MEMPRICE: ONLINE MEMORY ALLOCATION VIA ADAPTIVE PRICING FOR LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents face a fundamental allocation problem as interaction histories grow: which information to retain, how much context to devote to it, and at what granularity. Existing memory mechanisms largely focus on relevance, retrieval, or compression, leaving this allocation trade-off implicit. We introduce MEMPRICE, which treats active context as a shared resource and dynamically allocates it across memories and representation levels. MEMPRICE turns accumulated memory pressure into an adaptive price on context, encouraging detail only when its estimated retention value justifies the additional footprint. This pressure persists across memory refreshes, allowing the contents of memory to evolve while preserving long-run control over context usage. We show that irreversible compression can force linear worst-case regret against hindsight following its own memory trajectory, and introduce a same-state benchmark to evaluate allocation on MEMPRICE's realized memory states. MEMPRICE enforces the hard context budget at every round and regulates long-run occupancy through adaptive pricing, without prior knowledge of the interaction horizon. Under priced-frontier regularity, its average allocation-regret bound against this benchmark vanishes when accumulated estimation error is sublinear. We evaluate the framework under matched memory budgets on LongMemEval, LoCoMo, AppWorld, and WebShop across three foundation models, spanning both query-blind memory construction and goal-directed agent context management.

1 INTRODUCTION

Foundation models are increasingly deployed as long-horizon agents that reason, act, and interact over extended trajectories (Yao et al., 2023; Shinn et al., 2023). As these interactions grow, the accumulated history can quickly exceed the active context available to the model, making memory management a central component of agent design. Existing systems address this challenge through persistent memory, hierarchical storage, retrieval, summarization, and adaptive context construction (Park et al., 2023; Packer et al., 2023; Xu et al., 2025). More recent methods further predict which past information will matter for upcoming decisions or dynamically compress interaction trajectories to control context growth (Wei et al., 2026; Ye et al., 2026). Together, these advances substantially improve an agent's ability to operate beyond a fixed context window.

However, most existing approaches formulate memory management primarily as a problem of *retrieval, relevance estimation, or compression*, leaving the underlying allocation of context capacity largely implicit. Knowing that a memory is relevant does not determine whether it is worth the space it consumes: a detailed record may provide only slightly more value than a compact summary while occupying substantially more context, and retaining one item necessarily reduces the capacity available to others. Similarity, importance, or predicted relevance scores therefore do not directly capture the opportunity cost of memory occupancy. Moreover, this trade-off changes over time as new information arrives and the user's informational needs evolve. Once information has been compressed or discarded, early allocation mistakes may also be irreversible until the memory is refreshed. We argue that these effects are better understood by treating active context explicitly as a limited resource: rather than only asking *what information is relevant*, an agent must decide *which information deserves memory, at what granularity, and at what cost*.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

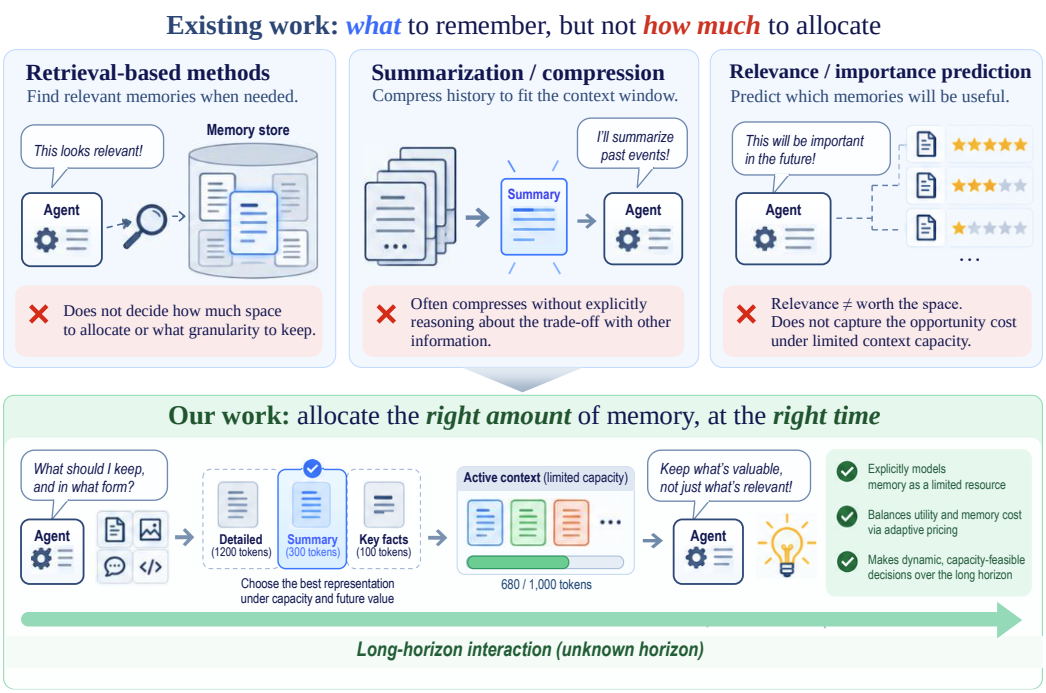

Figure 1: Comparison of existing works and ours.

To this end, we introduce MEMPRICE, a memory-allocation framework that makes the opportunity cost of active context explicit. Each memory can be retained at multiple levels of detail, while a persistent virtual queue turns accumulated occupancy pressure into a shared price on context. MEMPRICE then jointly chooses representations under the hard context budget, retaining finer detail only when its estimated retention value warrants the additional footprint. Crucially, the queue persists across memory refreshes, so retrieval can change what is brought back into context without erasing the resource pressure accumulated over the interaction. Irreversible compression changes which representations remain available, so hindsight may benefit from having preserved different information, not just from better allocation. We therefore evaluate allocation on MEMPRICE's realized memory-state trajectory, while still allowing hindsight to redistribute occupancy across time. MEMPRICE always satisfies hard capacity and, under bounded utilities and estimation error, drives excess average occupancy to zero without knowing the interaction horizon. Under priced-frontier regularity, its average-regret bound against this same-state benchmark also vanishes when estimation error accumulates sublinearly. We evaluate the framework under two complementary regimes: *Query-Blind Memory*, where the eventual question is revealed only after a long interaction, and *Goal-Observable Memory*, where the target is known from the outset and relevant evidence must be accumulated over time. Our main contributions are:

- We formulate long-horizon agent memory management as an explicit online resource-allocation problem, and propose MEMPRICE, which combines capacity-feasible representation selection with persistent memory pricing to adapt allocation over time while decoupling semantic refresh from resource accounting.
- We prove a linear worst-case regret lower bound against independent-state hindsight and introduce a same-state benchmark. Under stated assumptions, MEMPRICE admits regret bounds with explicit estimation-error and priced-frontier terms, and $O(\sqrt{T})$ cumulative occupancy violation.
- We evaluate memory allocation under two complementary settings: *Query-Blind*, where future information needs are unknown during memory construction, and *Goal-Observable*, where the task goal is known from the outset and can guide evidence accumulation. Our method achieves strong results in both settings.

Relation to prior work. Long-horizon agents often rely on persistent memory to preserve and recover useful information over time (Shinn et al., 2023; Park et al., 2023; Packer et al., 2023; Chhikara et al., 2025; Xu et al., 2025). Recent work further adapts the contents and granularity of working context through trajectory folding and relevance-aware context construction (Sun et al., 2025; Ye et al., 2026; Wei et al., 2026). We view active context as a shared resource and formulate memory management as an online allocation problem across competing memories and representation levels, connecting agent memory to online resource allocation and long-term constrained optimization (Devanur & Jain, 2012; Agrawal et al., 2014; Jenatton et al., 2016; Yu et al., 2017; Zong & Jiang, 2026). MemPrice realizes this view through adaptive pricing and persistent resource accounting across memory refreshes. A detailed discussion of related work is provided in Appendix A.

2 Problem Formulation

Agent memory allocation. We study online allocation of an agent's active context over an interaction of unknown length. At each round, the allocator selects representations using only information available so far. Because compression may be irreversible, past decisions can restrict future feasible allocations. The horizon T denotes an arbitrary evaluation prefix and is unknown to the allocator.

Let $B > 0$ denote the hard active-context capacity, and let $\alpha \in (0, 1]$ specify the target average occupancy ratio. At round t, the allocator chooses $a_t \in \mathcal{A}_t(B)$, where $\mathcal{A}_t(B)$ is the feasible set induced by the current memory state. Let $U_t(a_t)$ and $G_t(a_t)$ denote retention utility and memory occupancy, and define $g_t(a_t) = G_t(a_t)/B$. For a non-anticipating policy π, the finite-prefix reference objective is

$$\max_{\pi} \quad \mathbb{E}_\pi\left[\sum_{t=1}^{T} U_t(a_t)\right] \text{s.t.} \quad a_t \in \mathcal{A}_t(B)\ \forall t, \quad \mathbb{E}_\pi\left[\frac{1}{T}\sum_{t=1}^{T} g_t(a_t)\right] \le \alpha, \tag{1}$$

The feasible sets $\mathcal{A}_t(B)$ are induced by the memory-state trajectory generated by π.

Memory representation choices. Let $\mathcal{R}_t$ denote the set of memory items currently resident in active context at round t. Each item can be retained at one of five representation levels,

$$\text{raw} \succ \text{summary (sum)} \succ \text{abstract (abs)} \succ \text{keywords (key)} \succ \text{deletion (del)}$$

ordered from more detailed to more compressed representations. Let $c_i^-(t)$ denote the representation of item i available immediately before the round-t allocation. Between memory refreshes, compression is one-way: the allocator may keep an item's current representation or move to a coarser level, but cannot recover discarded detail. For item i at representation level j, let $s_{i,t}^{(j)}$ denote its memory footprint. Coarser representations use no more memory, with deletion having zero footprint. For an allocation $a = (c_i)_{i\in\mathcal{R}_t}$, its total memory occupancy is $G_t(a) = \sum_{i\in\mathcal{R}_t} s_{i,t}^{(c_i)}$. The feasible allocation set in Eq. equation 1 is therefore

$$\mathcal{A}_t(B) = \left\{a = (c_i)_{i\in\mathcal{R}_t} : c_i \preceq c_i^-(t)\ \forall i \in \mathcal{R}_t,\ G_t(a) \le B\right\}.$$

Let $u_{i,t}^{(j)}$ denote the true retention utility of keeping item i at representation j. We assume that retention utility does not increase under further compression and vanishes upon deletion. At each round, an LLM oracle provides utility estimates $\widehat{u}_{i,t}^{(j)}$ for the feasible representation choices. Let

$$U_t(a) = \sum_{i\in\mathcal{R}_t} u_{i,t}^{(c_i)}, \qquad \widehat{U}_t(a) = \sum_{i\in\mathcal{R}_t} \widehat{u}_{i,t}^{(c_i)}.$$

Throughout the analysis, U_t denotes the allocation-level retention objective; Section 5 evaluates how the resulting memory policies translate to downstream task performance.

