ContextBudget: Budget-Aware Context Management for Long-Horizon Search Agents

Yong Wu[1*] **YanZhao Zheng**[2*] **TianZe Xu**[2]

ZhenTao Zhang[2] **YuanQiang Yu**[2] **JiHuai Zhu**[2] **Chao Ma**[2]

BinBin Lin[1†] **BaoHua Dong**[2†] **HangCheng Zhu**[2] **RuoHui Huang**[2] **Gang Yu**[2]

[1]Zhejiang University [2]Alibaba Group

wu.yong@zju.edu.cn, binbinlin@zju.edu.cn
{zhengyanzhao.zyz, xutianze.xtz, zhangzhentao.zzt, yuyuanqiang.yyq,
zhujihuai.zjh, mc524716, baohua.dbh, linran.lr09, wentong,
ruohui.huang, gang.yu}@alibaba-inc.com

Abstract

LLM-based agents show strong potential for long-horizon reasoning, yet their context size is limited by deployment factors (e.g., memory, latency, and cost), yielding a constrained context budget. As interaction histories grow, this induces a trade-off between retaining past information and staying within the context limit. To address this challenge, we propose Budget-Aware Context Management (BACM), which formulates context management as a sequential decision problem with a context budget constraint. It enables agents to assess the available budget before incorporating new observations and decide when and how much of the interaction history to compress. We further develop BACM-RL, an end-to-end curriculum-based reinforcement learning approach that learns compression strategies under varying context budgets. Experiments on compositional multi-objective QA and long-horizon web browsing benchmarks show that BACM-RL consistently outperforms prior methods across model scales and task complexities, achieving over 1.6× gains over strong baselines in high-complexity settings, while maintaining strong advantages as budgets shrink, where most methods exhibit a downward performance trend.

1 Introduction

The growing capabilities of LLM agents and their increasing adoption in long-horizon interactive applications have made effective context management a pressing requirement(Hua et al., 2025; Luo et al., 2025). As agents increasingly operate over extended trajectories, interaction histories rapidly accumulate and cause substantial context expansion(Yao et al., 2023). At the same time, the maximum context window remains strictly bounded by deployment resources such as memory footprint, inference latency, and serving cost(Liu et al., 2024; Packer et al., 2024). Consequently, controlling context growth has emerged as a central concern for sustaining reliable long-horizon reasoning.

Recent agent systems therefore increasingly adopt context compression(Zhou et al., 2025; Wu et al., 2025; Hua et al., 2025), where past observations, intermediate traces, and retrieved evidence are summarized or rewritten during inference. This paradigm has gained prominence due to its empirical effectiveness and conceptual simplicity: it preserves task-critical information under fixed context limits while requiring no external memory or architectural modification(Hu et al., 2026).

[*]Equal contribution
[†]Corresponding author

However, most existing compression methods adopt a budget-free formulation(Zhou et al., 2025; Wu et al., 2025; Hua et al., 2025), treating compression as a static operation without explicit conditioning on the available context budget. This simplification introduces two critical failure modes. Under relaxed budgets, agents may over-compress and erase research-critical evidence, reducing information fidelity(Li et al., 2023; Tang et al., 2025). Under tight budgets, agents may under-compress and overflow the context limit, causing truncation or brittle reasoning failures(Liu et al., 2024; Hsieh et al., 2024).

Recent work has begun to incorporate budget awareness into agentic reasoning, studying constraints such as tool-call budgets and output token budgets (Liu et al., 2025; Tale et al., 2024). These methods highlight that interaction scaling is limited by finite resources in deployment(Snell et al., 2024). However, they primarily regulate budgets through external controls on computation or action counts. They do not systematically address proactive history compression when the context-window budget itself is the limiting resource.

In this paper, we propose Budget-Aware Context Management, a framework enabling LLM agents to perform long-horizon reasoning under explicit context-window budgets. The central idea is to formulate compression in context management as a budget-constrained sequential decision problem, allowing compression decisions to adapt dynamically to remaining context capacity throughout the reasoning process. To achieve this, the framework introduces an explicit budget-awareness signal enabling agents to determine when to compress, how much to compress, and which information to preserve for future reasoning. This budget-conditioned control retains research-critical evidence under strict budgets while avoiding unnecessary information loss when capacity is sufficient.

To operationalize this formulation, we develop a budget-constrained reinforcement learning approach that extends Group Relative Policy Optimization (GRPO) (Shao et al., 2024) to optimize context management end to end. Instead of relying on handcrafted heuristics, the approach directly aligns compression strategies with downstream task success under explicit budget constraints. It further incorporates a progressively tightened budget curriculum and a penalty for budget violations to improve robustness under strict context limits.

Extensive experiments on compositional multi-objective QA and long-horizon web browsing benchmarks demonstrate that the approach consistently improves reasoning robustness across a wide range of context budgets, with the largest gains under stringent budget regimes. Our contributions can be summarized as follows:

- We propose **Budget-Aware Context Management**, a framework that formulates budget-aware context compression for LLM agents as a sequential decision problem under explicit context-window constraints, enabling adaptive compression throughout long-horizon trajectories.
- We develop a budget-constrained RL method that extends GRPO with a progressively tightened budget curriculum and overflow-sensitive regularization for robust context management under strict budgets.
- We conduct extensive empirical studies on compositional multi-objective QA and long-horizon web browsing benchmarks, and further provide comparative analyses of compression behavior and efficiency under context constraints.

2 Related Work

Context Management in LLM Agents. Agent reasoning accumulates prior observations and intermediate reasoning traces in the prompt over successive steps, causing linear context growth that induces the lost in the middle effect and exceeds the effective processing window (Yao et al., 2023; Chen et al., 2023; Liu et al., 2024). To address this, early approaches introduce external memory systems representing context as tiered storage and retrieving relevant history (Packer et al., 2024; Zhong et al., 2023; Kang et al., 2025; Li et al., 2025b). Although these methods extend accessible context, they rely on external memory rather than updating in-context representations during reasoning.

