. Comem: Context management with a decoupled long-context model. *arXiv preprint arXiv:2605.30842*, 2026b.

Jinchang Zhu, Jindong Li, Cheng Zhang, Jiahong Liu, and Menglin Yang. Hela-mem: Hebbian learning and associative memory for llm agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13757–13769, 2026.

A MEMORY-DECISION DATA AND PROBE ANALYSIS

This appendix details the memory-decision corpus, representation probes, compact-context analysis, online controller, and supplementary evaluations. The mathematical results characterize the stated losses and control rules. The probing analysis quantifies how accurately compression and recall decisions can be predicted from the model's internal representations.

A.1 ANNOTATED MEMORY DECISIONS AND DECISION-STATE RECONSTRUCTION

The analysis corpus contains 72,912 decision points from 2,521 trajectories covering 600 questions. These questions are separate from the WorkBuddyBench evaluation tasks.

An LLM annotator identifies decision states where compression or recall is needed. For trajectory τ, let $\mathcal{A}^{\mathrm{c}}(\tau)$ and $\mathcal{A}^{\mathrm{r}}(\tau)$ denote the annotated compression and recall decision sets. At decision step t, with o_t denoting the context available before the next action, the labels are

$$y_t^m = \mathbf{1}[t \in \mathcal{A}^m(\tau)], \qquad m \in \{\mathrm{c}, \mathrm{r}\}. \tag{11}$$

Here c denotes compression and r denotes recall. Compression indicates that older context should be condensed while preserving task-relevant information. Recall indicates that the current reasoning state needs earlier evidence that is no longer available in the active context. Recall is evaluated only when valid external memory exists. A trajectory may contain multiple positive decision points.

Representation extraction uses the provenance-matched OPD iter-2 Qwen3.5-9B checkpoint, which remains frozen during probe fitting. The annotator output and the future action are used only to define the target and are not included in the representation input.

All states and trajectory replicas with the same `query_id` remain in one split. Training, validation, and testing use grouped 70/15/15 partitions with seeds 0, 1, 2, 3, and 4. The seed controls both the grouped split and readout initialization. Within-trajectory states remain statistically dependent.

A.2 PROBE ARCHITECTURE AND TRAINING

The primary feature is the final input-token state after the output RMSNorm of frozen Qwen3.5-9B, $\mathbf{h}_t \in \mathbb{R}^{4096}$. In online use, this feature extractor is separate from the task-executing model; the controller does not require hidden-state access to an API-backed acting model. Each target has an independent readout:

$$\begin{aligned} p^m_{t,\mathrm{lin}} &= \sigma((\mathbf{w}^m)^\top \mathbf{h}_t + b^m), \\ p^m_{t,\mathrm{mlp}} &= \sigma\big((\mathbf{v}^m)^\top \mathrm{GELU}(\mathbf{W}^m \mathbf{h}_t + \mathbf{b}^m) + c^m\big). \end{aligned} \tag{12}$$

The MLP hidden width is 256. Including biases, the linear and MLP probes have 4,097 and 1,049,089 parameters per target. Only readout parameters are trained; the language model remains frozen.

The Full Context reference used in the compact-context analysis is the **faithful-hint MLP** with hidden width 256. It retains the recorded context-token hint in the serialized input. Table 6 reports its test means. The same configuration supplies the Full Context teacher and reference in the compact-context analysis. The online compression heads are the separately trained Recent2 decision-distillation students, not the Full Context teacher.

Table 6: **Full Context reference for the compact-context analysis.** Five-seed mean test metrics for the faithful-hint MLP; F1 uses per-seed validation-selected thresholds. Values are rounded only for display.

Target	AUROC ↑	AUPRC ↑	F1 ↑
Compression	**0.829**	0.690	0.639
Recall	**0.780**	0.597	0.558

Among the hint/capacity settings in Table 8, this MLP achieves the highest F1 for both targets, whereas the faithful-hint linear compression probe achieves higher AUROC. This distinction separates thresholded classification performance from ranking performance. Probe training updates only

the readout. The recorded MLP defaults are AdamW, 50 epochs, learning rate 10^{-3}, batch size 256, and hidden width 256. The checkpoint is chosen by validation label loss; the classification threshold is then chosen by validation F1.

For each training split,

$$\mathcal{L}_m = -\frac{1}{N_m}\sum_{i=1}^{N_m}[\rho_m y_i^m \log p_i^m + (1-y_i^m)\log(1-p_i^m)], \qquad \rho_m = \frac{N_m^-}{N_m^+}. \tag{13}$$

The threshold for a fitted probe is selected by validation F1 and held fixed on the test set:

$$\widehat{\theta}_{m,s} \in \arg\max_{\theta\in\Theta} \mathrm{F1}_{\mathrm{val},s}(\mathbf{1}\{p_t^m \geq \theta\}, y_t^m). \tag{14}$$

AUROC is the primary ranking metric. AUPRC complements it for imbalanced targets. Classification recall is distinct from the memory operation called recall. For positive and negative scores, empirical AUROC is

$$\widehat{\mathrm{AUROC}} = \frac{1}{N^+N^-}\sum_{i:y_i=1}\sum_{j:y_j=0}\left[\mathbf{1}\{z_i > z_j\} + \frac{1}{2}\mathbf{1}\{z_i = z_j\}\right]. \tag{15}$$

A.3 Properties of the Weighted Readout

For $p = \sigma(z)$, the per-example loss has the stable form

$$\ell_\rho(z,y) = (\rho y + 1 - y)\log(1+e^z) - \rho y z, \qquad \frac{\partial \ell_\rho}{\partial z} = (\rho y + 1 - y)p - \rho y. \tag{16}$$

The logarithm is evaluated as $\max(z,0) + \log(1+e^{-|z|})$ for numerical stability.

Proposition A.1 (Population score under class weighting) *Let $\eta(\mathbf{h}) = \Pr(Y=1 \mid \mathbf{h})$ and $\rho > 0$. For $0 < \eta < 1$, conditional weighted BCE has the unique minimizer*

$$p_\rho^*(\mathbf{h}) = \frac{\rho\eta}{1-\eta+\rho\eta}, \qquad \operatorname{logit} p_\rho^* = \operatorname{logit}\eta + \log\rho. \tag{17}$$

Proof. The conditional risk is $R(p) = -\rho\eta\log p - (1-\eta)\log(1-p)$. Solving $R'(p) = -\rho\eta/p + (1-\eta)/(1-p) = 0$ gives the stated minimizer. Since $R''(p) = \rho\eta/p^2 + (1-\eta)/(1-p)^2 > 0$, it is unique. Taking log odds gives the second identity. The inverse relation is

$$\eta = \frac{p_\rho^*}{\rho - (\rho-1)p_\rho^*}. \tag{18}$$

Boundary cases follow by limits. □

The weighting changes the probability interpretation of the population-optimal score. Finite-probe calibration remains a separate empirical property, and validation-based threshold selection determines the operating point used in the evaluation protocol.