Memory refresh. The interaction is organized into consecutive refresh windows of L rounds, except possibly for the final window. Within each window, the resident set $\mathcal{R}_t$ consists of the context carried from the previous boundary together with newly observed interaction items. Motivated by persistent-memory designs that retain interaction history for later retrieval, we store raw transcripts in an external archive and use a fixed compression–retrieval pipeline at refresh boundaries to construct the carried context for the next window. Thus, compression is irreversible within a refresh window but archived information can be recovered at subsequent boundaries. Memory occupancy and retention utility are defined over the active context only.

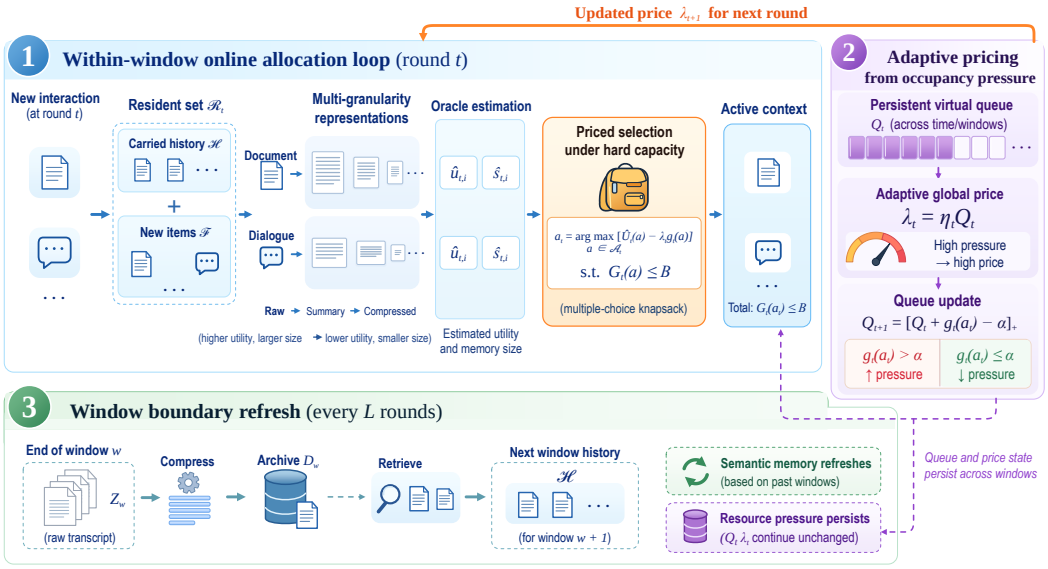

Figure 2: An overview of our MemPrice framework.

3 Algorithm

MemPrice controls active-context allocation through a history-dependent memory price. At each round, accumulated occupancy pressure determines the current memory price. The allocator then jointly selects a representation for each resident memory, retaining additional detail only when its estimated value justifies the corresponding context cost. This pressure carries across memory refreshes, allowing the agent to refresh its active context without forgetting how heavily context has been used. Algorithm 1 summarizes the full procedure.

Adaptive memory pressure. Although the hard capacity B prevents context overflow, enforcing a fixed fraction of B at each round would be unnecessarily rigid: the active context at some rounds contains information that warrants substantially more detailed retention than at others. MemPrice therefore uses αB as a long-run occupancy target, allowing temporary increases in context usage when additional memory detail is valuable. A virtual queue Q_t tracks accumulated occupancy pressure relative to this target and converts it into a memory price:

$$\eta_t = \frac{c}{\sqrt{t}}, \ \lambda_t = \eta_t Q_t, \ Q_{t+1} = [Q_t + g_t(a_t) - \alpha]_+ ,$$

where $[x]_+ = \max\{x, 0\}$ and $c > 0$ controls the strength of the pricing feedback: larger c makes the allocator respond more aggressively to accumulated occupancy pressure. Occupancy above the target increases Q_t, while lighter usage gradually releases this pressure. The resulting memory price λ_t is shared across all resident memories: when pressure is low, detailed representations are relatively cheap; as pressure accumulates, retaining extra detail requires correspondingly higher estimated utility.

Price-aware joint allocation. Given the global price λ_t, MemPrice selects the capacity-feasible representation assignment that maximizes a price-adjusted retention objective:

$$a_t \in \arg\max_{a \in \mathcal{A}_t(B)} \left\{ \widehat{U}_t(a) - \lambda_t g_t(a) \right\}. \tag{2}$$

The price controls how much detail is worth retaining rather than simply filtering memories by relevance. In particular, λ_t / B can be viewed as a per-token context price: additional detail is retained only when its estimated value justifies its extra footprint. Since all resident memories share the same context budget, their representation choices are coupled: a detailed representation may be worthwhile when competition for context is light, but not when more valuable information is competing for the same space. Therefore, Eq. equation 2 becomes a multiple-choice knapsack problem

Algorithm 1 MemPrice: Online Memory Allocation with Adaptive Pricing

Require: Hard capacity B, target occupancy ratio α, pricing constant c, window length L

Initialize round $t \leftarrow 1$, pressure $Q_1 \leftarrow 0$, carried context $H \leftarrow \varnothing$, current-window items $\mathcal{F} \leftarrow \varnothing$
while true do
  for $k = 1, \ldots, L$ **do**
    Observe a new interaction item θ_t, add it to $\mathcal{F}$ at level raw, and form the resident set $\mathcal{R}_t \leftarrow \{H\} \cup \mathcal{F}$
    For each $i \in \mathcal{R}_t$, query the LLM oracle for admissible representations and utility estimates $\{\widehat{u}_{i,t}^{(j)}\}$; record their sizes $\{s_{i,t}^{(j)}\}$
    Set $\eta_t \leftarrow c/\sqrt{t}$ and $\lambda_t \leftarrow \eta_t Q_t$
    Select a_t by solving Eq. 3
    Apply a_t and update

$$Q_{t+1} \leftarrow [Q_t + g_t(a_t) - \alpha]_+$$

    if the conversation ends **then**
      Archive the current-window interaction history and **terminate**
    end if
    $t \leftarrow t + 1$
  end for
  Archive the current-window interaction history and perform MEMORYREFRESH to obtain the carried context H for the next window; set $\mathcal{F} \leftarrow \varnothing$
end while

(MCKP), which jointly selects one admissible representation for each resident memory:

$$(c_i(t))_{i \in \mathcal{R}_t} \in \underset{(c_i)_{i \in \mathcal{R}_t} \in \mathcal{A}_t(B)}{\arg\max} \sum_{i \in \mathcal{R}_t} \left(\widehat{u}_{i,t}^{(c_i)} - \frac{\lambda_t}{B} s_{i,t}^{(c_i)} \right). \tag{3}$$

Persistent memory pressure across refreshes. At each refresh boundary, the raw transcript from the current window is added to the persistent archive, and a fixed compression–retrieval pipeline constructs the carried context for the next window. This refresh can reintroduce information that was previously compressed out of the active context, making earlier detail available again when needed. The refreshed context remains subject to the hard capacity B, while the virtual queue Q_t and global round index t are carried across the boundary.

4 THEORETICAL ANALYSIS

Our analysis distinguishes two sources of hindsight advantage: access to representations preserved along a different memory trajectory, and the ability to redistribute occupancy across time under a global budget. We first characterize the effect of compression-induced state divergence, then evaluate allocation quality on MemPrice's realized memory-state trajectory. To formalize the first effect, consider an *independent-state hindsight comparator*: an optimal policy that observes the full T-round interaction prefix and follows its own memory trajectory under the same fixed refresh pipeline. Its allocations b_t are subject to the hard capacity B at every round and the cumulative occupancy budget $\sum_{t=1}^{T} g_t(b_t) \leq \alpha T$. Let $\mathrm{OPT}_T^{\mathrm{ind}}$ denote its optimal cumulative retention utility, and define

$$\mathrm{Reg}_T^{\mathrm{ind}} := \mathbb{E}\left[\mathrm{OPT}_T^{\mathrm{ind}} - \sum_{t=1}^{T} U_t(a_t) \right].$$

The following proposition establishes a linear worst-case regret lower bound for every capacity-feasible, non-anticipating allocator, even with exact utility estimates.

Proposition 1 (Linear regret under independent-state comparison)**.** *Let $B > 0$, $\alpha \in (0, 1]$, and $M := \lfloor 1/\alpha \rfloor + 1$. For any non-anticipating allocator with $a_t \in \mathcal{A}_t(B)$ for all t and integers $L, T > M$, there exists a T-round instance with exact utility estimates such that*

$$\mathrm{Reg}_T^{\mathrm{ind}} \geq \frac{T - M}{M(M+1)}. \tag{4}$$

Proposition 1 identifies an information-loss barrier rather than an estimation barrier. Before future relevance is revealed, hard capacity forces the online allocator to discard some candidate detail,

whereas hindsight preserves the candidate that will later matter. Importantly, the hindsight policy in the construction uses at most αB at every round, so the lower bound does not rely on redistributing occupancy across time. To isolate the allocation component, we next introduce a same-state comparator that evaluates hindsight decisions on the realized feasible-state trajectory.