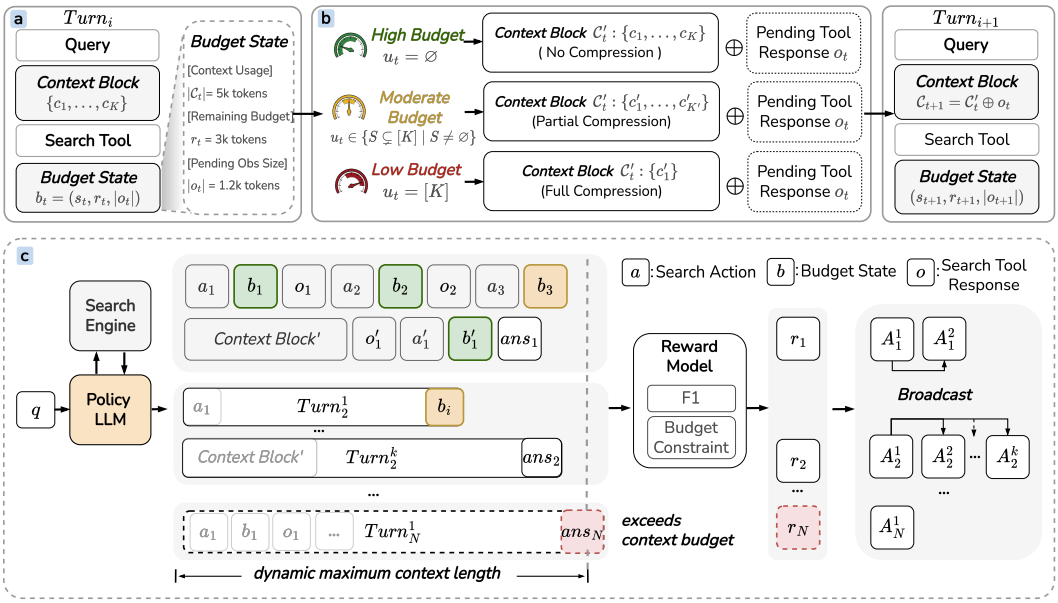

Figure 1: Overview of the proposed framework. (a) The agent first observes the budget-conditioned state $b_t = (s_t, r_t, |o_t|)$ before loading the pending observation. (b) Conditioned on b_t, the policy selects a refinement action u_t to perform NULL, PARTIAL, or FULL commit-block aggregation, yielding an updated context $\mathcal{C}'_t$. (c) The policy is trained with multi-turn GRPO under a progressively tightened budget curriculum, where only trajectories satisfying the context budget contribute reward and optimization.

More recent work formulates context management as a compression problem, condensing observations and reasoning traces to maximize information density under fixed context budgets (Jiang et al., 2023; 2024; Chevalier et al., 2023; Mu et al., 2023). In single-turn settings, compression operates at the input level. In multi-turn settings, it becomes continuous state management, mapping interaction histories into bounded representations, recursively compressing trajectories into higher-level abstractions, and maintaining structured task states for stable reasoning (Wu et al., 2025; Zhou et al., 2025; Yu et al., 2025; Yuan et al., 2025; Chen et al., 2026; Sun et al., 2025; Ye et al., 2025; Zhang et al., 2026). However, existing methods adopt a *budget-free* formulation where compression is treated as a static or periodically triggered operation without conditioning on the available context budget. In contrast, this work models compression as a dynamic process conditioned on the context budget, enabling adaptive trade-offs between retained information and resource constraints.

Budget-aware Inference Scaling. Budget-aware inference scaling studies optimize LLM reasoning under explicit test-time resource constraints (Snell et al., 2024). Prior work mainly controls output length or computation, including token-budgeted reasoning and adaptive token allocation (Li et al., 2023; Tale et al., 2024; Wen et al., 2025; Li et al., 2025a), showing performance depends on computation allocation but is largely limited to single-turn settings. In agent settings, recent work introduces budget-aware control over interaction processes, such as constraining tool use and tracking resource consumption in multi-step reasoning (Liu et al., 2025), showing performance is bounded by finite budgets.

However, these approaches primarily regulate observable interaction behaviors such as action frequency or search strategies, rather than adapting reasoning to varying context-window limits imposed by deployment constraints. Consequently, when the context window becomes the primary bottleneck, existing methods lack mechanisms to adapt context management to the available budget. In contrast, this work formulates context-window management as a budget-aware decision process, enabling agents to dynamically adjust their history representations under real-time budget signals.

3 Budget-Aware Context Management

We propose a budget-aware context management framework for long-horizon search agents under strict context limits. The core idea is to formulate context management as a budget-constrained sequential decision problem. To realize this formulation, we introduce two key mechanisms: (1) a budget-conditioned inference state with deferred observation loading that exposes remaining context headroom before appending new observations (Section 3.1), and (2) a commit-block aggregation mechanism that enables the agent to adaptively control when and how much to compress under varying budget pressures (Section 3.2). We then optimize this decision process using a multi-turn GRPO objective with a progressively tightened context-window curriculum (Section 3.3). Together, these components enable effective long-horizon context allocation under strict budgets.

3.1 Budget-Conditioned State with Deferred Loading

We formulate context management as a budget-aware decision process to avoid both premature information loss from aggressive compression and reasoning failures due to context overflow. Instead of immediately appending new observations, the agent first evaluates available capacity and adjusts its existing context before incorporating additional information. Intuitively, the agent must reorganize its context based on how much space remains and how much space the incoming observation will require.