A.4 Observable Controls, Probe Capacity, and Memory Hints

The controls measure whether length, interaction progress, tool metadata, or surface text explain the memory-decision signal. Recent2 text controls use compact inputs, whereas the hidden-state probes use Full Context; predictive performance is therefore compared across different input-token budgets.

The Full Context faithful-hint MLP reference outperforms the tested observable controls for both targets. Table 8 reports additional probe-capacity and memory-hint measurements. The no-hint setting removes explicit memory-token hints, whereas the faithful setting preserves the recorded hint.

Table 7: **Memory-decision prediction controls.** Diagnostic rows show mean ± standard deviation over five question-grouped splits. The faithful-hint MLP reference is listed separately with mean metrics.

Signal	Compression		Recall	
	AUROC ↑	AUPRC ↑	AUROC ↑	AUPRC ↑
Length + turn	0.751 ± 0.066	0.533 ± 0.053	0.552 ± 0.074	0.336 ± 0.036
Last tool type	0.562 ± 0.026	0.335 ± 0.054	0.527 ± 0.056	0.315 ± 0.043
Last action type	0.583 ± 0.019	0.346 ± 0.057	0.535 ± 0.056	0.319 ± 0.039
Length + turn + tool type	0.758 ± 0.056	0.543 ± 0.054	0.549 ± 0.058	0.361 ± 0.066
Recent2 TF-IDF	0.616 ± 0.027	0.416 ± 0.072	0.610 ± 0.056	0.398 ± 0.075
Recent2 text embedding	0.546 ± 0.015	0.363 ± 0.047	0.594 ± 0.041	0.385 ± 0.086
Full Context (faithful-hint MLP)	0.829	0.690	0.780	0.597

Table 8: **Additional probe-capacity and memory-hint measurements.** Mean AUROC, AUPRC, and validation-threshold F1 over five splits. Each row denotes the named probe and hint configuration.

Setting	AUROC ↑	AUPRC ↑	F1 ↑
Compression			
No hint, linear	0.831	0.670	0.601
No hint, MLP	0.827	0.665	0.606
Faithful hint, linear	0.836	0.697	0.624
Faithful hint, MLP	0.829	0.690	**0.639**
Recall			
No hint, linear	0.765	0.585	0.526
No hint, MLP	0.763	0.592	0.541
Faithful hint, linear	0.774	0.598	0.543
Faithful hint, MLP	0.780	0.597	**0.558**

A.5 Layerwise Diagnostic Measurements

The layerwise study uses independently fitted no-hint linear readouts on Qwen3.5-9B to compare depths and pooling locations at a fixed readout capacity. The faithful-hint MLP supplies the Full Context reference for the separate compact-context analysis. The final linear diagnostic completes the same-capacity comparison between last-token and mean-pooled representations at the model's final normalized output state.

At layer ℓ, independently trained probes use either the last input state or mean pooling:

$$\mathbf{h}^{(\ell)}_{t,\text{last}} = \phi^{(\ell)}(o_t)[-1], \qquad \mathbf{h}^{(\ell)}_{t,\text{mean}} = \frac{1}{n_t}\sum_{j=1}^{n_t} \phi^{(\ell)}(o_t)[j]. \tag{19}$$

At every intermediate checkpoint with both pooling measurements, the last-token linear readout outperforms mean pooling. Recall reaches its highest listed intermediate-layer mean at block 16, while compression AUROC increases across the listed intermediate last-token checkpoints. The compact-context Full Context reference uses a faithful-hint MLP, whereas the layerwise diagnostics use no-hint linear readouts. The online controller uses the final state for compression decisions; recall probing characterizes the predictive signal separately from online retrieval over available stored summaries.

Proposition A.2 (Scale dependence of logit separation and AUROC invariance) *For* $z = \mathbf{w}^\top \mathbf{h} + b$ *with finite class-conditional means,*

$$\Delta = \mathbb{E}[z \mid Y = 1] - \mathbb{E}[z \mid Y = 0] = \mathbf{w}^\top(\boldsymbol{\mu}_+ - \boldsymbol{\mu}_-). \tag{20}$$

Table 9: **Layerwise linear diagnostics and the compact-context reference.** Diagnostic rows show mean ± standard deviation over five question-grouped splits using same-capacity linear readouts. The bottom row reports the faithful-hint MLP reference used in the separate compact-context analysis, identifying its distinct readout configuration.

Representation	Pooling	Compression		Recall	
		AUROC ↑	AUPRC ↑	AUROC ↑	AUPRC ↑
Block 4	Last	0.757 ± 0.051	0.563 ± 0.063	0.722 ± 0.058	0.526 ± 0.087
	Mean	0.706 ± 0.027	0.483 ± 0.063	0.632 ± 0.073	0.435 ± 0.079
Block 8	Last	0.799 ± 0.044	0.623 ± 0.082	0.768 ± 0.053	0.594 ± 0.112
	Mean	0.735 ± 0.044	0.493 ± 0.068	0.690 ± 0.058	0.483 ± 0.093
Block 16	Last	0.810 ± 0.031	0.644 ± 0.082	0.780 ± 0.054	0.583 ± 0.108
	Mean	0.762 ± 0.048	0.547 ± 0.064	0.741 ± 0.048	0.529 ± 0.099
Block 24	Last	0.814 ± 0.024	0.649 ± 0.089	0.765 ± 0.061	0.577 ± 0.116
	Mean	0.756 ± 0.049	0.534 ± 0.047	0.726 ± 0.059	0.515 ± 0.100
Block 32	Last	0.824 ± 0.032	0.659 ± 0.087	0.743 ± 0.088	0.560 ± 0.118
	Mean	0.708 ± 0.059	0.478 ± 0.069	0.692 ± 0.045	0.474 ± 0.066
Final RMSNorm	Last	0.831 ± 0.036	0.670 ± 0.100	0.765 ± 0.067	0.584 ± 0.106
	Mean	0.712 ± 0.058	0.479 ± 0.074	0.696 ± 0.047	0.479 ± 0.067
Compact-context Full Context reference: faithful-hint MLP					
Final RMSNorm	Last	0.829	0.690	0.780	0.597

Replacing z by $z' = az + c$, $a > 0$, preserves AUROC and changes the gap to $\Delta' = a\Delta$.