For each round t, let $\mathcal{A}_t^{\mathrm{alg}}(B)$ denote the feasible allocation set induced by MemPrice's realized memory state. We define the global same-state (GSS) hindsight benchmark over a T-round prefix as

$$\mathrm{OPT}_T^{\mathrm{gss}} := \max_{b_1,\dots,b_T} \sum_{t=1}^{T} U_t(b_t) \quad \text{s.t.} \quad b_t \in \mathcal{A}_t^{\mathrm{alg}}(B),\ \forall t, \qquad \sum_{t=1}^{T} g_t(b_t) \le \alpha T. \tag{5}$$

The GSS benchmark performs counterfactual reallocation on MemPrice's realized memory states. It retains full hindsight and may redistribute occupancy across the entire prefix, but its alternative choices do not propagate to future states. Thus, GSS regret measures allocation loss conditional on the states reached by MemPrice, rather than the long-term cost of generating those states. The occupancy budget is imposed pathwise on the realized prefix, and Appendix B gives the expectation-feasible counterpart aligned with Eq. 1.

$$\mathrm{Reg}_T^{\mathrm{gss}} = \mathbb{E}\left[\mathrm{OPT}_T^{\mathrm{gss}} - \sum_{t=1}^{T} U_t(a_t)\right], \qquad \mathrm{Viol}_T = B\left[\sum_{t=1}^{T}\big(g_t(a_t) - \alpha\big)\right]_+. \tag{6}$$

To account for errors in the LLM oracle's retention-utility estimates, we make the following bounded-error assumption.

Assumption 1 (Bounded utility and estimation error)**.** There exist finite constants $\overline{U} > 0$ and $\overline{\epsilon} \ge 0$ such that, for every t, almost surely, $0 \le U_t(a) \le \overline{U}$ for all $a \in \mathcal{A}_t^{\mathrm{alg}}(B)$ and $e_t := \sup_{a \in \mathcal{A}_t^{\mathrm{alg}}(B)} |\widehat{U}_t(a) - U_t(a)| \le \overline{\epsilon}$.

We write $\mathcal{E}_T := \sum_{t=1}^{T} \mathbb{E}[e_t]$ for the expected accumulated estimation error. Under a global occupancy budget, hindsight may redistribute memory usage across time. To control this temporal advantage, define the priced frontier

$$\Phi_t(\lambda) := \max_{a \in \mathcal{A}_t^{\mathrm{alg}}(B)} \big\{U_t(a) - \lambda\big(g_t(a) - \alpha\big)\big\}.$$

Assumption 2 (Priced-frontier regularity)**.** There exists a deterministic sequence $\Delta_T = o(T)$ such that, for every T, almost surely,

$$\inf_{\lambda \ge 0} \sum_{t=1}^{T} \left[\Phi_t(\lambda) - \inf_{\mu \ge 0} \Phi_t(\mu)\right] \le \Delta_T.$$

Priced-frontier regularity requires the realized utility–occupancy frontiers to admit an approximately common minimizing price. In particular, $\Delta_T = 0$ whenever the round-wise frontiers share a minimizer, even if the resident memories and optimal allocations vary across rounds. Appendix C gives a sufficient condition allowing dispersion in these minimizing prices. This condition controls the remaining hindsight advantage from reallocating occupancy across time and is not required for hard-capacity feasibility or long-run occupancy control. The following result gives the corresponding same-state allocation and occupancy guarantees.

Theorem 1 (Same-state allocation and occupancy guarantees)**.** *Algorithm 1 maintains $G_t(a_t) \le B$ at every round by construction. Under Assumptions 1 and 2, for any evaluation prefix $T \ge 1$,*

$$\mathrm{Reg}_T^{\mathrm{gss}} \le c\sqrt{T} + 2\mathcal{E}_T + \Delta_T, \qquad \mathrm{Viol}_T \le B\left(\frac{\overline{U} + 2\overline{\epsilon}}{\alpha c}\sqrt{T} + 1\right) \quad \textit{a.s.} \tag{7}$$

Theorem 1 establishes same-state allocation and occupancy guarantees on MemPrice's realized memory-state trajectory. MemPrice never exceeds the hard capacity B, whereas αB acts as a long-run operating target rather than a per-round cap; consequently, the excess average occupancy vanishes at rate $O(T^{-1/2})$. The regret bound decomposes into the online pricing term $c\sqrt{T}$, the accumulated estimation error $\mathcal{E}_T$, and the priced-frontier deviation Δ_T. For fixed c, $\mathcal{E}_T + \Delta_T = o(T)$ yields vanishing average GSS regret, while $\mathcal{E}_T, \Delta_T = O(\sqrt{T})$ gives $O(\sqrt{T})$ cumulative GSS regret. Larger c strengthens occupancy feedback, tightening the $O(\sqrt{T}/c)$ occupancy bound at the cost of a larger $O(c\sqrt{T})$ online pricing term.

Table 1: **Results for Setting I** under matched hard memory budgets. All budget-constrained methods use the same persistent-memory budget, retrieval protocol, and answer model within each backbone block. Best and second-best results among budget-constrained methods are shown in **bold** and underline, respectively. Full Context† is an unconstrained reference and is excluded from ranking.

	LongMemEval-S ($B = 16$k), QA Acc. (%)					LoCoMo ($B = 8$k), F1 (%)			
Method	S-Hop $n = 156$	M-Hop $n = 133$	Temp. $n = 133$	K-Upd. $n = 78$	Overall $n = 500$	S-Hop $n = 841$	M-Hop $n = 282$	Temp. $n = 321$	Overall $n = 1444$
GLM-5.3-Flash									
Full Context†	94.9	80.5	89.5	87.2	88.4	62.5	48.8	53.3	57.8
Recent / FIFO	43.2	24.7	29.1	39.8	34.0	35.2	23.7	27.1	31.2
Uniform Allocation	54.1	39.8	45.2	40.6	45.8	39.4	30.2	29.7	35.4
Content Salience	78.6	64.3	72.8	68.9	71.7	42.8	35.9	34.7	39.7
Dense Salience	72.3	59.4	65.1	62.7	65.5	46.6	37.8	40.5	43.5
LRE-SelfSup	75.2	61.7	69.8	66.4	68.8	**49.9**	41.3	43.0	46.7
MemPrice (Ours)	**80.0**	**66.8**	**79.6**	**77.3**	**76.0**	49.4	**41.9**	**48.8**	**47.8**
DeepSeek-V4.1-Flash									
Full Context†	92.8	81.8	87.4	88.1	87.7	58.9	52.0	48.7	55.3
Recent / FIFO	40.5	27.1	26.8	41.2	33.4	31.9	25.2	26.6	29.4
Uniform Allocation	52.6	44.1	42.8	42.2	46.1	37.5	28.1	31.8	34.4
Content Salience	76.1	66.2	70.4	68.7	70.8	40.9	34.7	39.0	39.3
Dense Salience	69.2	60.5	64.3	59.8	64.1	44.2	39.1	**45.3**	43.4
LRE-SelfSup	73.5	63.4	68.7	66.9	68.5	47.2	40.6	44.8	45.4
MemPrice (Ours)	**77.4**	**73.0**	**73.4**	**75.1**	**74.8**	**47.3**	**45.0**	44.1	**46.1**
Qwen3.8-Flash									
Full Context†	95.4	82.0	90.8	88.2	89.5	63.7	49.9	55.8	59.2
Recent / FIFO	42.4	28.3	26.1	34.5	33.1	34.8	24.7	24.5	30.5
Uniform Allocation	55.3	38.9	49.0	50.2	48.5	38.1	29.6	33.0	35.3
Content Salience	78.2	63.9	73.5	68.4	71.6	43.5	34.2	36.7	40.2
Dense Salience	72.5	59.3	68.2	66.8	67.0	**49.1**	38.5	41.9	45.4
LRE-SelfSup	75.4	62.1	71.0	67.9	69.5	48.8	42.7	42.0	46.1
MemPrice (Ours)	**80.4**	**67.5**	**80.2**	**72.0**	**75.6**	48.5	**43.0**	**48.1**	**47.3**

5 EXPERIMENTS

5.1 EXPERIMENTAL SETTINGS

Experimental settings. We evaluate memory allocation under two information regimes. **Setting I** is *query-blind*: the terminal query is hidden while memory is constructed, and the allocator can use only the chronological history observed so far. We evaluate on **LongMemEval-S** (Wu et al., 2025) and **LoCoMo** (Maharana et al., 2024). We compare with query-blind controls including **Recent/FIFO**, **Uniform Allocation**, **Content Salience**, **Dense Salience**, and **LRE-SelfSup** (Lia & Mazumder, 2026). **Full Context** is reported separately as an unbudgeted reference. **Setting II** is *goal-observable*: the task goal is visible from reset and may guide allocation, while future observations, tool outputs, and evaluator information remain hidden. We evaluate on **AppWorld** (Trivedi et al., 2024) and **WebShop** (Yao et al., 2022), comparing against **ReAct/FIFO History** (Yao et al., 2023), a **Summary Agent**, and **ZipAct-B**, our budget-matched adaptation of ZipAct (Pan et al., 2026). All budgeted methods share the same tokenizer-measured historical-memory capacity, information contract, task split, evaluator, and failure denominator.

We repeat the main comparison with **GLM-5.3-Flash** Zeng et al. (2026), **Qwen3.8-Flash** Qiu et al. (2026), and **DeepSeek-V4.1-Flash** Xu et al. (2026). For each model–dataset pair, all methods use the same reader/agent, prompting interface, tokenizer, and decoding configuration. The historical-memory budget B excludes the current query or goal, current observation, system prompt, tool schemas, and output reserve. Our method uses five ordered representations—raw, summary, abstraction, keywords, and delete—and jointly allocates them with an exact multiple-choice knapsack solver. We set the target occupancy to $\alpha = 0.8$, maintain the persistent pressure state $\lambda_t = cQ_t/\sqrt{t}$, and trigger maintenance at capacity pressure or after at most $L = 10$ interactions. The scale c is selected on development data using occupancy only. We report the primary comparison at $B^\star = B_{\max}/2$ and average all main results over three random seeds.

5.2 MAIN RESULTS

Setting I: query-blind resource allocation Table 1 shows a consistent advantage of MemPrice under matched memory budgets. †Full Context uses the complete history and is reported only as an

Table 2: **Results for Setting II** in the goal-observable setting at $B_{\text{main}} = 8\text{K}$ historical-memory tokens. All budget-constrained methods use the same persistent-memory budget, retrieval protocol, and answer model within each backbone block. Reported results are mean ± SD over three independent seeds. All values are simulated targets rather than observed experimental evidence.