Formally, we extend the standard MDP with a budget-conditioned state. Let $\mathcal{S}$, $\mathcal{A}$, and B denote the state space, action space, and fixed context budget. At step t, the agent is given an augmented state

$$b_t = (s_t, r_t, |o_t|), \qquad r_t = B - |\mathcal{C}_t|,$$

where B denotes the total context budget, $|\mathcal{C}_t|$ denotes the token length of the current context buffer, and r_t represents the remaining context budget. Here s_t denotes the current reasoning state, and $|o_t|$ denotes the token length of the pending observation. The size of o_t is observable, while its content remains hidden.

Given this state, the agent first determines a refinement action before accessing the new observation. Specifically, the policy samples $u_t \sim \pi_\theta(\cdot \mid b_t)$ to produce an updated context $\mathcal{C}'_t$ that satisfies the budget constraint $|\mathcal{C}'_t| \leq B - |o_t|$. The observation is then appended to form the next context:

$$\mathcal{C}_{t+1} = \mathcal{C}'_t \oplus o_t.$$

As a result, context compression is guided by the future capacity implied by $|o_t|$, rather than a reactive post-processing step. This ordering reserves sufficient capacity for the incoming observation before incorporation, ensuring the context remains within the budget constraint.

3.2 Commit-Block Aggregation for Budget-Aware Context Compression

To support adaptive compression under dynamic budgets, we introduce a commit-block aggregation mechanism. It allows the agent to decide when to compress and how much to reduce based on the current budget. This avoids context overflow and unnecessary information loss. It also integrates compression into the policy instead of treating it as a separate post-processing step.

The mechanism induces three budget-dependent regimes, as illustrated in Figure 1. Under high budget, the policy favors preserving the full interaction history, deferring compression to retain maximal context. Under moderate budget, it shifts toward selective aggregation, compressing redundant segments while preserving salient information. Under low budget, the policy collapses the context into a fully aggregated representation. This keeps it within budget while supporting long-horizon reasoning.

Formally, at step t, the context is a buffer of K coherent segments, denoted by $\mathcal{C}_t = \{c_1, \ldots, c_K\}$. Each segment c_i is a semantically contiguous portion of the interaction history. Conditioned on the budget-aware state $b_t = (s_t, r_t, |o_t|)$, the policy samples a structured

action $u_t \sim \pi_\theta(\cdot \mid b_t)$ from three mutually exclusive categories:

$$u_t \in \underbrace{\{\emptyset\}}_{\text{NULL}} \uplus \underbrace{\{\mathcal{S} \subsetneq [K] \mid \mathcal{S} \neq \emptyset\}}_{\text{PARTIAL}} \uplus \underbrace{\{[K]\}}_{\text{FULL}}.$$

where $[K] = \{1, \ldots, K\}$ denotes the index set of all segments, and $\mathcal{S}$ denotes a subset of segment indices, the selected blocks for aggregation.

These categories correspond to the three cases above and jointly encode both compression timing and intensity. NULL ($u_t = \emptyset$) skips compression when the budget is sufficient. PARTIAL selects a non-empty proper subset $\mathcal{S}$, where $|\mathcal{S}|$ controls the compression strength. FULL ($u_t = [K]$) aggregates all segments under severe budget constraints.

Thus, a single action u_t determines whether and how much to compress. After aggregation, the buffer is updated and the observation appended under the budget constraint. Because π_θ generates both compression decisions u_t and reasoning actions in a unified action space, the agent learns to coordinate context reduction with downstream task performance. The policy therefore learns not only to solve the task under a fixed memory rule, but also to adjust context usage under changing budget conditions.

3.3 Budget-Aware GRPO Objective with Progressive Context Curricula

To optimize the Agent for context management across varying budgets, we employ reinforcement learning under a progressive context budget curriculum. We utilize Group Relative Policy Optimization (GRPO) for sample efficiency without a value critic.

Formally, we define J curriculum stages with monotonically decreasing budgets $B_{\max}^{(1)} \geq \cdots \geq B_{\max}^{(J)}$. For each query, we sample N rollouts. Each rollout i is a full multi-turn interaction with M_i turns, where the model performs context management and generates responses across turns. The final outcome of rollout i, denoted by R_i, is defined as the F1 score between the predicted answer and the ground truth. We assign rewards based on whether the entire rollout satisfies the stage-specific budget constraint. In the j-th curriculum stage, let $|\mathcal{C}_{i,t}|$ denote the size of the managed context at turn t of rollout i. The budget-constrained reward $\tilde{R}_i^{(j)}$ and the group-relative advantage $A_i^{(j)}$ are computed as:

$$\tilde{R}_i^{(j)} = \begin{cases} R_i, & \text{if } \forall t,\ |\mathcal{C}_{i,t}| \leq B_{\max}^{(j)} \\ 0, & \text{otherwise} \end{cases}, \quad A_i^{(j)} = \frac{\tilde{R}_i^{(j)} - \text{mean}\big(\{\tilde{R}_n^{(j)}\}_{n=1}^N\big)}{\text{std}\big(\{\tilde{R}_n^{(j)}\}_{n=1}^N\big) + \epsilon}. \tag{1}$$

Each rollout produces token sequences across multiple turns, denoted as $\{o_{i,m,t}\}_{t=1}^{T_{i,m}}$, where $T_{i,m}$ is the number of tokens generated at turn m in rollout i. We broadcast the trajectory-level advantage $A_i^{(j)}$ to all tokens by assigning the same advantage to each token within the trajectory and optimize the policy at the token level:

$$\mathcal{L}_{\text{PG}}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\sum_{m=1}^{M_i} T_{i,m}} \sum_{m=1}^{M_i} \sum_{t=1}^{T_{i,m}} \min\left(r_{i,m,t} A_i^{(j)}, \text{clip}(r_{i,m,t}, 1-\epsilon, 1+\epsilon) A_i^{(j)}\right) \tag{2}$$

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{PG}}(\theta) - \beta \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \tag{3}$$

where $r_{i,m,t} = \frac{\pi_\theta(o_{i,m,t} | s_{i,m,t})}{\pi_{\text{old}}(o_{i,m,t} | s_{i,m,t})}$ is the token-level probability ratio, and $s_{i,m,t}$ denotes the token state conditioned on the managed context at turn m.