Proof. Linearity of expectation gives the displayed gap; the intercept cancels. For each positive and negative pair, $z'_i > z'_j$ if and only if $z_i > z_j$, and ties are also preserved. Every summand in Equation (15) is consequently unchanged. Taking the difference of conditional expectations of $az + c$ gives $\Delta' = a\Delta$. □

Layerwise logit gaps depend on score scale and are therefore interpreted alongside AUROC when comparing predictive signals across representations. The capacity-control and layerwise tables report separately aggregated measurements. For no-hint linear recall, AUPRC is 0.584743 in the capacity-control export and 0.584336 in the layerwise export. Rounding these respective aggregates to three decimal places yields the displayed values of 0.585 and 0.584 in the two tables.

A.6 Paired Compact-Context Comparisons

For paired split-level differences $d_s = M_{A,s} - M_{B,s}$ and $\bar{d} = S^{-1} \sum_s d_s$, the reported two-sided sign-flip calculation is

$$p_{\text{flip}} = 2^{-S} \sum_{\boldsymbol{\epsilon} \in \{-1,1\}^S} \mathbf{1}\left\{ \left| S^{-1} \sum_s \epsilon_s d_s \right| \geq |\bar{d}| \right\}. \tag{21}$$

The inferential interpretation requires sign exchangeability. With $S = 5$, the observed sign pattern and its global negation have the same absolute statistic, so $p_{\text{flip}} \geq 2/2^5 = 0.0625$. The smallest attainable two-sided p-value in this five-split calculation is therefore 0.0625, above the 0.05 threshold.

B Compact Context and Historical Evidence

B.1 Context Construction and Evaluation

Let P_t be the system and task prefix, $\mathcal{R}_t$ the two most recent complete interaction blocks, and $\mathcal{E}_t$ retrieved historical evidence. The principal constructions are

$$C_t^{\text{recent}} = P_t \oplus \mathcal{R}_t, \qquad C_t^{\text{recover}} = P_t \oplus \mathcal{E}_t \oplus \mathcal{R}_t. \tag{22}$$

Table 10: **Paired compact-context differences across split seeds.** C and R denote compression and recall; positive differences favor the first method. All listed tests are computed from paired split-level differences for the compact-context interventions and their corresponding probe configurations.

Comparison	Target / metric	Mean difference	Sign-flip p
Recent2 KD − Recent2	C / AUROC	+0.0058	0.500
	R / AUROC	−0.0080	0.500
	C / F1	+0.0570	0.0625
	R / F1	+0.0278	0.3125
Raw Top5 − Hidden Bank Recent5	C / AUROC	+0.0627	0.0625
	R / AUROC	+0.0389	0.1875

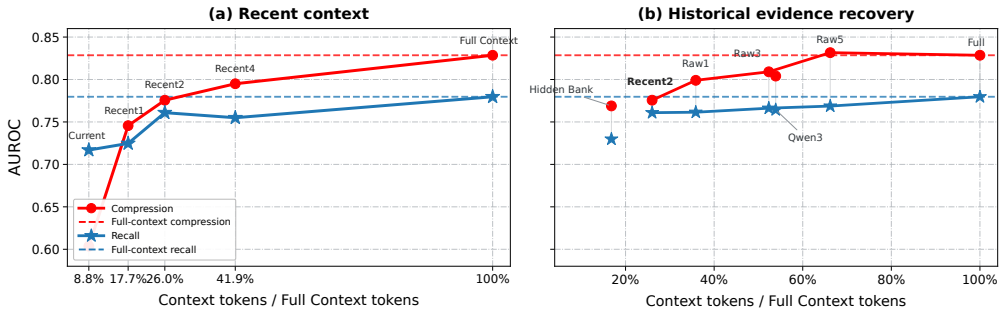


Figure 4: **Decision information at different input budgets.** Left: recent context. Right: historical evidence recovery. Token fractions refer to the corresponding offline comparison.

The prefix and annotation labels are held fixed when changing history. The `Current` condition uses the system and task prefix alone, providing a no-raw-history reference for the compact-context comparisons. Other conditions retain the same prefix and add the specified recent interaction blocks or historical evidence, so the experimental variation concerns the available history. Relative input cost on the evaluated states is

$$B(C) = 100 \frac{\sum_i \mathrm{Tok}(C_i)}{\sum_i \mathrm{Tok}(C_i^{\mathrm{full}})}. \tag{23}$$

This ratio measures offline text-input volume; full-task token consumption and hidden-state storage require separate accounting under their respective measurement protocols. Agreement with the Full Context teacher is a separate measurement:

$$\mathrm{Agr}_m = N_m^{-1} \sum_i \mathbf{1}\{\widehat{y}^m_{i,\mathrm{candidate}} = \widehat{y}^m_{i,\mathrm{full}}\}. \tag{24}$$

Annotation-based metrics evaluate prediction quality; teacher agreement measures consistency.

Recent2 uses 26.0% of the full textual input and retains much of the memory-decision signal. Selected raw evidence outperforms the evaluated hidden-bank alternatives, whereas recurrent and scalar-state additions show mixed predictive results relative to Recent2. Each comparison reports its token budget alongside predictive performance, allowing the input cost of these history representations to be considered explicitly.

B.2 Decision Distillation

For student logit z, annotation label y, and fixed Full Context teacher score $q \in [0, 1]$, the compression heads use

$$\ell_{\mathrm{KD}}(z, y, q) = \ell_\rho(z, y) + \lambda \,\mathrm{BCE}(\sigma(z), q), \qquad \lambda = 0.5. \tag{25}$$

The teacher is used only during training. The logit gradient is

$$\partial_z \ell_{\mathrm{KD}} = (\rho y + 1 - y + \lambda)\sigma(z) - \rho y - \lambda q. \tag{26}$$

Table 11: **Compact-context and historical-evidence comparison.** C and R denote compression and recall; dots denote unreported teacher agreement.