Model	Method	AppWorld TGC (%)		WebShop SR (%)
		Normal	Challenge	
GLM-5.3-Flash	ReAct / FIFO History	35.1 ± 1.9	19.3 ± 1.7	43.6 ± 1.5
	Summary Agent	33.8 ± 2.1	18.1 ± 1.9	40.8 ± 1.7
	ZipAct	$\underline{39.8 \pm 1.5}$	$\underline{23.7 \pm 1.4}$	$\underline{49.6 \pm 1.3}$
	MemPrice (Ours)	$\mathbf{42.2 \pm 1.7}$	$\mathbf{25.8 \pm 1.5}$	$\mathbf{53.1 \pm 1.4}$
DeepSeek-V4.1-Flash	ReAct / FIFO History	36.9 ± 1.9	20.9 ± 1.7	45.8 ± 1.6
	Summary Agent	35.3 ± 1.6	19.4 ± 2.0	43.1 ± 1.8
	ZipAct	$\underline{41.4 \pm 1.4}$	$\underline{25.0 \pm 1.5}$	$\underline{52.0 \pm 1.5}$
	MemPrice (Ours)	$\mathbf{44.0 \pm 1.6}$	$\mathbf{27.0 \pm 1.8}$	$\mathbf{55.8 \pm 1.3}$
Qwen3.8-Flash	ReAct / FIFO History	36.0 ± 1.7	20.1 ± 1.5	46.2 ± 1.5
	Summary Agent	34.7 ± 1.9	18.7 ± 1.7	43.5 ± 1.9
	ZipAct	$\underline{40.4 \pm 1.6}$	$\underline{24.1 \pm 1.3}$	$\underline{50.8 \pm 1.4}$
	MemPrice (Ours)	$\mathbf{42.7 \pm 1.5}$	$\mathbf{26.3 \pm 1.6}$	$\mathbf{55.1 \pm 1.6}$

unconstrained reference; it is excluded from best/second-best ranking. All reported Overall scores are weighted by the corresponding category sizes. On LongMemEval-S, MemPrice improves the strongest constrained baseline across the three backbones and ranks first in every query category. More importantly, the gain is highly non-uniform across query types: improvements are modest on single-hop questions, but become substantially larger on multi-hop, temporal, and knowledge-update queries. This pattern suggests that MemPrice is most useful when memory items compete for capacity over longer dependencies, where simple recency or salience scores are insufficient to capture their future value. LoCoMo exhibits the same trend in a weaker form: MemPrice remains best overall across all backbones, while individual single-hop or temporal subsets can still favor LRE-SelfSup or Dense Salience. This indicates that the benefit comes primarily from difficult allocation decisions rather than uniformly improving retrieval. Finally, MemPrice still trails Full Context, showing that better allocation cannot fully recover information removed by a hard budget; instead, it narrows the performance loss by preserving a more useful subset of the history.

Setting II: goal-observable resource allocation Table 2 shows that MemPrice consistently outperforms all baselines across three backbones and both agent benchmarks, with ZipAct being the strongest competitor in every setting. The improvement is particularly pronounced on WebShop, where MemPrice exceeds ZipAct by 2.7–3.3 points, compared with more moderate gains on AppWorld. This suggests that adaptive memory allocation is especially beneficial for long-horizon interactive tasks, where preserving goal-relevant states and past actions is critical for subsequent decisions. In contrast, the smaller gains on AppWorld Challenge indicate that, as task difficulty increases, failures are increasingly driven by planning and execution rather than memory alone. We also observe that Summary Agent consistently underperforms even FIFO history, suggesting that aggressive summarization can discard fine-grained actionable information that remains important for agent execution. Overall, the consistent gains across backbones indicate that MemPrice improves how a fixed memory budget is allocated rather than relying on a particular underlying model.

5.3 Ablation Study

Table 3 reveals several complementary roles of our design. The largest degradation comes from *Raw/Delete*, especially on LongMemEval-S and LoCoMo, showing that memory management should not be reduced to deciding what to keep or discard; intermediate-fidelity representations are essential for preserving useful evidence under tight budgets. *Fixed Price* and *Reset* Q also hurt more on memory-intensive benchmarks, suggesting that memory pressure is both task-dependent and persistent over time, and therefore benefits from adaptive pricing and a stateful pressure signal. Replacing joint allocation with a greedy rule causes a smaller but consistent loss, indicating that coordination across memory items matters beyond local decisions. Finally, *No Recovery* remains competitive on AppWorld but degrades more on LoCoMo and WebShop, suggesting that recovery

Table 3: Ablation study with GLM-5.3-Flash as the backbone. Each variant removes or replaces a single component of the full method while keeping all other components unchanged. FIFO History is included as a non-ablation reference.

Method	LongMemEval-S QA Acc. (%)	LoCoMo F1 (%)	AppWorld TGC (%)	WebShop SR (%)
Ours (Full)	**76.0**	**47.8**	**42.2**	**53.1**
Fixed Price	70.9	44.6	41.5	50.9
Raw/Delete	34.7	32.6	36.3	45.1
Greedy Allocation	73.5	45.8	40.8	51.7
Reset Q	71.6	45.2	41.7	50.5
No Recovery	74.4	44.1	41.8	50.7
FIFO History	30.4	31.2	35.1	43.6

Table 4: Sensitivity to the target occupancy parameter α at $B^\star$. Task columns report each benchmark's primary quality metric; for AppWorld, we report TGC on Test-Normal. Mean occupancy is the normalized historical-memory load averaged across the four benchmarks. The default is $\alpha = 0.8$.

α	LongMemEval-S QA Acc. (%)	LoCoMo F1 (%)	AppWorld TGC (%)	WebShop SR (%)	Mean Occupancy
0.6	73.2	45.6	40.7	51.4	0.59
0.7	75.1	47.2	41.8	52.7	0.70
0.8	**76.0**	47.8	42.2	**53.1**	0.79
0.9	75.8	**48.0**	**42.3**	53.0	0.88

is particularly useful when earlier compression decisions can become important again later in a long interaction. Overall, the ablations support our view that effective memory management requires not only compression, but adaptive and reversible allocation under evolving memory pressure.

5.4 Sensitivity Analysis

Table 4 studies the target occupancy parameter α, which determines how aggressively the memory allocator frees capacity when memory pressure is encountered. Increasing α from 0.6 to 0.8 consistently improves task quality across all four benchmarks. With a small α, the allocator leaves substantial unused headroom after compression, which reduces future memory pressure but also removes or downsamples useful historical information too aggressively. Allowing the memory to operate at a higher occupancy preserves more relevant evidence and therefore improves downstream performance. Importantly, α also controls a systems-level trade-off between memory utilization and inference cost. A larger α allows the model to use a greater fraction of the available context but leaves less free capacity after each compression event. As new observations accumulate, the memory therefore reaches the allocation boundary again sooner or compress tighter, leading to more frequent compression and reallocation operations. These additional memory-management calls increase auxiliary token consumption and can also increase end-to-end inference latency. Conversely, a smaller α reduces this overhead by maintaining more headroom but sacrifices task quality through more aggressive compression.

6 Conclusions

We study long-term agent memory as a resource-allocation problem under limited memory budgets. Rather than treating memory management as only retention, deletion, or compression, we consider how an agent should allocate limited capacity across historical information with different utility and storage costs. We propose a budget-aware framework that combines adaptive memory pricing, multi-fidelity representations, and joint allocation. Experiments on long-context QA and interactive agent benchmarks show that explicitly managing memory pressure leads to more effective and robust use of limited memory. Our work has several limitations. The current framework relies on estimated utility signals and a predefined set of memory operations, and mainly focuses on memory capacity rather than jointly considering computation, retrieval, and latency.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, LaTeX restructuring, and code-level consistency checks. All AI-assisted material, including mathematical notation, citations, and experimental descriptions, was reviewed by the authors. The authors take responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

The complete proofs of our theoretical results are provided in Appendix B- F. The complete experimental code and reproduction scripts are available in our anonymous repository at `https://anonymous.4open.science/r/ICLR2027-MemPrice`.

REFERENCES

Shipra Agrawal, Zizhuo Wang, and Yinyu Ye. A dynamic near-optimal algorithm for online linear programming. *Operations Research*, 62(4):876–890, 2014.

Ruicheng Ao, Jiashuo Jiang, and David Simchi-Levi. Learning to price with resource constraints: From full information to machine-learned prices. *Advances in Neural Information Processing Systems*, 38:91243–91281, 2026.

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

Nikhil R Devanur and Kamal Jain. Online matching with concave returns. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 137–144, 2012.

Rodolphe Jenatton, Jim Huang, and Cédric Archambeau. Adaptive algorithms for online convex optimization with long-term constraints. In *International Conference on Machine Learning*, pp. 402–411. PMLR, 2016.

Jiashuo Jiang and Jiawei Zhang. Online resource allocation with stochastic resource consumption. *arXiv preprint arXiv:2012.07933*, 2020.

Jiashuo Jiang, Xiaocheng Li, and Jiawei Zhang. Online stochastic optimization with wasserstein-based nonstationarity. *Management Science*, 71(11):9104–9122, 2025.

Xiaocheng Li and Yinyu Ye. Online linear programming: Dual convergence, new algorithms, and regret bounds. *Operations Research*, 70(5):2948–2966, 2022.

Nusrat Jahan Lia and Aritra Mazumder. Learning what not to forget: Long-horizon agent memory from a few kilobytes of learning, 2026. URL `https://arxiv.org/abs/2606.20954`.

Shang Liu, Jiashuo Jiang, and Xiaocheng Li. Non-stationary bandits with knapsacks. *Advances in Neural Information Processing Systems*, 35:16522–16532, 2022.

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. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.747. URL `https://aclanthology.org/2024.acl-long.747/`.

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.

Zhiming Pan, Junyu Luo, Zhiping Xiao, Kaize Ding, Xiao Luo, and Ming Zhang. Zipact: Zipping interaction history into a compact state for efficient LLM agents. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL `https://openreview.net/forum?id=ZssIalqqrz`.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 2023. doi: 10.1145/3586183.3606763.