This design provides a clear learning signal: only trajectories that succeed and respect the budget are rewarded, while group-relative normalization stabilizes optimization. By broadcasting trajectory-level advantages to all tokens, the model learns to align local decisions with globally effective context management under progressively tighter constraints.

4 Experiments

This section evaluates Budget-Aware Context Management on compositional multi-objective QA and long-horizon web browsing benchmarks. We introduce the datasets and metrics (Sections 4.1–4.2), followed by the compared methods (Section 4.3). We then present the main results (Section 4.4), including performance under different context budgets and compression behavior, and conclude with ablation studies (Section 4.5).

4.1 Datasets

Existing multi-hop benchmarks are limited in horizon and do not explicitly require context management. Following MEM1 (Zhou et al., 2025), we aggregate multiple independent questions into a single composite query to form multi-objective tasks, increasing interaction length and reasoning complexity and inducing long-horizon trajectories. Consistent with MEM1, we restrict training data construction to 2-objective compositions, and evaluate generalization to larger compositions at test time.

For evaluation, we adopt the Wikipedia-based QA benchmarks used in Search-R1 (Jin et al., 2025), covering single-hop datasets (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023)) and multi-hop datasets (HotpotQA (Yang et al., 2018), 2WikiMultiHopQA (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2022)). Following MEM1 (Zhou et al., 2025), we convert them into multi-objective evaluation sets. We also include BrowseComp-Plus(Chen et al., 2025), a long-horizon benchmark with a verified corpus, to evaluate performance in extended multi-step reasoning settings.

4.2 Evaluation Metrics

For the multi-objective QA benchmarks, following MEM1 (Zhou et al., 2025), we compute token-level F1 between predicted and reference answers for each objective and report the sum of all objectives. For an N-objective task, the total score ranges from 0 to N, measuring answer accuracy and the ability to solve multiple sub-tasks within a single trajectory.

For BrowseComp-Plus, we follow the official LLM-as-a-Judge protocol (Chen et al., 2025), where correctness is determined by comparing model responses with ground-truth answers. Following the benchmark setup, we use Qwen3-32B as the judge model. Under this protocol, disagreement between Qwen3-32B and GPT-4.1 is below 1%, and LLM judgments in BrowseComp-Plus baselines have been human-verified as reliable. We report average accuracy over all samples. Full judge prompts and details are provided in Appendix A.7.

4.3 Compared Methods

To evaluate Budget-Aware Context Management, we compare against the following baselines categorized by their context maintenance strategies:

(1) **Methods without Context Management**: We include ReAct (Yao et al., 2023), a reasoning-and-acting baseline without learned context control, and Search-R1 (Jin et al., 2025), an RL-based search method without explicit context constraints.

(2) **Reactive Context Management**: We employ the Summary Agent (Wu et al., 2025; Yu et al., 2025) as a conservative baseline that invokes summarization only when the context buffer is full.

(3) **Proactive Context Management**: We compare against MEM1 (Zhou et al., 2025), a reinforcement learning approach that maintains a compact, fixed-size context by iteratively consolidating past information into an internal state at each step.

Our method employs a curriculum learning strategy with progressive context budgets to stabilize learning under limited budgets, gradually tightening the context budget from 8k to 4k tokens during training to encourage adaptation as pressure increases. Detailed curriculum schedules are provided in Appendix A.5.2.

Method	Context	BrowseComp-Plus				Multi-objective QA			
		Easy	Mid	Hard	Avg	2-Obj	8-Obj	16-Obj	32-Obj
Qwen3-235B-A22B-Inst-2507									
ReACT	128k	0.380	0.164	0.034	0.136	0.948	1.873	1.315	1.412
ReACT	8k	0.420	0.143	0.011	0.118	0.886	1.782	1.233	0.374
Qwen2.5-7B-Inst									
ReACT	8k	0.320	0.098	0.027	0.089	0.395	0.458	0.376	0.090
Search-R1(RL)‡	8k	0.400	0.111	0.015	0.099	0.760	1.719	2.497	1.022
Summary§	8k	0.340	0.085	0.015	0.078	0.678	2.176	2.379	0.567
MEM1(RL)†	8k	0.040	0.044	0.015	0.035	0.838	2.345	2.391	1.210
BACM (w/o RL)	8k	0.160	0.027	0.015	0.031	0.690	1.698	1.179	0.305
BACM-RL	8k	**0.420**	**0.146**	**0.031**	**0.127**	**0.909**	**2.790**	**4.011**	**2.938**
Qwen3-30B-A3B-Inst									
ReACT	8k	0.440	0.148	0.034	0.130	0.938	2.078	0.931	0.208
Search-R1(RL)‡	8k	0.420	0.150	0.027	0.128	1.013	3.310	1.949	0.998
Summary§	8k	0.480	**0.164**	0.019	0.137	0.916	2.456	2.992	2.848
MEM1‡	8k	0.360	0.141	**0.069**	0.131	0.978	1.327	1.383	0.909
BACM (w/o RL)	8k	0.200	0.100	0.015	0.080	0.474	0.098	0.170	0.208
BACM-RL	8k	**0.520**	**0.164**	0.042	**0.147**	**1.032**	**3.587**	**6.255**	**4.545**

Table 1: Performance comparison across QA settings. BrowseComp-Plus is evaluated using *LLM-as-Judge*, while Multi-object QA reports *F1 scores summed across objectives*.† Reproduced using officially released open-source checkpoints.; ‡ Re-implemented based on released code; § Re-implemented based on the original paper due to no public implementation.