Representation	Tokens(%)	Compression		Recall		Agreement	
		AUROC	F1	AUROC	F1	Comp.	Recall
Full Context	100.0	0.829	0.639	0.780	0.558	·	·
Recent context							
Current	8.8	0.604	0.451	0.717	0.532	·	·
Recent1	17.7	0.746	0.520	0.725	0.524	·	·
Recent2	26.0	0.776	0.541	0.761	0.520	·	·
Recent4	41.9	0.795	0.550	0.755	0.555	·	·
Scalar state and decision distillation							
Recent2 + scalar state	26.0	0.767	0.569	0.746	0.552	·	·
Recent4 + scalar state	41.9	0.792	0.579	0.757	0.586	·	·
Recent2 + decision KD	26.0	0.781	0.598	0.753	0.548	0.819	0.840
Recent4 + decision KD	41.9	0.794	0.571	0.758	0.530	0.824	0.857
Recovered raw evidence							
Selective Raw Top1	35.9	0.799	0.569	0.761	0.542	0.826	0.817
Selective Raw Top3	52.3	0.809	0.565	0.766	0.554	0.826	0.834
Selective Raw Top5	66.2	0.832	0.602	0.769	0.567	0.859	0.865
Hidden and recurrent memory							
Hidden Bank Uniform5	16.8	0.722	0.526	0.729	0.540	0.731	0.779
Hidden Bank Recent5	16.8	0.769	0.529	0.730	0.520	0.785	0.744
Hidden Bank Similarity1	16.8	0.716	0.514	0.713	0.507	0.729	0.721
Hidden Bank Similarity3	16.8	0.722	0.509	0.703	0.510	0.729	0.722
Hidden Bank Similarity5	16.8	0.728	0.555	0.712	0.493	0.722	0.747
Hidden Bank Similarity8	16.8	0.725	0.552	0.703	0.535	0.758	0.736
GRU512	16.8	0.745	0.499	0.727	0.513	0.773	0.757
Hybrid GRU + Similarity5	16.8	0.734	0.517	0.672	0.508	0.734	0.684

Proposition B.1 (Distillation as weighted soft-label learning) *For $\rho > 0$, $\lambda \geq 0$, and binary y, define*

$$a = \rho y + 1 - y + \lambda, \qquad \widetilde{y} = \frac{\rho y + \lambda q}{a}. \tag{27}$$

Then $\ell_{\mathrm{KD}} = a\,\mathrm{BCE}(\sigma(z), \widetilde{y})$. Moreover, conditional on the student's compact representation h, let $\eta(h) = \Pr(Y = 1 \mid h)$ and $\mu(h) = \mathbb{E}[q \mid h]$. The optimal unrestricted student probability is

$$p^*_\lambda(h) = \frac{\rho\eta(h) + \lambda\mu(h)}{1 + (\rho - 1)\eta(h) + \lambda}. \tag{28}$$

Proof. Writing $p = \sigma(z)$, the coefficient of $-\log p$ in the combined loss is $\rho y + \lambda q$, and that of $-\log(1-p)$ is $1 - y + \lambda(1 - q)$. Their sum is $a > 0$, and their ratio to a gives the soft target. After conditioning on h, the coefficients become $A = \rho\eta + \lambda\mu$ and $B = 1 - \eta + \lambda(1 - \mu)$. Minimizing $-A \log p - B \log(1 - p)$ gives $p^* = A/(A + B)$, yielding Equation (28). When $A, B > 0$, strictly positive second derivative proves uniqueness. Endpoint solutions follow by continuity. The conditional excess risk is

$$R_h(p) - R_h(p^*_\lambda) = (A + B)\,\mathrm{KL}(\mathrm{Bern}(p^*_\lambda) \| \,\mathrm{Bern}(p)), \tag{29}$$

which follows directly by subtracting the two cross-entropies. □

Distillation transfers the teacher's conditional score signal into the compact representation, with the derived optimum depending on that signal regardless of its calibration. Empirically, Recent2 KD yields a larger improvement in compression F1 than in compression AUROC, alongside a decrease in recall AUROC.

B.3 Decision and Ranking Stability

Proposition B.2 (Stability under compact-context perturbations) *Apply the same fixed linear readout $z = \mathbf{w}^\top \mathbf{h} + b$ to full and compact representations. Suppose $\|\mathbf{h}_f - \mathbf{h}_c\|_2 \le \varepsilon$ and put $\delta = \|\mathbf{w}\|_2 \varepsilon$. For a probability threshold $0 < \theta < 1$ and $a = \operatorname{logit} \theta$, the two decisions agree whenever*

$$|z_f - a| > \delta. \tag{30}$$

If the score bound holds almost surely under the evaluated state distribution (including the empirical distribution on fixed examples), then

$$\Pr(\widehat{Y}_f \neq \widehat{Y}_c) \le \Pr(|Z_f - a| \le \delta). \tag{31}$$

For fixed sets containing $N^+ > 0$ positive and $N^- > 0$ negative examples, it also implies

$$\left|\widehat{\mathrm{AUROC}}_f - \widehat{\mathrm{AUROC}}_c\right| \le \frac{1}{N^+ N^-} \sum_{i:y_i=1} \sum_{j:y_j=0} \mathbf{1}\{|z_{f,i} - z_{f,j}| \le 2\delta\}. \tag{32}$$

Proof. The Cauchy Schwarz inequality gives $|z_f - z_c| \le \|\mathbf{w}\|_2 \|\mathbf{h}_f - \mathbf{h}_c\|_2 \le \delta$. A logit farther than δ from a cannot cross the threshold, proving the decision statements. For any positive and negative pair, the difference between its compact and full score gaps is at most 2δ. A full gap with absolute value greater than 2δ therefore retains its sign. Its AUROC summand is unchanged. Every remaining pair can change that summand by at most one, including ties. Summing these bounds and dividing by $N^+ N^-$ proves Equation (32). □

These sufficient conditions apply to a shared fixed readout under the stated representation bound; the experimental tables separately evaluate probes trained for each context configuration.

B.4 Retrieval Controls and Evidence Budgets

The offline retrieval study compares saved raw historical blocks. Selective Raw Top$K$ ranks eligible older blocks using token-frequency cosine similarity to the current last block in the recorded decision state, excluding Recent2. Ranking uses this observed text similarity alone, with future actions and annotation labels excluded from the selector's inputs. Qwen conditions instead use Qwen3-Embedding-8B and the recorded task/Recent2 query serialization. Offline Qwen retrieval returns raw text, whereas the canonical online PAMER controller retrieves stored summaries. Accordingly, the offline analysis evaluates raw-block recovery, and the system experiments evaluate the deployed summary-retrieval controller.

Table 12: **Offline retrieval controls.** Token percentages are relative to Full Context; agreement is measured against the corresponding teacher.