Zihan Qiu, Zekun Wang, Xiao Li, Yanpeng Li, Yang Xu, Yixuan Wang, Huaqing Zhang, Rui Men, Bochao Mao, Chengruidong Zhang, et al. On the design of qwen3. 8-next architecture: Evaluation, efficiency, and training stability. *arXiv preprint arXiv:2608.30320*, 2026.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling long-horizon llm agent via context-folding. *arXiv preprint arXiv:2510.11967*, 2025.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.850. URL https://aclanthology.org/2024.acl-long.850/.

Lei Wei, Xiao Peng, Tt, Guannan Zhang, Chenhao Jiang, Hongyu Li, Lanbo Lin, Yuanwu Xu, Jiayao Liu, Kesu Wang, and Bin Wang. PACE: Predictive adaptive context extraction for long-horizon LLM agents. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1252. URL https://aclanthology.org/2026.acl-long.1252/.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/d813d324dbf0598bbdc9c8e79740ed01-Abstract-Conference.html.

Anyi Xu, B Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, Bowei Zhang, Boyi Deng, CC Yu, et al. Deepseek-v4. 1-flash: Pushing the limits of kv cache compression. *arXiv preprint arXiv:2609.19969*, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0593.

Shunyu Yao, Howard Chen, John Yang, and Karthik R Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=R9KnuFlvnU.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Wenbiao Yin, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Jingren Zhou, Siheng Chen, and Yong Jiang. Agentfold: Long-horizon web agents with proactive context folding. In *International Conference on Learning Representations*, 2026.

Hao Yu, Michael Neely, and Xiaohan Wei. Online convex optimization with stochastic constraints. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

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.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*, 2023.

Yiming Zong and Jiashuo Jiang. Online semi-infinite linear programming: Efficient algorithms via function approximation. *arXiv preprint arXiv:2603.16200*, 2026.

A Related Works

Agent memory and long-horizon context management. A growing body of work equips language agents with persistent memory beyond the model's immediate context window. Reflexion stores linguistic feedback from previous trials in an episodic memory buffer to improve subsequent decisions (Shinn et al., 2023), while Generative Agents maintain a stream of experiences and retrieve memories according to factors such as relevance, recency, and importance, together with higher-level reflection (Park et al., 2023). MemoryBank further introduces selective reinforcement and forgetting for long-term conversational memory (Zhong et al., 2023). Other systems emphasize memory architecture and organization. MemGPT manages information across hierarchical memory tiers through an operating-system-inspired virtual context mechanism (Packer et al., 2023); Mem0 dynamically extracts, consolidates, and retrieves salient conversational information (Chhikara et al., 2025); and A-MEM organizes memories into an evolving network whose representations and associations can change as new experiences arrive (Xu et al., 2025). These approaches substantially extend the temporal reach of LLM agents, but primarily focus on how memories are stored, organized, updated, or retrieved. In contrast, our work isolates a complementary question: given the information currently available to an agent, how should scarce active-context capacity be allocated among competing memories and representation levels?

Adaptive context selection and compression. Recent work has moved beyond static memory retrieval toward actively controlling the contents and granularity of an agent's working context. Context-Folding allows an agent to branch into temporary sub-trajectories and subsequently fold intermediate reasoning into compact summaries, thereby preventing context growth on long-horizon tasks (Sun et al., 2025). AgentFold similarly treats the context as a dynamic workspace and learns proactive folding operations at different granularities, explicitly addressing the tension between unbounded raw trajectories and irreversible information loss from fixed summarization (Ye et al., 2026). Most closely related to our setting, PACE formulates context construction as a Next Step Prediction problem and adjusts the representation granularity of historical information according to its predicted relevance for the agent's upcoming action (Wei et al., 2026). These methods demonstrate the importance of selectively preserving detail rather than applying uniform truncation or summarization. Our perspective is complementary: rather than primarily predicting which history will be useful next or learning when to fold a trajectory, we explicitly model active context as a shared resource. Memory items with heterogeneous retention value and footprint compete for this resource, and allocation jointly determines both which items remain active and the granularity at which they are represented. This makes the opportunity cost of memory occupancy explicit and enables conversation-level resource control rather than treating context pressure only as a local selection or compression signal.

Online Resource Allocation and Long-Term Constraints. Online resource allocation can be viewed as sequential optimization under intertemporal constraints, where current decisions affect both immediate rewards and future resource availability. A classical approach is to associate resource constraints with dual prices that quantify the marginal value of capacity (Devanur & Jain, 2012). These prices can be updated dynamically as information accumulates (Agrawal et al., 2014; Li & Ye, 2022), and related models have been studied under stochastic resource consumption (Jiang & Zhang, 2020) and non-stationary input distributions (Jiang et al., 2025). Such constrained allocation problems are closely connected to online optimization with long-term constraints, where primal-dual mechanisms balance objective performance and cumulative feasibility. This perspective has been studied in OCO through regret and constraint-violation guarantees (Jenatton et al., 2016;

Yu et al., 2017; Ao et al., 2026; Zong & Jiang, 2026), and also arises in resource-constrained learning problems such as bandits with knapsacks (Liu et al., 2022).

B EXPECTATION-FEASIBLE GLOBAL SAME-STATE BENCHMARK

The main text evaluates MemPrice against a global same-state comparator whose occupancy budget is enforced on each realized evaluation prefix. Here we consider a more permissive expectation-feasible counterpart that matches the average-occupancy constraint in Eq. 1. Let ω denote a realization of all randomness in the interaction and in MemPrice, including any randomness in the oracle and refresh pipeline. For each round t and realization ω, let

$$\mathcal{A}_t^{\mathrm{alg}}(B;\omega)$$

denote the feasible allocation set induced by MemPrice's realized memory state. The memory-state trajectory is fixed to that generated by MemPrice: choices made by the hindsight comparator are evaluated counterfactually and do not alter the feasible sets at later rounds.

Let ζ denote an auxiliary random seed independent of the randomness generating MemPrice's trajectory. A randomized same-state hindsight selector π observes the complete realized prefix and selects

$$b_t^\pi(\omega,\zeta) \in \mathcal{A}_t^{\mathrm{alg}}(B;\omega).$$

Expectations below are taken jointly over ω and ζ. Thus, π may use full hindsight when choosing each b_t^π, but it cannot recover representations that were unavailable on MemPrice's realized trajectory or induce a different future memory state. Let Π_T^{ss} denote the set of all such measurable selectors. We call a selector $\pi \in \Pi_T^{\mathrm{ss}}$ *expectation-feasible* if

$$\mathbb{E}_{\omega,\zeta}\left[\sum_{t=1}^{T} g_t(b_t^\pi)\right] \le \alpha T. \tag{8}$$

The per-round feasible-set constraint holds pathwise, whereas the cumulative occupancy budget in Eq. equation 8 is imposed only in expectation. Hence, the comparator may use more than αT total normalized occupancy on some realizations and less on others, while remaining within the hard capacity at every round.

Define

$$\Pi_T^{\mathrm{egss}} := \left\{\pi \in \Pi_T^{\mathrm{ss}} : \mathbb{E}_{\omega,\zeta}\left[\sum_{t=1}^{T} g_t(b_t^\pi)\right] \le \alpha T\right\}.$$

The expectation-feasible global same-state hindsight value and its associated regret are

$$\mathrm{OPT}_T^{\mathrm{egss}} := \sup_{\pi\in\Pi_T^{\mathrm{egss}}} \mathbb{E}_{\omega,\zeta}\left[\sum_{t=1}^{T} U_t(b_t^\pi)\right], \tag{9}$$

and

$$\mathrm{Reg}_T^{\mathrm{egss}} := \mathrm{OPT}_T^{\mathrm{egss}} - \mathbb{E}_{\omega,\zeta}\left[\sum_{t=1}^{T} U_t(a_t)\right]. \tag{10}$$

We refer to this comparator as EGSS. Every pathwise-feasible same-state selector is expectation-feasible, so this comparator is at least as permissive as the benchmark in the main text.

Since the cumulative occupancy constraint is imposed only in expectation, the corresponding dual comparison requires a deterministic shadow price shared across realizations.

Assumption 3 (Expected priced-frontier regularity)**.** For every T, there exists a deterministic $\lambda_T^\star \ge 0$ such that

$$\mathbb{E}\left[\sum_{t=1}^{T}\left(\Phi_t(\lambda_T^\star) - \inf_{\lambda\ge 0}\Phi_t(\lambda)\right)\right] \le \Delta_T^{\mathrm{E}},$$

where $\Delta_T^{\mathrm{E}} = o(T)$ is deterministic.

Theorem 2 (Expectation-level global guarantee)**.** *Under Assumptions 1 and 3, for any evaluation prefix $T \geq 1$,*

$$\mathrm{Reg}_T^{\mathrm{egss}} \leq c\sqrt{T} + 2\mathcal{E}_T + \Delta_T^{\mathrm{E}}. \tag{11}$$

Moreover, MemPrice satisfies

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} g_t(a_t)\right] \leq \alpha + \frac{\overline{U} + 2\overline{\epsilon}}{\alpha c\sqrt{T}} + \frac{1}{T}. \tag{12}$$

Theorem 2 shows that the main guarantees extend to the more permissive expectation-feasible comparator aligned with Eq. equation 1. If $\mathcal{E}_T + \Delta_T^{\mathrm{E}} = o(T)$, the average EGSS regret vanishes; if both terms are $O(\sqrt{T})$, the cumulative regret remains $O(\sqrt{T})$. Moreover, the excess expected average occupancy above α vanishes as $O(T^{-1/2})$.

Importantly, this extension relaxes only how the long-run occupancy budget is enforced. The comparator remains same-state: it may redistribute context usage across time and stochastic realizations, but cannot undo MemPrice's earlier compression decisions or follow an independent memory trajectory.

C A Sufficient Condition for Priced-Frontier Regularity

We give a simple sufficient condition for Assumption 2 in terms of the dispersion of the round-wise optimal shadow prices.