For fair comparison, all methods are evaluated on two backbone models: Qwen2.5-7B-Instruct and Qwen3-30B-A3B-Instruct. We follow Search-R1's official implementation and training data (Appendix A.5.3), and train the model under an 8k context budget for fair comparison. For the Summary Agent, we adopt a reactive summarization paradigm similar to ReSum and MemAgent, triggering summarization when the context window reaches a predefined threshold, the number of compressions is capped at 10 per task. For MEM1, we use the released 7B checkpoint and evaluation framework, adapting compression tags for the 30B backbone (Appendix A.5.4). Our method uses MEM1's multi-objective training data and limits compression to 10 operations per task. Additional details are provided in Appendix A.5. We plan to release our training code in future work to support reproducibility and facilitate follow-up research.

4.4 Main Results

Performance Comparison. As shown in Table 1, BACM-RL achieves the best average performance on the BrowseComp-Plus and Multi-objective QA benchmarks. This holds across two backbone models with major architectural and parameter differences (7B and 30B), showing broad applicability. Compared to MEM1, the strongest prior baseline trained on the same data, BACM-RL achieves especially large gains in harder settings, including an approximate 5.0× improvement in the 32-objective regime (4.545 vs. 0.909). This robustness extends to the out-of-domain BrowseComp-Plus benchmark, where our 30B variant, limited to an 8k budget, achieves 0.147 accuracy and surpasses the 235B Qwen3-Inst model (0.136) with a 128k context window. In the more context-demanding 32-objective regime, the large gap over its non-RL ablation (4.545 vs. 0.208) supports the effectiveness of our budget-aware GRPO objective design. By adopting a curriculum that progressively tightens the context budget, our framework guides the model to operate under increasingly strict constraints, encouraging local compression decisions that improve overall performance.

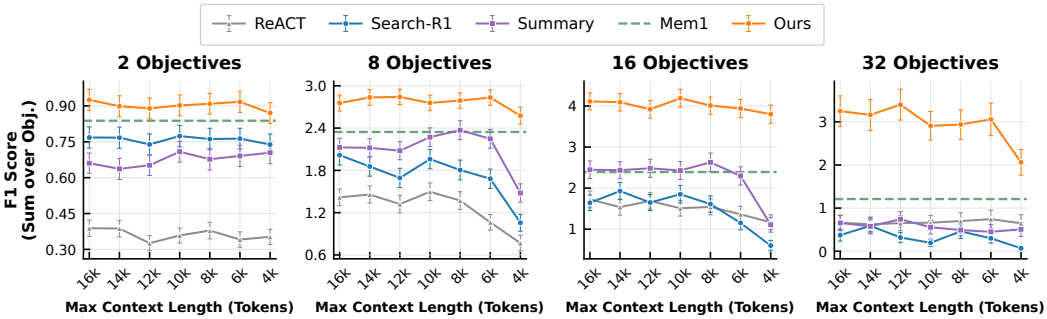

Figure 2: Performance under different maximum context window sizes (16k-4k tokens) with varying numbers of objectives.

Robustness to Varying Context Budgets Figure 2 demonstrates the method's superiority and stability under all task complexities and extensive context budget settings. Unlike context-management-free baselines (ReACT, RL-based Search-R1), which compete only on simple tasks with sufficient budgets, our approach remains nearly invariant under budget reductions from 16k to 4k tokens. While budget-free strategies like MEM1 and Summary partially alleviate degradation, they remain inferior. MEM1 incurs information loss via turn-by-turn over-compression, while Summary suffers reasoning failures by delaying compression. By contrast, formulating context management as a budget-aware *sequential decision problem* enables adaptive modulation of commit-block aggregation intensity. This is most pronounced in the extreme 32-objective setting at the 4k limit, where the method achieves a 1.7× cumulative F1 improvement over the best baseline (2.06 vs. 1.21).

Compression Efficiency Under Budget Constraints Figure 3 indicates that under a fixed 8k budget, our gains reflect budget-aware management rather than indiscriminately increasing compression frequency. Under light workloads (2/8 objectives), our method improves F1 over MEM1 by 8.3%/18.7% (0.84→0.91, 2.35→2.79) while reducing compression calls by 41.7%/35.8% (1.92→1.12, 2.15→1.38). At 16/32 objectives, compression calls rise by 43%/109% (1.62→2.32, 1.28→2.68), with F1 improving by 67%/143% (2.40→4.01, 1.21→2.94). MEM1 fails because of context saturation and signal loss, causing premature responses. Meanwhile, our method balances token usage and performance; token statistics are in Appendix A.3.

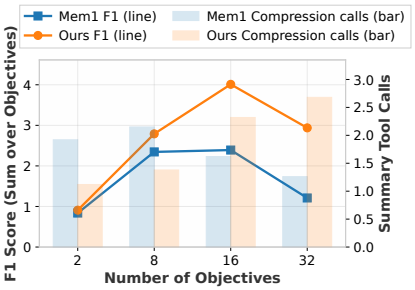

Figure 3: Cumulative F1 and average compression calls under a fixed 8k context budget.

4.5 Ablation Study

Impact of Budget-Aware State To analyze the framework's performance contributions, we evaluate four configurations by decoupling budget metadata from the compression policy (Table 2). We define *Base* (Search-R1) as the vanilla model without context management. *Search-R1 (w/ B)* adds budget metadata to the agent state, while *Ours (w/o B)* uses the learned compression policy without explicit budget signals. *Ours (Full)* integrates both for budget-conditioned context optimization.

Variant	Comp.	Budget
Search-R1 (Base)	–	–
Search-R1 (w/ B)	–	✓
Ours (w/o B)	✓	–
Ours (Full)	✓	✓

Table 2: Ablation configurations.

Figure 4 shows that removing budget information from our compression mechanism causes significant performance degradation across context budgets and multi-objective settings.