Retriever	Tokens(%)	**AUROC**		**Agreement**	
		Comp.	Recall	Comp.	Recall
Random Top3	47.2	0.793	0.752	0.786	0.849
Recency Top3	46.2	0.792	0.753	0.818	0.857
Selective Raw Top1	35.9	0.799	0.761	0.826	0.817
Selective Raw Top3	52.3	0.809	0.766	0.826	0.834
Selective Raw Top5	66.2	0.832	0.769	0.859	0.865
Qwen Top1	37.8	0.794	0.753	0.793	0.839
Qwen Top3	53.9	0.804	0.764	0.838	0.838
Qwen Top5	67.7	0.816	0.775	0.834	0.872

Increasing retrieval cardinality changes both evidence and input cost. Top3 provides an intermediate operating point between lower-input Top1 and the higher-AUROC Top5 condition. For complete-block costs $\ell_j \ge 0$, an evidence budget obeys

$$\sum_{j \in \mathcal{J}_t} \ell_j \le B_{\text{evidence}}. \tag{33}$$

Table 13: **Qwen Top3 evidence budgets.** Mean, P95, and maximum summarize the recalled-evidence portion of each offline input.

Variant	Recalled tokens			Agreement	
	Mean	P95	Max	Comp.	Recall
Top3 cap 8K	5,540	8,021	8,192	0.818	0.835
Top3 cap 16K	8,765	14,801	16,383	0.843	0.837
Top3 uncapped	9,167	16,509	32,245	0.838	0.838

The prefix, recent context, and serialization overhead are additional parts of the final request.

The 16K cap retains agreement close to the uncapped condition, whereas the 8K cap reduces agreement. These offline measurements characterize the relation between evidence-token budgets and agreement with the Full Context teacher.

C Online Memory Controller

C.1 Compression Voting and Deferred Retrieval

After removing duplicate blocks already represented in stored memories, write the working history as $C_t^{\text{eff}} = P_t \oplus \mathcal{O}_t \oplus \mathcal{R}_t$. Five Recent2 decision-distillation MLP heads read the final normalized Qwen3.5-9B state of the serialized $P_t \oplus \mathcal{R}_t$ input. The saved probe system prompt and context-token hint are applied before serialization. With per-seed validation thresholds θ_s,

$$v_{t,s} = \mathbf{1}\{p^{\text{c}}_{t,s} \geq \theta_s\}, \qquad u_t = \mathbf{1}\{\text{Tok}(\mathcal{O}_t) \geq 1024\}\mathbf{1}\left\{\sum_{s=1}^{5} v_{t,s} \geq 3\right\}. \tag{34}$$

On successful compression, an entry stores summary s_j, its embedding, and protocol-aware identities $\mathcal{B}_j$ of the covered raw blocks:

$$M_j = (s_j, \mathbf{e}_j, \mathcal{B}_j), \qquad \mathbf{e}_j = E(s_j). \tag{35}$$

The encoder is Qwen3-Embedding-8B with 1,024 output dimensions, distinct from the 4,096-dimensional probe feature. Raw blocks leave active context only after successful storage. The same request receives neither the new summary nor an immediate recall; its reduced context is $P_t \oplus \mathcal{R}_t$.

On a later request, the task and Recent2 define an embedding query. For nonzero embeddings,

$$a_{t,j} = \frac{\mathbf{q}_t^\top \mathbf{e}_j}{\|\mathbf{q}_t\|_2 \|\mathbf{e}_j\|_2}, \qquad \mathcal{I}_t = \text{TopK}_j(a_{t,j}), \quad K = 3. \tag{36}$$

At most the available number of memories is retrieved. Query text is serialized from the task and Recent2 and limited to its final 30,000 characters. Selected summaries are inserted after the prefix and before *all* retained raw blocks, preserving both unarchived older blocks and the Recent2 history:

$$C_t^{\text{recall}} = P_t \oplus \mathcal{E}_t \oplus \mathcal{O}_t \oplus \mathcal{R}_t. \tag{37}$$

Here $\mathcal{E}_t$ contains the selected summaries, retrieved independently of the learned recall probe on requests eligible for recall. A failed probe skips compression but may still use existing memory; a successful compression request never performs recall on that same request.

C.2 Stability of Voting and Retrieval

Proposition C.1 (Majority voting without an independence assumption) *At an eligible state with binary target Y, let E_s indicate an incorrect decision by head s. The five-head majority error satisfies*

$$\Pr(E_{\text{maj}} \mid \mathcal{C}) \leq \min\left\{1, \frac{1}{3}\sum_{s=1}^{5} \Pr(E_s \mid \mathcal{C})\right\} \tag{38}$$

for any common conditioning information $\mathcal{C}$. Further, if at least three heads retain their original majority vote under an input perturbation, the majority decision is unchanged.

Algorithm 1 **Online context transformation with PAMER**

Require: Canonical request C_t^{raw}, persisted memory $\mathcal{M}_t$, frozen Qwen3.5-9B, five Recent2 KD heads and thresholds.

if *the session identifier or conversation/tool structure is invalid* **then**
  return C_t^{raw}
end
Load valid memory; if its saved state is corrupt, discard the invalid state and start with empty memory
Construct C_t^{eff} by removing raw blocks already covered by valid persisted memories
Split C_t^{eff} into P_t, $\mathcal{O}_t$, and $\mathcal{R}_t$
Set $u_t \leftarrow 0$
if $\text{Tok}(\mathcal{O}_t) \geq 1024$ **then**
  Extract the final Qwen3.5-9B state and evaluate the five heads
  if *feature extraction and head evaluation succeed* **then**
    Set $u_t \leftarrow \mathbf{1}\{\sum_s \mathbf{1}[p_{t,s} \geq \theta_s] \geq 3\}$
  else
    Record the probe error and leave $u_t = 0$ `// Existing recall remains possible`
  end
end
if $u_t = 1$ **then**
  Generate a summary and its embedding, and validate protocol-aware source identities
  Require successful storage and a strict reduction in active request tokens
  if *all transaction checks succeed* **then**
    Commit the memory and **return** $P_t \oplus \mathcal{R}_t$ `// No same-request recall`
  else
    Commit no partial memory and **return** C_t^{raw}
  end
end
if $\mathcal{M}_t \neq \varnothing$ **then**
  Retrieve at most three summaries using the task and Recent2
  Construct $C_t' = P_t \oplus \mathcal{E}_t \oplus \mathcal{O}_t \oplus \mathcal{R}_t$
  if *retrieval and final tool-protocol validation succeed* **then**
    return C_t'
  else
    return C_t^{raw}
  end
end
return C_t^{eff}

Proof. A wrong majority requires at least three wrong heads, hence pointwise $\mathbf{1}\{E_{\text{maj}}\} \leq \frac{1}{3}\sum_s \mathbf{1}\{E_s\}$. Taking conditional expectations gives the probability bound. For stability, three unchanged votes already constitute a majority, regardless of the remaining two votes. In particular, a head with logit z_s retains its vote if $|z_s - \text{logit}\,\theta_s|$ exceeds its maximum logit perturbation. □

The bound characterizes majority-error probability for the five-head controller while allowing statistical dependence among the errors of its individual prediction heads.