For each round t, define

$$\Lambda_t^\star := \arg\min_{\lambda \geq 0} \Phi_t(\lambda).$$

The set $\Lambda_t^\star$ is nonempty and contained in $[0, \overline{U}/\alpha]$. Indeed, the all-delete action gives $\Phi_t(\lambda) \geq \alpha\lambda$, while $\Phi_t(0) \leq \overline{U}$. Moreover, since $g_t(a) \in [0,1]$, the function Φ_t is 1-Lipschitz in λ. Therefore, for any $\lambda \geq 0$,

$$\Phi_t(\lambda) - \inf_{\mu \geq 0} \Phi_t(\mu) \leq \mathrm{dist}(\lambda, \Lambda_t^\star),$$

where

$$\mathrm{dist}(\lambda, S) := \inf_{\mu \in S} |\lambda - \mu|.$$

Consequently, Assumption 2 holds whenever there exists a deterministic sequence $R_T = o(T)$ such that, almost surely,

$$\inf_{\lambda \in [0, \overline{U}/\alpha]} \sum_{t=1}^{T} \mathrm{dist}(\lambda, \Lambda_t^\star) \leq R_T.$$

A more concrete sufficient condition is obtained when, for each prefix T, there exists a common price $\bar{\lambda}_T \in [0, \overline{U}/\alpha]$ such that all but $m_T = o(T)$ rounds satisfy

$$\mathrm{dist}(\bar{\lambda}_T, \Lambda_t^\star) \leq \rho_T, \qquad \rho_T = o(1).$$

Then

$$\inf_{\lambda \in [0, \overline{U}/\alpha]} \sum_{t=1}^{T} \mathrm{dist}(\lambda, \Lambda_t^\star) \leq (T - m_T)\rho_T + m_T \frac{\overline{U}}{\alpha} = o(T),$$

and hence Assumption 2 follows. Thus, priced-frontier regularity permits a sublinear number of atypical rounds while requiring the optimal memory shadow prices on the remaining rounds to concentrate around a common value.

D Proof of Proposition 1

Fix $B > 0$ and $\alpha \in (0, 1]$, and write

$$q := \left\lfloor \frac{1}{\alpha} \right\rfloor, \qquad M := q + 1.$$

Let $L, T > M$ be integers, and fix any possibly randomized, non-anticipating allocator satisfying $a_t \in \mathcal{A}_t(B)$ at every round. The fixed, non-anticipating refresh pipeline is the same for the online allocator and the hindsight comparator. Partition the T-round prefix into windows of lengths $n_1, \ldots, n_W$, with $n_w = L$ for $w < W$, $1 \le n_W \le L$, and $T = \sum_{w=1}^{W} n_w$. We construct a family of instances $\{\mathcal{I}_{\boldsymbol{z}} : \boldsymbol{z} \in \{1, \ldots, M\}^W\}$, where z_w identifies the candidate that becomes relevant in window w. We choose exogenous arrivals, so allocation decisions change memory states but not the subsequent interaction sequence. All raw candidate records and all auxiliary arrivals other than the target-revelation signals are fixed independently of $\boldsymbol{z}$. A window's target affects the interaction only through its revelation and the subsequent utility profiles in that window.

A common prefix in each window. In the first $\min\{M, n_w\}$ rounds of window w, one new candidate arrives per round. Candidates have distinct identities across windows, and their raw contents are absent from earlier transcripts. Throughout its arrival window, each candidate has raw size αB; all non-raw representations have zero size and utility. This is a special case of the five-level representation model and satisfies its monotonicity requirements.

All candidates have zero utility through local round M. If $n_w > M$, the interaction reveals z_w at round $M + 1$ without reproducing the candidate's raw contents. Its raw representation then has utility one until the window ends, while all other representations have utility zero. Carried history and items from earlier windows have zero utility throughout the current window. Other arrivals have zero utility and never reproduce a raw candidate. The oracle is exact, and $0 \le U_t(a) \le 1$ for every feasible allocation on either policy's memory trajectory.

The arrivals and oracle profiles in the first $\min\{M, n_w\}$ rounds do not depend on z_w. Thus the allocator receives no information about the target before its revelation. Allocation within a window only selects representation levels for existing resident items; it does not introduce new copies of discarded candidates. Together with the irreversible representation constraint and the absence of within-window archive access, this prevents recovery of a lost raw representation before the window ends. The allocator may carry its full internal state across boundaries; no reset is imposed in the construction.

Feasibility and value of the hindsight benchmark. In each window with $n_w > M$, the hindsight policy deletes all carried and non-target items and retains candidate z_w in raw form from its arrival until the window ends. In a window with $n_w \le M$, it deletes all items. Its post-allocation occupancy satisfies

$$G_t^\star \le \alpha B \le B, \qquad t = 1, \ldots, T, \qquad \sum_{t=1}^{T} g_t^\star \le \alpha T.$$

This policy uses the same fixed refresh pipeline: the returned context is deleted at the next allocation.

The policy earns $(n_w - M)_+$ in window w. No policy can earn more, since each round after the first M has utility at most one and all earlier rounds have utility zero. Writing $\mathrm{OPT}_T^{\mathrm{ind}}(\mathcal{I}_{\boldsymbol{z}})$ for the independent-state hindsight optimum, we obtain

$$\mathrm{OPT}_T^{\mathrm{ind}}(\mathcal{I}_{\boldsymbol{z}}) = \sum_{w=1}^{W} (n_w - M)_+. \tag{13}$$

This value holds for every realization of the boundary pipeline.

Online utility under fresh uncertainty. For the averaging argument, draw $Z_1, \ldots, Z_W$ independently and uniformly from $\{1, \ldots, M\}$. Let ξ be an exogenous random tape driving the allocator and the fixed boundary pipeline, sampled independently of $\boldsymbol{Z} = (Z_1, \ldots, Z_W)$. Conditional on ξ, their responses are deterministic functions of the information available to them.

Consider a window with $n_w > M$. Let S_w be the set of its candidates retained in raw form immediately after the local round-M allocation. Since $q\alpha B \le B < (q+1)\alpha B$, hard feasibility gives $|S_w| \le q$. By non-anticipation and the common observable prefix, S_w is determined by $(\xi, Z_1, \ldots, Z_{w-1})$ and does not depend on Z_w. Hence

$$\mathbb{P}(Z_w \in S_w \mid \xi, Z_1, \ldots, Z_{w-1}) = \frac{|S_w|}{M} \le \frac{q}{M}.$$

The allocator earns positive utility after local round M only if $Z_w \in S_w$, and even then earns at most $n_w - M$. Therefore,

$$\mathbb{E}_{\boldsymbol{Z},\xi}\left[\sum_{k=1}^{n_w} U_{w,k}(a_{w,k})\right] \le \frac{q}{M}(n_w - M)_+.$$

The same inequality holds trivially when $n_w \le M$. Summing over windows yields

$$\mathbb{E}_{\boldsymbol{Z},\xi}\left[\sum_{t=1}^{T} U_t(a_t)\right] \le \frac{q}{M}\sum_{w=1}^{W}(n_w - M)_+.$$

A fixed hard instance over the full interaction. For a fixed target vector $\boldsymbol{z}$, define

$$\rho(\boldsymbol{z}) := \mathbb{E}_{\xi}\left[\mathrm{OPT}_T^{\mathrm{ind}}(\mathcal{I}_{\boldsymbol{z}}) - \sum_{t=1}^{T} U_t(a_t)\right].$$

Combining the preceding inequality with equation 13 gives

$$\frac{1}{M^W}\sum_{\boldsymbol{z}\in\{1,\dots,M\}^W} \rho(\boldsymbol{z}) \ge \left(1 - \frac{q}{M}\right)\sum_{w=1}^{W}(n_w - M)_+ = \frac{1}{M}\sum_{w=1}^{W}(n_w - M)_+.$$

Consequently, there exists a fixed vector $\boldsymbol{z}^\star$ such that, on $\mathcal{I}_{\boldsymbol{z}^\star}$,

$$\mathrm{Reg}_T^{\mathrm{ind}} = \rho(\boldsymbol{z}^\star) \ge \frac{1}{M}\sum_{w=1}^{W}(n_w - M)_+. \tag{14}$$

The vector $\boldsymbol{z}^\star$ may depend on the allocator and T, but not on the realized random seed.

A linear bound in the interaction length. Write $T = KL + r$, where $K = \lfloor T/L \rfloor$ and $0 \le r < L$. Then

$$\sum_{w=1}^{W}(n_w - M)_+ = K(L - M) + (r - M)_+.$$

Since $L \ge M + 1$,

$$L - M \ge \frac{L}{M+1}, \qquad (r - M)_+ \ge \frac{r - M}{M+1}.$$

Substituting into equation 14 gives

$$\mathrm{Reg}_T^{\mathrm{ind}} \ge \frac{K(L-M) + (r-M)_+}{M} \ge \frac{KL + r - M}{M(M+1)} = \frac{T - M}{M(M+1)},$$

which proves equation 4. For fixed α, M is constant; in particular, for $T \ge 2M$,

$$\mathrm{Reg}_T^{\mathrm{ind}} \ge \frac{T}{2M(M+1)}.$$

Thus the worst-case expected regret is $\Omega_\alpha(T)$.

E Proof of Theorem 1

Fix an evaluation prefix T and work on the probability-one event on which Assumptions 1 and 2 hold up to T. Set $z_t := g_t(a_t) - \alpha$. Since $g_t(a_t) \in [0,1]$ and $\alpha \in (0,1]$, we have $z_t \in [-\alpha, 1-\alpha] \subseteq [-1,1]$.

Hard-capacity feasibility. Along MemPrice's realized trajectory, $\mathcal{A}_t^{\mathrm{alg}}(B) = \mathcal{A}_t(B)$, and Algorithm 1 selects $a_t \in \mathcal{A}_t(B)$ at every round. Hence

$$G_t(a_t) \le B, \qquad t = 1, \dots, T.$$

Thus hard-capacity feasibility holds for every execution.