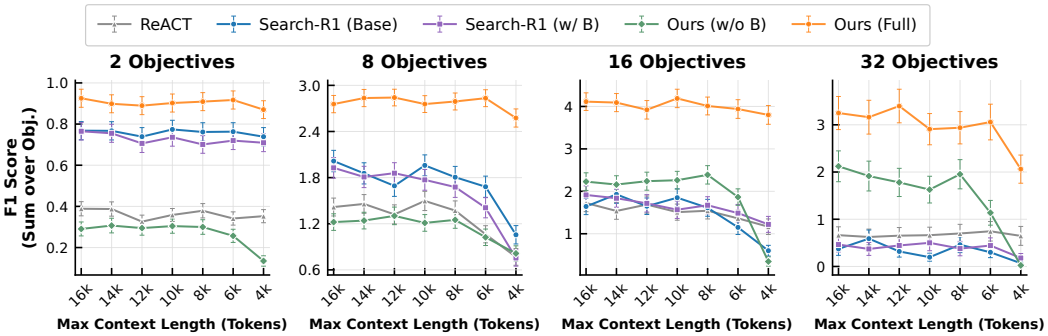

Figure 4: Ablation of key components across context budgets (16k–4k tokens), measured by summed F1 across objectives. Removing budget metadata (B) degrades performance (*Ours w/o B*), while adding B alone to a baseline without context management does not improve performance (*Search-R1 w/ B*). Our full model (*Ours Full*), which conditions compression on B, achieves the best performance across all objectives and budgets.

While *Search-R1 (w/ B)* yields only marginal gains, our full framework consistently achieves the highest cumulative F1 scores. By explicitly integrating budget signals, our approach enables superior dynamic trade-offs between preserving raw context fidelity and invoking aggressive summarization as the budget diminishes.

To further analyze our deferred-loading mechanism, we compare it with the pre-loading strategy under the same budget-constrained multi-objective setting (Table 3). The deferred version accounts for the upcoming observation size in the budget-aware state before loading. In contrast, pre-loading incorporates the observation without such awareness. Deferred loading consistently outperforms pre-loading, demonstrating the benefit of budget-aware deferred loading.

Variant	2-Obj	8-Obj	16-Obj	32-Obj
w/o Def.	0.833	2.706	3.972	2.818
w/ Def.	**0.909**	**2.790**	**4.011**	**2.938**

Table 3: Cumulative F1 of deferred vs. non-deferred loading across different objectives at 8k

Ablation of Progressive Context Budget Curriculum To isolate the effect of the progressive context budget curriculum, we ablate on Qwen2.5-7B-Instruct with three training schedules: a static 8k context budget, a random context budget sampled from $\{4k, 8k\}$, and a progressive context budget schedule of 8k $\rightarrow$ 4k, all evaluated under the same maximum context budget of 8k (see Appendix A.8 for more training dynamics).

Figure 5 shows the progressive curriculum is critical for budget-aware context management. Randomized training performs worst, indicating unstructured budget variation hinders learning. Although static 8k training performs well in the 2-objective setting (0.967), it scales poorly, reaching 1.391 and 1.319 on 16 and 32 objectives. In contrast, the progressive curriculum achieves the best overall performance (2.662 average), with strong gains on complex tasks, scoring 4.011 and 2.938 on 16 and 32 objectives. These results show progressively increasing budget pressure is essential for generalizing context-management behaviors to long-horizon settings.

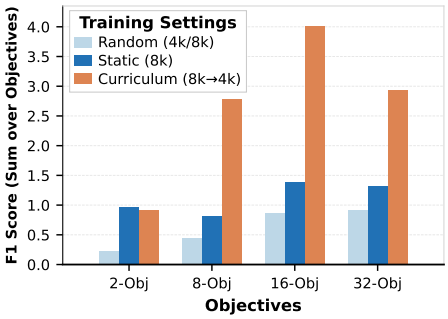

Figure 5: Ablation of progressive budget curricula under a common 8k evaluation budget summed F1 over objectives

4.6 Case Study: Analysis of Model Behavior

We analyze behavior under `[NONE]`, `[Selective]`, and `[ALL]` (see App. 6). With sufficient budget (e.g., $\sim$45.7%), the agent adopts `NONE`, preserving `commit_blocks` and deferring

compression. Under moderate pressure (e.g., ∼30.6%), it switches to `Selective`, merging earlier blocks (e.g., `commit_ids: "c1,c2,c3"`) to condense history while retaining newer, task-relevant details (e.g., `c4` reorganized as `c2`, `c3`). When the budget is tight (e.g., ∼28.7%), it escalates to `ALL`, invoking `commit_ids: "ALL"` to compress history into a `merged_summary`, increasing compression to meet the budget before appending new observations.

5 Conclusion

We study context management for long-horizon LLM agents under strict context constraints and show that budget-agnostic compression leads to information loss or overflow. We propose BACM, which formulates compression as a budget-conditioned sequential decision problem and optimizes it via reinforcement learning (BACM-RL). Experiments demonstrate consistent improvements in robustness across context budgets, especially under tight constraints. Overall, our results highlight the importance of budget-aware context management for reliable long-horizon reasoning.

References

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. Fireact: Toward language agent fine-tuning, 2023. URL https://arxiv.org/abs/2310.05915.

Guoxin Chen, Zile Qiao, Xuanzhong Chen, Donglei Yu, Haotian Xu, Wayne Xin Zhao, Ruihua Song, Wenbiao Yin, Huifeng Yin, Liwen Zhang, Kuan Li, Minpeng Liao, Yong Jiang, Pengjun Xie, Fei Huang, and Jingren Zhou. Iterresearch: Rethinking long-horizon agents with interaction scaling, 2026. URL https://arxiv.org/abs/2511.07327.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Sahel Sharifymoghaddam, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Nandan Thakur, Crystina Zhang, Luyu Gao, Wenhu Chen, and Jimmy Lin. Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent, 2025. URL https://arxiv.org/abs/2508.06600.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 3829–3846, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.232. URL https://aclanthology.org/2023.emnlp-main.232/.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 2414–2426, 2020.

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.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, Zhenrong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchunshu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. Memory in the age of ai agents, 2026. URL https://arxiv.org/abs/2512.13564.