Proposition C.2 (Top-K retrieval stability) *Let scores a_j and perturbed scores a_j' satisfy $|a_j' - a_j| \leq \delta$ for every memory entry. If there are more than K entries and the ordered-score boundary satisfies*

$$a_{(K)} - a_{(K+1)} > 2\delta, \tag{39}$$

then the retrieved Top-K set is unchanged. For unit-normalized memory embeddings and fixed entries, a query change from $\widehat{\mathbf{q}}$ to $\widehat{\mathbf{q}}'$ permits $\delta = \|\widehat{\mathbf{q}}' - \widehat{\mathbf{q}}\|_2$.

Proof. For an originally selected entry i and an unselected entry j, $a_i - a_j \geq a_{(K)} - a_{(K+1)}$. Therefore, $a'_i - a'_j \geq a_i - a_j - 2\delta > 0$. No unselected entry can overtake any selected entry, proving set equality. For cosine scores with unit memory vector $\widehat{\mathbf{e}}_j$, The Cauchy Schwarz inequality gives $|a'_j - a_j| = |\langle \widehat{\mathbf{q}}' - \widehat{\mathbf{q}}, \widehat{\mathbf{e}}_j \rangle| \leq \delta$. □

This result characterizes Top-K membership stability under score perturbations through the margin at the selection boundary.

C.3 Step-Level Evidence in PAMER+

In the described implementation, PAMER+ provides step-granular evidence selection through the LLM-router integration. The canonical embedding sidecar described above retrieves whole summaries. These are distinct retrieval paths: the component algorithm below specifies the exact-step extension, while canonical PAMER uses the summary-retrieval path.

The router is shown the current Recent2 context and the structured summary index, including `step_details`; it selects summary identifiers and the smallest requested `raw_steps_needed` set. The target future action is not an input. Summary identifiers locate candidate memories but do not authorize expansion of every covered step. Only complete raw blocks explicitly named by the selector are supplied to the Recall Agent. The step-granular component is responsible for validating and normalizing the selection; compression gating and model invocation belong to the surrounding integration. For requested identifiers $\mathcal{J}_t$,

$$\mathcal{J}_t \subseteq \bigcup_{j \in \mathcal{I}_t} \mathcal{B}_j. \tag{40}$$

Partitions are checked for gaps, overlaps, missing steps, and invalid identifiers. The requested step identifiers determine the subset restored from each selected summary. Cache identity uses sorted summary and step identifiers; reuse assumes unchanged associated content and configuration. The key is computed from sorted summary and raw-step identifiers, rather than the router's goal wording or unrelated history updates. The recorded cache protocol distinguishes an empty selection, an invalid packet, selector changes, material changes, and a hit. A successful packet or a declined recall may be reused for an unchanged exact selector.

The caller distinguishes a valid empty selection from a failed operation. An empty selection injects no recalled text. Invalid or failed recall results are not injected and do not authorize deletion of raw history. The calling controller applies its operation-specific fallback. Final conversation and tool-structure validation is required before a transformed request is sent to the agent. The compared implementations also differ in routing, retries, and request scaffolding; their historical system results therefore measure the combined effect of these implementation changes.

C.4 Storage, Evidence Selection, and Token Accounting

Proposition C.3 (Successful storage precedes removal) *Suppose a compression update removes active raw blocks only after the summary, embedding, and valid provenance links are stored successfully, and a failed update returns the original request. Then no failed update removes information from the returned active request. If raw-step validation enforces Equation* (40)*, any raw block supplied to the recall operation comes from an authorized memory.*

Proof. Before the storage condition holds, the algorithm has no branch returning the reduced context. Failure instead returns the preserved original request, proving the first statement. In the recall branch, valid source links are checked before raw blocks are selected. Invalid selections are rejected. Every surviving selected block consequently has an identifier in the authorized union, proving the second statement. □

Structural validation and provenance checks establish protocol consistency and source membership; summary fidelity and the sufficiency of selected evidence remain separate task-level properties. Under additive complete-block accounting, selecting $\mathcal{J}_t \subseteq \mathcal{U}_t := \bigcup_{j \in \mathcal{I}_t} \mathcal{B}_j$ reduces the raw evidence volume by

$$\sum_{j \in \mathcal{U}_t} \ell_j - \sum_{j \in \mathcal{J}_t} \ell_j = \sum_{j \in \mathcal{U}_t \setminus \mathcal{J}_t} \ell_j \geq 0. \tag{41}$$

Algorithm 2 **Exact-step selection validation in the PAMER$^+$ component**

Require: Stored memories, selected summary identifiers $\mathcal{I}_t$, requested raw step identifiers $\mathcal{J}_t$.
Ensure: Validated recalled evidence, a valid empty selection, or failure $\perp$.

if $\mathcal{J}_t = \varnothing$ **then**
  return an empty evidence set
end
Validate the partitions and provenance links of the selected memories
if *the links are invalid* **then**
  return $\perp$
end
Select the complete raw blocks specified by $\mathcal{J}_t$
if *a step is missing, duplicated, or unauthorized* **then**
  return $\perp$
end
Compute cache identity from the sorted summary and step identifiers
if *a cached result exists for this selection* **then**
  return the cached result
end
Run the recall operation using only the selected evidence
if *the recall call fails or the result cites steps outside* $\mathcal{J}_t$ **then**
  return $\perp$
end
Cache the validated result
return the recalled evidence

This concerns raw evidence volume; selection calls and serialization can add their own costs.

For a fixed request sequence, let L_j be an archived block's length and m_j the number of avoided occurrences in future requests. Let A_{add} include additional counted memory-operation tokens and retrieval text. Then

$$\Delta T_{\mathrm{history}} = \sum_j m_j L_j - A_{\mathrm{add}}. \tag{42}$$

Each avoided occurrence contributes L_j, giving the first sum; subtracting added material proves the identity. Complete online runs can have different actions and lengths, so task totals are measured directly. Recent2 and Top3 control block counts, while total context length also depends on block sizes.

C.5 Implementation Settings and Failure Handling

Summary-source limits are in characters, while model input and output limits are in tokens. If a source exceeds approximately 40,000 characters, messages receive coverage from their beginning and end. Retrieved entries use a stable identifier order, and similarity scores are not inserted into the prompt.