Global same-state comparison. Recall the priced frontier

$$\Phi_t(\lambda) := \max_{a \in \mathcal{A}_t^{\mathrm{alg}}(B)} \{U_t(a) - \lambda\big(g_t(a) - \alpha\big)\}.$$

For any $\lambda \geq 0$ and any feasible sequence $(b_t)_{t=1}^T$ in the global same-state benchmark,

$$\begin{aligned}\sum_{t=1}^T U_t(b_t) &= \sum_{t=1}^T \big[U_t(b_t) - \lambda\big(g_t(b_t) - \alpha\big)\big] + \lambda \sum_{t=1}^T \big(g_t(b_t) - \alpha\big) \\ &\leq \sum_{t=1}^T \Phi_t(\lambda),\end{aligned}$$

where the inequality uses $\sum_{t=1}^T g_t(b_t) \leq \alpha T$. For the fixed prefix T, let $\lambda_T^\star$ be a common price satisfying Assumption 2, and it holds that

$$\mathrm{OPT}_T^{\mathrm{gss}} \leq \sum_{t=1}^T \Phi_t(\lambda^\star). \tag{15}$$

At round t, exact maximization of the estimated priced objective and Assumption 1 imply

$$U_t(a_t) - \lambda_t\big(g_t(a_t) - \alpha\big) \geq \Phi_t(\lambda_t) - 2e_t. \tag{16}$$

Indeed, the additive term $\lambda_t \alpha$ is independent of the action, and the definition of e_t implies that replacing estimated utilities by true utilities loses at most $2e_t$.

Combining equation 15 and equation 16 gives, pathwise,

$$\begin{aligned}\mathrm{OPT}_T^{\mathrm{gss}} - \sum_{t=1}^T U_t(a_t) &\leq \sum_{t=1}^T \big[\Phi_t(\lambda^\star) - \Phi_t(\lambda_t)\big] - \sum_{t=1}^T \lambda_t z_t + 2\sum_{t=1}^T e_t \\ &\leq \Delta_T - \sum_{t=1}^T \lambda_t z_t + 2\sum_{t=1}^T e_t. \end{aligned} \tag{17}$$

where the second inequality follows from $\Phi_t(\lambda_t) \geq \inf_{\lambda \geq 0} \Phi_t(\lambda)$ and Assumption 2.

Global queue drift. The queue recursion gives

$$Q_{t+1}^2 = [Q_t + z_t]_+^2 \leq Q_t^2 + 2Q_t z_t + z_t^2.$$

Using $\lambda_t = \eta_t Q_t$ and $z_t^2 \leq 1$,

$$-\lambda_t z_t \leq \frac{\eta_t}{2}\big(Q_t^2 - Q_{t+1}^2\big) + \frac{\eta_t}{2}.$$

Since $\eta_t = c/\sqrt{t}$ is nonincreasing and $Q_1 = 0$,

$$\begin{aligned}\sum_{t=1}^T \eta_t\big(Q_t^2 - Q_{t+1}^2\big) &= \eta_1 Q_1^2 + \sum_{t=2}^T (\eta_t - \eta_{t-1}) Q_t^2 - \eta_T Q_{T+1}^2 \\ &\leq 0.\end{aligned}$$

Therefore,

$$-\sum_{t=1}^T \lambda_t z_t \leq \frac{1}{2}\sum_{t=1}^T \eta_t = \frac{c}{2}\sum_{t=1}^T \frac{1}{\sqrt{t}} \leq c\sqrt{T}. \tag{18}$$

Combining equation 17 and equation 18 yields

$$\mathrm{OPT}_T^{\mathrm{gss}} - \sum_{t=1}^T U_t(a_t) \leq c\sqrt{T} + 2\sum_{t=1}^T e_t + \Delta_T.$$

Taking expectations and using $\mathcal{E}_T = \sum_{t=1}^T \mathbb{E}[e_t]$ gives

$$\mathrm{Reg}_T^{\mathrm{gss}} \leq c\sqrt{T} + 2\mathcal{E}_T + \Delta_T. \tag{19}$$

Controlling occupancy through the all-delete action. Let a_t^0 denote the all-delete allocation. It is feasible and satisfies

$$U_t(a_t^0) = 0, \qquad g_t(a_t^0) = 0.$$

Exact maximization of the estimated priced objective gives

$$\widehat{U}_t(a_t) - \lambda_t g_t(a_t) \geq \widehat{U}_t(a_t^0).$$

Assumption 1 therefore implies

$$\begin{aligned}\lambda_t g_t(a_t) &\leq \widehat{U}_t(a_t) - \widehat{U}_t(a_t^0) \\ &\leq U_t(a_t) - U_t(a_t^0) + 2e_t \\ &\leq \overline{U} + 2\overline{\epsilon} \qquad \text{a.s.}\end{aligned}$$

Set

$$C_U := \overline{U} + 2\overline{\epsilon}, \qquad \Theta_t := \frac{C_U}{\alpha\eta_t} = \frac{C_U}{\alpha c}\sqrt{t}.$$

The sequence Θ_t is nondecreasing. Whenever $Q_t > \Theta_t$,

$$\lambda_t = \eta_t Q_t > \frac{C_U}{\alpha},$$

and hence

$$g_t(a_t) \leq \frac{C_U}{\lambda_t} < \alpha.$$

Thus $Q_{t+1} \leq Q_t$ whenever $Q_t > \Theta_t$.

We claim that

$$Q_{t+1} \leq \Theta_t + 1, \qquad t \geq 1. \tag{20}$$

For $t = 1$,

$$Q_2 = [g_1(a_1) - \alpha]_+ \leq 1 \leq \Theta_1 + 1.$$

Suppose $t \geq 2$ and $Q_t \leq \Theta_{t-1} + 1$. If $Q_t \leq \Theta_t$, then

$$Q_{t+1} \leq Q_t + 1 \leq \Theta_t + 1.$$

If $Q_t > \Theta_t$, then

$$Q_{t+1} \leq Q_t \leq \Theta_{t-1} + 1 \leq \Theta_t + 1.$$

This proves equation 20, and in particular

$$Q_{T+1} \leq \frac{\overline{U} + 2\overline{\epsilon}}{\alpha c}\sqrt{T} + 1. \tag{21}$$

Cumulative occupancy violation. Since $[x]_+ \geq x$,

$$Q_{t+1} \geq Q_t + z_t.$$

Summing from $t = 1$ to T and using $Q_1 = 0$ gives

$$Q_{T+1} \geq \sum_{t=1}^{T} z_t.$$

Since $Q_{T+1} \geq 0$,

$$\left[\sum_{t=1}^{T} z_t\right]_+ \leq Q_{T+1}.$$

Therefore, by equation 21,

$$\begin{aligned}\mathrm{Viol}_T &= B\left[\sum_{t=1}^{T} z_t\right]_+ \\ &\leq BQ_{T+1} \\ &\leq B\left(\frac{\overline{U} + 2\overline{\epsilon}}{\alpha c}\sqrt{T} + 1\right)\end{aligned}$$

almost surely. This proves all claims of Theorem 1.

F Proof of Theorem 2

Fix an evaluation prefix T, and write

$$z_t := g_t(a_t) - \alpha.$$

Expectation-feasible hindsight comparison. Let $\pi \in \Pi_T^{\mathrm{egss}}$ be any expectation-feasible same-state hindsight selector. By definition,

$$b_t^\pi(\omega, \zeta) \in \mathcal{A}_t^{\mathrm{alg}}(B; \omega)$$

for every round t almost surely, and

$$\mathbb{E}\left[\sum_{t=1}^{T} g_t(b_t^\pi)\right] \le \alpha T.$$

For the deterministic price $\lambda_T^\star$ in Assumption 3,

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} U_t(b_t^\pi)\right] &= \mathbb{E}\left[\sum_{t=1}^{T} \big(U_t(b_t^\pi) - \lambda_T^\star\big(g_t(b_t^\pi) - \alpha\big)\big)\right] \\ &\quad + \lambda_T^\star \mathbb{E}\left[\sum_{t=1}^{T} \big(g_t(b_t^\pi) - \alpha\big)\right] \\ &\le \mathbb{E}\left[\sum_{t=1}^{T} \Phi_t(\lambda_T^\star)\right].\end{aligned}$$

The last inequality uses $\lambda_T^\star \ge 0$, expectation feasibility, and $b_t^\pi \in \mathcal{A}_t^{\mathrm{alg}}(B)$ almost surely. Taking the supremum over $\pi \in \Pi_T^{\mathrm{egss}}$ therefore gives

$$\mathrm{OPT}_T^{\mathrm{egss}} \le \mathbb{E}\left[\sum_{t=1}^{T} \Phi_t(\lambda_T^\star)\right]. \tag{22}$$

Online frontier comparison. At each round, MemPrice exactly maximizes

$$\widehat{U}_t(a) - \lambda_t g_t(a)$$

over $\mathcal{A}_t^{\mathrm{alg}}(B)$. Under Assumption 1, the same argument as in the proof of Theorem 1 gives, almost surely,

$$U_t(a_t) - \lambda_t\big(g_t(a_t) - \alpha\big) \ge \Phi_t(\lambda_t) - 2e_t \qquad \text{a.s.} \tag{23}$$

Equivalently,

$$U_t(a_t) \ge \Phi_t(\lambda_t) + \lambda_t z_t - 2e_t.$$

Summing over t, taking expectations, and using $\mathcal{E}_T = \sum_{t=1}^{T} \mathbb{E}[e_t]$ yields

$$\mathbb{E}\left[\sum_{t=1}^{T} U_t(a_t)\right] \ge \mathbb{E}\left[\sum_{t=1}^{T} \Phi_t(\lambda_t)\right] + \mathbb{E}\left[\sum_{t=1}^{T} \lambda_t z_t\right] - 2\mathcal{E}_T. \tag{24}$$

Combining Eqs. equation 22 and equation 24,

$$\mathrm{Reg}_T^{\mathrm{egss}} \le \mathbb{E}\left[\sum_{t=1}^{T} \big(\Phi_t(\lambda_T^\star) - \Phi_t(\lambda_t)\big)\right] - \mathbb{E}\left[\sum_{t=1}^{T} \lambda_t z_t\right] + 2\mathcal{E}_T. \tag{25}$$

Since

$$\Phi_t(\lambda_t) \ge \inf_{\lambda \ge 0} \Phi_t(\lambda),$$

Assumption 3 implies

$$\mathbb{E}\left[\sum_{t=1}^{T} \big(\Phi_t(\lambda_T^\star) - \Phi_t(\lambda_t)\big)\right] \le \Delta_T^{\mathrm{E}}. \tag{26}$$

Queue drift. The queue recursion

$$Q_{t+1} = [Q_t + z_t]_+$$

implies, pathwise,

$$Q_{t+1}^2 \le Q_t^2 + 2Q_t z_t + z_t^2.$$

Since $z_t^2 \le 1$, $\lambda_t = \eta_t Q_t$, and $\eta_t = c/\sqrt{t}$,

$$-\lambda_t z_t \le \frac{\eta_t}{2}\big(Q_t^2 - Q_{t+1}^2\big) + \frac{\eta_t}{2}.$$

Because η_t is nonincreasing and $Q_1 = 0$,

$$\sum_{t=1}^{T} \eta_t \big(Q_t^2 - Q_{t+1}^2\big) \le 0.$$

Hence, almost surely,

$$-\sum_{t=1}^{T} \lambda_t z_t \le \frac{1}{2}\sum_{t=1}^{T} \eta_t \le c\sqrt{T}.$$

Taking expectations preserves the bound:

$$-\mathbb{E}\left[\sum_{t=1}^{T} \lambda_t z_t\right] \le c\sqrt{T}. \tag{27}$$

Combining Eqs. equation 25, equation 26, and equation 27 gives

$$\mathrm{Reg}_T^{\mathrm{egss}} \le c\sqrt{T} + 2\mathcal{E}_T + \Delta_T^{\mathrm{E}},$$

which proves Eq. equation 11.