Qishuo Hua, Lyumanshan Ye, Dayuan Fu, Yang Xiao, Xiaojie Cai, Yunze Wu, Jifan Lin, Junfei Wang, and Pengfei Liu. Context engineering 2.0: The context of context engineering, 2025. URL https://arxiv.org/abs/2510.26493.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.825. URL https://aclanthology.org/2023.emnlp-main.825/.

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 Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Peter Griffin Jin, Yifan Zeng, et al. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 1601–1611, 2017.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory os of ai agent, 2025. URL https://arxiv.org/abs/2506.06326.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, et al. Natural questions: a benchmark for question answering research. In *Transactions of the Association for Computational Linguistics*, volume 7, pp. 453–466, 2019.

Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. Steering llm thinking with budget guidance, 2025a. URL https://arxiv.org/abs/2506.13752.

Yucheng Li et al. Compressing context to enhance inference efficiency of large language models. In *EMNLP*, 2023. URL https://aclanthology.org/2023.emnlp-main.391/.

Zhiyu Li, Shichao Song, Hanyu Wang, Simin Niu, Ding Chen, Jiawei Yang, Chenyang Xi, Huayi Lai, Jihao Zhao, Yezhaohui Wang, Junpeng Ren, Zehao Lin, Jiahao Huo, Tianyi Chen, Kai Chen, Kehang Li, Zhiqiang Yin, Qingchen Yu, Bo Tang, Hongkang Yang, Zhi-Qin John Xu, and Feiyu Xiong. Memos: An operating system for memory-augmented generation (mag) in large language models, 2025b. URL https://arxiv.org/abs/2505.22101.

Nelson F. Liu, Kevin Lin, et al. Lost in the middle: How language models use long contexts. *TACL*, 2024. URL https://aclanthology.org/2024.tacl-1.9/.

Zeyu Liu et al. Budget-aware tool-use: Scaling agentic reasoning with finite tool-call budgets. *arXiv preprint arXiv:2511.17006*, 2025. URL https://arxiv.org/abs/2511.17006.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, Rongcheng Tu, Xiao Luo, Wei Ju, Zhiping Xiao, Yifan Wang, Meng Xiao, Chenwu Liu, Jingyang Yuan, Shichang Zhang, Yiqiao Jin, Fan Zhang, Xian Wu, Hanqing Zhao, Dacheng Tao, Philip S. Yu, and Ming Zhang. Large language model agent: A survey on methodology, applications and challenges, 2025. URL https://arxiv.org/abs/2503.21460.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, et al. When not to trust language models: Investigating effectiveness of parametric and non-parametric knowledge. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pp. 14734–14753, 2023.

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 19327–19352. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/3d77c6dcc7f143aa2154e7f4d5e22d68-Paper-Conference.pdf.

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

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, A Smith Noah, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *arXiv preprint arXiv:2210.03350*, 2022.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Charlie Snell et al. Scaling llm test-time compute optimally can be more effective than scaling model parameters. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org.

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

Abhishek Tale et al. Token-budgeted reasoning: Optimizing inference scaling for long-horizon tasks. *arXiv preprint arXiv:2412.18547*, 2024. URL https://arxiv.org/abs/2412.18547.

Zecheng Tang et al. L-citeeval: A suite for evaluating fidelity of long-context models. *ACL*, 2025. URL https://aclanthology.org/2025.acl-long.263/.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Hao Wen, Xinrui Wu, Yi Sun, Feifei Zhang, Liye Chen, Jie Wang, Yunxin Liu, Yunhao Liu, Ya-Qin Zhang, and Yuanchun Li. Budgetthinker: Empowering budget-aware llm reasoning with control tokens, 2025. URL https://arxiv.org/abs/2508.17196.

Xixi Wu, Kuan Li, Yida Zhao, Liwen Zhang, Litu Ou, Huifeng Yin, Zhongwang Zhang, Xinmiao Yu, Dingchu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, Minhao Cheng, Shuai Wang, Hong Cheng, and Jingren Zhou. Resum: Unlocking long-horizon search intelligence via context summarization, 2025. URL https://arxiv.org/abs/2509.13313.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

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

Rui Ye, Zhongwang Zhang, Kuan Li, et al. Agentfold: Long-horizon web agents with proactive context management. *arXiv preprint arXiv:2510.24699*, 2025. URL https://arxiv.org/abs/2510.24699.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent, 2025. URL https://arxiv.org/abs/2507.02259.

Qianhao Yuan, Jie Lou, Zichao Li, Jiawei Chen, Yaojie Lu, Hongyu Lin, Le Sun, Debing Zhang, and Xianpei Han. Memsearcher: Training llms to reason, search and manage memory via end-to-end reinforcement learning, 2025. URL https://arxiv.org/abs/2511.02805.

Yuxiang Zhang, Jiangming Shu, Ye Ma, Xueyuan Lin, Shangxi Wu, and Jitao Sang. Memory as action: Autonomous context curation for long-horizon agentic tasks, 2026. URL https://arxiv.org/abs/2510.12635.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory, 2023. URL https://arxiv.org/abs/2305.10250.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents, 2025. URL https://arxiv.org/abs/2506.15841.

A Appendix

A.1 Limitations and future work

Despite the strong empirical performance achieved by formulating context management as a budget-aware sequential decision problem, several limitations remain that warrant further investigation. (1) The reliance on trajectory-level reinforcement learning introduces challenges associated with sparse and delayed reward signals, which may limit the effectiveness of process-level supervision for context management decisions; a promising direction is to incorporate more structured credit assignment mechanisms, such as intermediate supervision, hierarchical reward design, or explicit modeling of long-term information utility (Feng et al., 2025). (2) While commit-block aggregation enables flexible and adaptive control over compression under varying budget conditions, it operates at a coarse segment level; exploring finer-grained importance modeling, such as token- or span-level saliency estimation, may further improve the preservation of critical information in tasks requiring high factual precision. (3) Although the evaluation demonstrates clear improvements on multi-objective QA and long-horizon browsing benchmarks designed to stress-test context management, extending the framework to more realistic and diverse environments, including open-ended tool use, multimodal reasoning, and human-agent interaction, would provide a more comprehensive assessment of generalization and practical applicability.