D Evaluation Protocol and System Results

D.1 Benchmark and Task Aggregation

All full-benchmark comparisons use the same fixed 260 WorkBuddyBench tasks: 80 Code, 50 Office, 60 Security, and 70 Web. All Pilot40 comparisons use the same fixed 40 tasks, with ten tasks from each domain. No method-specific task filtering is applied.

For task rewards r_i, the aggregate used in this paper is

$$\overline{r} = N^{-1} \sum_{i=1}^{N} r_i = \sum_d \frac{N_d}{N} \overline{r}_d, \qquad \mathrm{Score} = 100\overline{r}. \tag{43}$$

Table 14: **Controller settings.** Character budgets, token budgets, and embedding dimensions are distinct quantities.

Parameter	Value
Compression controller	
Retained complete blocks	2
Feature extractor	Frozen, provenance-matched Qwen3.5-9B
Feature dimension	4,096
Compression heads	5 Recent2 decision-distillation MLPs
Vote rule	3 of 5
Decision thresholds	Validation best F1 threshold, selected independently for each seed
Probe representation	Final token representation after RMSNorm
MLP hidden width (compact-context reference and online heads)	256
Input and output budgets	
Probe maximum input	35,000 tokens
Prefix retained under truncation	2,048 tokens
Output reserve used in the context hint	16,384 tokens
Minimum compression candidate	1,024 tokens
Summary and retrieval	
Summary source budget	Approximately 40,000 characters
Summary model	Qwen3.5-9B (default), temperature 0, thinking disabled
Summary maximum output	1,024 tokens
Summary timeout / attempts	100 seconds / 8
Embedding timeout / attempts	30 seconds / 8
Embedding model	Qwen3-Embedding-8B
Embedding dimension	1,024
Recall cardinality	Top3

Table 15: **Memory-operation fallback rules.**

Condition	Required behavior
Missing session identifier	Return the original request and record the condition.
Unsafe tool protocol	Return the original request without applying a memory transformation.
Candidate below 1,024 tokens	Skip compression as a normal control-flow branch; recall from existing memory remains possible.
Probe failure or OOM	Skip compression and record the error; existing-memory recall may still be attempted.
Summary, embedding, or storage failure	Return the original request, commit no incomplete memory, and remove no raw block.
Recall transport or shape failure	Return the original request rather than a partially rewritten context.
Invalid step selection or unsafe injection	Reject the recall result; the canonical controller returns the original request on unsafe injection. The step component authorizes no raw-history deletion.
Valid empty step selection	Treat the selection as valid and inject no recalled evidence into the request.
Corrupt state file	Discard invalid saved memory, initialize an empty memory store, and continue processing requests from that state.

Let $S_d = 100\overline{r}_d$ denote a domain score. For complete measurements on the stated fixed sets, the corresponding aggregates are

$$S_{260} = \frac{80S_{\text{Code}} + 50S_{\text{Office}} + 60S_{\text{Security}} + 70S_{\text{Web}}}{260}, \qquad S_{40} = \frac{S_{\text{Code}} + S_{\text{Office}} + S_{\text{Security}} + S_{\text{Web}}}{4}. \tag{44}$$

Because Pilot40 contains exactly ten tasks from each of the four domains, the overall score is the equal-weight average of the four domain scores. All averages and score changes are computed before final display rounding. Domain scores and overall scores are rounded separately for presentation, so recomputing an overall score from the displayed one-decimal domain values may differ slightly from the reported task-level aggregate. The tables report graded task rewards and token usage using consistent score and cost columns.

The outer agent loop uses a turn limit of `CBC_MAX_TURNS`=256. Turn counts are tracked separately from tool calls, transport retries, and internal context operations during agent execution. ACM uses a released policy rather than the same API-backed acting model.

D.2 Token and Context Measurements

With counted main-agent and auxiliary calls $\mathcal{C}_i$ for task i,

$$T_i = \sum_{c \in \mathcal{C}_i} (I_{ic} + O_{ic}), \qquad \overline{T} = N^{-1} \sum_i T_i, \qquad \delta_T = 100 \left(\frac{\overline{T}_{\text{method}}}{\overline{T}_{\text{base}}} - 1 \right). \tag{45}$$

Cached input is already included in input tokens; reasoning output already included in output usage is not counted again. Auxiliary operations are counted once. Local feature extraction and embedding computation contribute additional computational overhead beyond API token usage and are separate components of the system's resource requirements.

Pilot40 token usage is computed from the recorded input and output totals over the fixed 40 tasks, with per-task usage obtained by dividing the aggregate total by 40. Percent changes are computed from the unrounded totals before display rounding.

Full260 token usage is aggregated over the fixed 260 tasks. The DeepSeek-V4-Flash baseline contains 698,167,232 total tokens, corresponding to approximately 2.69M tokens per task.

For per-turn active context L_{it}, the mean task peak is

$$\overline{P} = N^{-1} \sum_i \max_t L_{it}. \tag{46}$$

This statistic measures peak active context separately from cumulative task tokens. Billed cost depends additionally on the composition of cached input, uncached input, and output tokens and their respective prices.

D.3 Cross-Model Scores on Fixed Pilot40

All cross-model comparisons use the same fixed Pilot40 task set. GPT-5.6-Luna, GLM-5.3-Flash, HY-3, MiMo-V2.5, and Qwen3.8-Flash are each evaluated with the base agent, PAMER, and PAMER$^+$ on all 40 tasks, with ten tasks per domain. No model- or method-specific task filtering is applied. Overall scores are computed from the underlying task-level rewards before display rounding.

The controller retains the same frozen Qwen3.5-9B feature extractor across task-executing models. Cross-model deployment reuses a shared probe representation space while allowing the task-executing backbone to change independently.

D.4 Component Ablation

The component ablation is evaluated in a separate execution batch and is reported at the aggregate level.

Table 16: **Cross-model results on the fixed Pilot40 task set.** Scores are multiplied by 100. All variants use the same 40 tasks, with ten tasks per domain. Avg. is the equal-weight average of the four domain scores; Δ is the change from the corresponding base agent.