Expected average occupancy. It remains to establish Eq. equation 12. As in the proof of Theorem 1, comparison with the all-delete allocation and Assumption 1 give

$$\lambda_t g_t(a_t) \le \overline{U} + 2\bar{\epsilon}.$$

Let

$$C_U := \overline{U} + 2\bar{\epsilon}, \qquad \Theta_t := \frac{C_U}{\alpha \eta_t} = \frac{C_U}{\alpha c}\sqrt{t}.$$

Whenever $Q_t > \Theta_t$,

$$\lambda_t = \eta_t Q_t > \frac{C_U}{\alpha},$$

and therefore

$$g_t(a_t) \le \frac{C_U}{\lambda_t} < \alpha.$$

Thus $Q_{t+1} \le Q_t$ whenever $Q_t > \Theta_t$. The same induction as in Theorem 1 gives

$$Q_{T+1} \le \frac{\overline{U} + 2\bar{\epsilon}}{\alpha c}\sqrt{T} + 1 \qquad \text{a.s.}$$

Moreover, since

$$Q_{t+1} = [Q_t + z_t]_+ \ge Q_t + z_t,$$

summing over t gives

$$\sum_{t=1}^{T} \big(g_t(a_t) - \alpha\big) = \sum_{t=1}^{T} z_t \le Q_{T+1}.$$

Hence, almost surely,

$$\frac{1}{T}\sum_{t=1}^{T} g_t(a_t) \le \alpha + \frac{\overline{U} + 2\bar{\epsilon}}{\alpha c\sqrt{T}} + \frac{1}{T}.$$

Taking expectations yields

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} g_t(a_t)\right] \le \alpha + \frac{\overline{U} + 2\bar{\epsilon}}{\alpha c\sqrt{T}} + \frac{1}{T},$$

which proves Eq. equation 12.

G Implementation Details

G.1 Detailed Experimental Settings

Information regimes. Our experiments distinguish two memory-allocation regimes according to when the downstream information need becomes observable. In **Setting I**, memory is constructed without access to the terminal query. For a chronological history H, the allocator produces a bounded memory state before the target question is released. Target questions, answers, evidence annotations, question types, and evaluation feedback are therefore unavailable to both the allocator and its utility estimator during memory construction. This setting evaluates whether a memory policy can preserve information that is useful for unknown future requests. In **Setting II**, the task goal is available from reset and may be used to determine which parts of the interaction history should be retained. Future observations, tool outputs, environment states, and evaluator-only information remain unavailable. This setting corresponds to long-horizon agents that know what they are trying to accomplish but must manage an increasing trajectory online.

LongMemEval-S. LongMemEval (Wu et al., 2025) evaluates long-term interactive memory over histories containing multiple temporally separated sessions. We use the **LongMemEval-S** variant as a primary query-blind benchmark. For each history, the complete chronological interaction stream is processed before its associated evaluation questions are released. All questions attached to the same history therefore reuse the same frozen terminal memory state rather than reconstructing memory separately for each question. This prevents the allocator from specializing memory to individual queries. We report the benchmark's official QA accuracy as the primary metric. The larger LongMemEval-M variant is reserved for long-context stress analysis rather than the primary comparison.

LoCoMo. LoCoMo (Maharana et al., 2024) evaluates very long-term conversational memory across extended multi-session dialogues. Its questions cover several forms of retrieval and reasoning, including single-hop factual recall, temporal reasoning, multi-hop questions, and cases requiring information to be integrated across distant portions of a conversation. We expose only the chronological conversation while memory is constructed; evaluation questions and their annotations are revealed after the memory state has been frozen. The complete conversation is treated as the evaluation unit, and all methods receive identical session segmentation and memory budgets. We use normalized partial-match F1 as the primary evaluation metric.

AppWorld. AppWorld (Trivedi et al., 2024) is a stateful interactive benchmark in which an agent operates a collection of applications through structured APIs to satisfy a natural-language goal. Unlike Setting-I benchmarks, the goal is visible before the first action, so memory policies may condition retention decisions on task relevance. The agent does not observe future tool results, future application state, or the benchmark evaluator. Each completed action together with its resulting observation forms a historical interaction unit for memory management. We evaluate separately on the official Test-Normal and Test-Challenge splits. The primary metric is **Task Goal Completion (TGC)**, for which a task is successful only when its benchmark-native state-based tests are satisfied. Scenario Goal Completion and difficulty-based breakdowns are retained as secondary analyses.

WebShop. WebShop (Yao et al., 2022) provides a reproducible simulated e-commerce environment in which an agent must search, inspect, and select products satisfying a user-specified shopping goal. The complete goal is available from reset, while future search results, product pages, and environment transitions are revealed only through the agent's own actions. WebShop therefore provides a second goal-observable setting with a different interaction structure from AppWorld. Historical search and browsing observations are subject to the same bounded memory interface as other Setting-II tasks. We report **Success Rate**, the fraction of tasks receiving the maximum terminal reward, as the primary metric and average terminal reward as a secondary metric.

Setting-I baselines. All Setting-I competitors obey the same query-blind information contract and hard historical-memory budget. **Recent/FIFO** retains history in reverse chronological order until the budget is exhausted and provides a pure recency control (cf. Park et al., 2023). **Uniform Allocation** uses the same candidate representation levels as our method but distributes resources uniformly

across memory items, isolating the effect of adaptive allocation. **Content Salience** ranks memory using query-independent lexical salience, while **Dense Salience** uses a dense semantic representation; both retain the highest-valued content under the common capacity. **LRE-SelfSup** (Lia & Mazumder, 2026) learns to predict the future usefulness of history units without observing the future query. **Full Context** is reported separately as an unbudgeted upper-reference condition and is not included in matched-resource comparisons.

Setting-II baselines. **ReAct/FIFO History** (Yao et al., 2023) uses the standard iterative reasoning-and-action interface while retaining the most recent complete trajectory items under budget. **Summary Agent** periodically compresses accumulated trajectory history into a global summary, providing a conventional long-horizon context compression control. **ZipAct-B** is our resource-matched adaptation of ZipAct (Pan et al., 2026). It preserves ZipAct's recurrent Goal–World–Constraint state update but requires the state visible to the actor to satisfy the same tokenizer-measured hard historical-memory budget B before every action. Thus, differences in task performance cannot be attributed simply to one method receiving a larger actor-visible context.

Historical-memory budget. The resource budget B applies only to historical information made visible to the reader or agent at a decision point. The current query or task goal, current observation, system instructions, tool schemas, and reserved output tokens are handled separately when enforcing the physical context-window constraint. Dataset-specific budget grids are determined without using test outcomes from development-history length statistics and available physical-context headroom. We use four budget levels derived from the corresponding maximum feasible budget $B_{\max}$, and report the primary single-budget comparison at $B^\star = B_{\max}/2$. All methods within a dataset use exactly the same budget values.

Multi-fidelity allocation. Each memory item is represented by five ordered fidelity levels: *raw*, *summary*, *abstraction*, *keywords*, and *delete*. Candidate representations are generated only from the currently available text and cannot restore information already removed from an active representation. Successive compressed representations target 0.6 of their parent length. Given candidate utilities and tokenizer-measured sizes, our method jointly selects one representation per item through an exact multiple-choice knapsack problem subject to the hard capacity B.

The utility estimator obeys the information contract of its setting. For Setting I it evaluates general-purpose future retention utility without access to the terminal query, whereas in Setting II it may condition utility on the task goal already visible to the agent. Utility scores are constrained to $[0, 1]$, delete has zero utility, and candidate utilities are monotonically repaired across decreasing fidelity levels when necessary.

Adaptive pricing and memory maintenance. We target average normalized occupancy $\alpha = 0.8$ and maintain a persistent pressure state

$$Q_{t+1} = \left[Q_t + \frac{G_t}{B} - \alpha \right]_+ ,$$

where G_t denotes the selected historical-memory size. The corresponding adaptive price is

$$\lambda_t = \frac{cQ_t}{\sqrt{t}}.$$

Both Q_t and the global time index persist across memory-maintenance boundaries. The scale c is chosen separately for the two information regimes using development data only, minimizing the deviation between mean occupancy and the target α; downstream task accuracy is never used for this calibration.

Allocation is updated as new memory items are admitted. A maintenance boundary occurs when either active memory reaches capacity or the number of interactions since the previous boundary reaches $L = 10$. At a boundary, complete raw history is first copied to an append-only external archive, the active memory is compressed, and relevant archived chunks may then be recovered into the next active context. The archive is persistent storage and does not itself count toward the reader-visible budget B; any retrieved text does. Recovery uses BM25 with $k_1 = 1.5$, $b = 0.75$, and top-8 retrieval, subject again to the hard active-memory capacity.

Replication. Every main configuration is evaluated with three shared random seeds, namely 2026, 2027, and 2028. The same seed set is used for all competing policies on a given benchmark so that stochastic variation is paired across methods. Language-model decoding itself remains deterministic where supported. We report the mean across the three runs together with dispersion in the corresponding result tables.