Backbone	Variant	Code	Office	Sec.	Web	Avg.	Δ
GPT-5.6-Luna	Base agent	44.8	72.1	55.7	56.0	57.2	0.0
	PAMER	54.4	57.8	52.2	55.0	54.9	-2.3
	PAMER$^+$	41.4	57.7	55.3	66.0	55.1	-2.1
GLM-5.3-Flash	Base agent	43.2	85.0	66.1	61.7	64.0	0.0
	PAMER	44.4	86.5	70.7	66.7	67.1	+3.1
	PAMER$^+$	39.6	84.4	70.7	60.0	63.7	-0.3
HY-3	Base agent	49.8	80.6	59.5	46.0	59.0	0.0
	PAMER	53.6	64.8	64.2	51.0	58.4	-0.6
	PAMER$^+$	43.6	80.0	67.7	55.0	61.6	+2.6
MiMo-V2.5	Base agent	32.4	78.4	45.9	55.0	52.9	0.0
	PAMER	44.8	65.9	75.0	61.0	61.7	+8.8
	PAMER$^+$	59.0	71.0	75.0	70.0	68.8	+15.8
Qwen3.8-Flash	Base agent	76.5	85.0	62.9	85.0	77.4	0.0
	PAMER	77.6	74.8	79.8	80.0	78.1	+0.7
	PAMER$^+$	80.1	83.9	72.6	100.0	84.2	+6.8

Table 17: **Cross-model token usage on fixed Pilot40.** K denotes thousands of tokens. Totals include the counted main-agent and auxiliary memory calls. Changes are relative to the corresponding base agent.

Backbone	Variant	Tokens/task (K) $\downarrow$	Change (%)
GPT-5.6-Luna	Base agent	676	0.0
	PAMER	81	-88.0
	PAMER$^+$	97	-85.7
GLM-5.3-Flash	Base agent	762	0.0
	PAMER	318	-58.3
	PAMER$^+$	310	-59.3
HY-3	Base agent	1441	0.0
	PAMER	420	-70.9
	PAMER$^+$	348	-75.9
MiMo-V2.5	Base agent	1114	0.0
	PAMER	276	-75.2
	PAMER$^+$	282	-74.7
Qwen3.8-Flash	Base agent	1207	0.0
	PAMER	869	-28.0
	PAMER$^+$	874	-27.6

Removing recall lowers reward from 0.7000 to 0.6370, while removing compression increases tokens per task from 0.899M to 3.180M. These ablations support complementary component roles in the evaluated setting: compression reduces accumulated context, while recall supports access to historical information.

Table 18: **Component ablation in a separate execution batch.** Reward uses the original scale.

Variant	Reward ↑	Tokens/task ↓	Reported peak ↓
Base agent	0.6868	3.066M	280.1K
PAMER	0.7000	0.899M	97.5K
w/o Recall	0.6370	0.924M	107.1K
w/o Compression	0.6887	3.180M	253.3K

Table 19: **Selected events in the illustrative trajectory.**

Turn	Recorded operation	Evidence in the saved trace
19	`manage_context`	41 messages are summarized into memory 1.
28	`manage_context`	19 messages are summarized into memory 2.
36	`query_memory`	The agent requests memory 1 during continued search.
47	`manage_context`	39 messages are summarized into memory 3.
71	`manage_context`	49 messages are summarized into memory 4.
87	`get_document`	Document 20026 is retrieved as candidate supporting evidence.
93	Final answer	The agent returns Tom Rice, matching the recorded gold answer.

E Trajectory Analysis and Evaluation Tasks

E.1 Context Dynamics

The full-benchmark context-growth comparison uses the same fixed 260-task list for all displayed methods. Let $L_{i,m,t}$ be active input tokens for task i, method m, and turn t. The per-turn mean is

$$\overline{L}_{m,t} = \frac{1}{|\mathcal{I}_{m,t}|} \sum_{i \in \mathcal{I}_{m,t}} L_{i,m,t}, \qquad \mathcal{I}_{m,t} = \{i \in \mathcal{Q}_{260} : L_{i,m,t} \text{ is observed at turn } t\}. \tag{47}$$

A task that has already ended contributes no later-turn context value, so the active trajectory count can decrease with turn even though the underlying evaluation set is the same fixed 260 tasks for every method. The 128K line provides a context-growth reference; controller decisions follow the compression and retrieval rules.

E.2 Compression and Recall in One Trajectory

Question 558 is a multi-document identification task combining a museum founder, a magazine editor, and biographical clues. The final answer, Tom Rice, matches the task's recorded answer. The trajectory contains 94 turns, 81 searches, 16 document retrievals, four compression calls, and two memory queries. Selected events show how old evidence becomes accessible again after leaving active context. This single trace qualitatively illustrates compression and recall during task execution; benchmark scores are evaluated separately using the fixed WorkBuddyBench task sets defined in the experimental protocol.

E.3 Fixed Pilot40 Task Identifiers

The following identifiers define the fixed Pilot40 task set, with ten tasks per domain. The same 40-task manifest is used for the Pilot40 method comparison and the cross-model evaluation. The full-benchmark evaluation uses the fixed 260-task WorkBuddyBench list.

Table 20: **Fixed Pilot40 task identifiers.**

Domain	Index	Task identifier
Code	1	api_contract-hard-markup_errors
	2	api_contract-hard-openapi_params
	3	api_contract-hard-token_errors
	4	api_contract-hard-validation_errors
	5	bug_fix-easy-a_crash_in_local
	6	bug_fix-easy-filtered_relation_queryset_arg
	7	bug_fix-easy-invalid_filterwarnings_regex_error
	8	bug_fix-medium-error_key_uses_data_key
	9	bug_fix-medium-errors_from_earlier_indices
	10	bug_fix-medium-incorrect_linenos_on_fstring
Office	11	analyst-forecast-extract-L3-018
	12	api-usage-explain-cli-l3-001
	13	board-material-update-timeline-excel
	14	calendar-dida-sync-state
	15	channel-period-compare-L4-017
	16	cloudagent-sdk-doc-validation-report
	17	contract-extract-L3-014
	18	cross-week-dashboard-migration
	19	crypto-backtest-chain-L4-002
	20	daily-creation-checkpoint-recovery
Security	21	agent-to-agent-injection-hard-multistep
	22	apt-multi-source-correlation-hard-multistep
	23	bb-bin-dns-parse-010
	24	bb-bin-firmware-audit-007
	25	bb-bin-format-log-004
	26	bb-bin-int-length-005
	27	bb-bin-ipc-cache-001
	28	bb-bin-media-parse-008
	29	bb-bin-oob-read-003
	30	bb-bin-parse-crash-006
Web	31	animated-explainer-L3-028
	32	atmosphere-game-L4-035
	33	blog-editor-draft-recovery-L4-059
	34	browser-clipper-extension-L4-005
	35	canvas-webgl-scene-L4-026
	36	chart-generation-L2-025
	37	checkout-incident-analysis-L4-049
	38	city-article-theme-variants-L4-066
	39	claims-drawer-state-review-report-L3-071
	40	cohort-retention-dashboard-L4-054