Overall, budget-aware context management provides a promising direction for enabling robust long-horizon reasoning under strict resource constraints, and future advances along these directions may further enhance its effectiveness in real-world agent systems.

A.2 Case study of budget-aware context management

We provide additional details for the case study presented in the main text (Figure 6). The example illustrates how the agent dynamically adjusts context management strategies under varying budget conditions, selecting among `[NONE]`, `[Selective]`, and `[ALL]`.

Q1 Who is Jack the Ripper in Gotham by Gaslight? Q2 Who killed Barry Allen's mom in the TV show? Q3 Who won the first ancient Olympic 200-yard race? Q4 Who was selected for the 2018 football hall of fame? Q5 What type of speed does a speedometer measure? Q6 What is the name of the lymphatic vessels located in the small intestine? Q7 The inner sanctuary of the Parthenon where the statue of Athena is located is known as the? Q8 What is the emblematic rhythm of the Dominican Republic? Q9 Who voiced Mewtwo in Pokémon the first movie? Q10 Who wrote the theme song to Law and Order? Q11 Most of New Zealand's land supports which economic activity? Q12 Who owns the biggest house in Beverly Hills? Q13 What was the original purpose of the Kremlin? Q14 When is the opening ceremonies of the Olympics 2018? Q15 Who sang theme song for Dukes of Hazard? Q16 Points on a sphere or angles in a circle are measured in units called?

Case Example (NONE)

[Pre-Action Context State]
- c1 (997 tokens): evidence for Q1-Q3 — Jacob Packer; Thawne killed Nora Allen; stadion as the first recorded Olympic event.

- c2 (997 tokens): same evidence as c0001 for Q1-Q3.

[Budget state]
- Prompt: 4019 tokens
- Est. tool response: 997 tokens
- Remaining:
4224 tokens(45.7%)

[Tool call]
- commit_ids: "NONE"
- merged_commit: ""

[Post-Action Context Change]
- c1 (997 tokens): retained
- c2 (997 tokens): retained
- c3 (997 tokens): retained

Case Example (Merge: c1, c2, c3)

[Pre-Action Context State]
- c1 (222 tokens): compact summary over Q1-Q16, including Jacob Packer (Q1), Eobard Thawne (Q2), the ancient stadion race (Q3), Hall of Fame inductees (Q4), speedometer (Q5), lacteals (Q6), and the cella (Q7).

- c2 (997 tokens): detailed evidence for Q1-Q3, stating that Jacob Packer is the Ripper, Thawne caused Nora Allen's death, and the stadion was the first recorded Olympic event.

- c3 (1001 tokens): overlapping evidence for Q2-Q4, repeating the Thawne and stadion facts and adding 2018 Hall of Fame names such as Ray Lewis and Randy Moss.

- c4 (966 tokens): later evidence for Q4-Q6, including additional Hall of Fame names, speedometer definition, and lymphatic-vessel descriptions.

[Budget state]
- Current prompt length: 5463 tokens
- Estimated tool response: 953 tokens
- Remaining tokens: 2824 tokens (30.6%)

[Tool call]
- commit_ids: "c1,c2,c3"
- merged_commit: compact summary over Q1-Q16

[Post-Action Context Change]
- c1 (222 tokens): merged summary retaining the key answers for Q1-Q16.

- c2 (966 tokens): retained detailed evidence for Q4-Q6, including Hall of Fame names, speedometer definition, and lymphatic-vessel descriptions.

- c3 (953 tokens): retained new evidence for Q6-Q7, including lacteals in the small intestine and the sanctuary of Athena.

Case Example (ALL)

[Pre-Action Context State]
- block c1 (997 tokens, tool_obs)
  - detailed evidence for Q1-Q3 from multiple retrieved docs.
  - related to Q1: Jacob Packer is identified as Jack the Ripper.
  - related to Q2: Eobard Thawne is responsible for Nora Allen's death.
  - related to Q3: the stadion is described as the first recorded ancient Olympic event.

- block c2 (997 tokens, tool_obs)
  - largely overlapping evidence for Q1-Q3.
  - repeats the same conclusions on Jacob Packer, Nora Allen, and the stadion race.

- block c3 (997 tokens, tool_obs)
  - again preserves overlapping evidence for Q1-Q3.
  - mostly repeated content from the same retrieval sources.

[Budget state]
- Current prompt length: 5588 tokens
- Estimated tool response length: 997 tokens
- Remaining tokens for next turn: 2655 tokens (28.7% of usable context)

[Tool call]
- commit_ids: "ALL"
- merged_commit: compact summary over Q1-Q16

[Post-Action Context Change]

- block c1 (222 tokens, fold)
  - merged summary spanning Q1-Q16.
  - related to Q1-Q3: Jacob Packer; Eobard Thawne; first ancient Olympic 200-yard race winner unknown.
  - related to Q4-Q7: Hall of Fame inductees; speedometer; lacteals; the cella.
  - related to Q8-Q16: merengue; Vic Mignogna; Mike Post and Pete Carpenter; sheep farming; Tom Ford; defense; Feb. 9, 2018; Waylon Jennings; degrees.

- block c2 (997 tokens, tool_obs)
  - latest detailed evidence retained for Q1-Q3.
  - preserves the same answer-relevant details in full form.

Figure 6: Case study of budget-aware context management. The agent adaptively selects `[NONE]`, `[Selective]`, or `[ALL]` based on remaining budget, transitioning from full retention to partial and full aggregation as budget tightens, while preserving task-relevant information.

A.3 Efficiency–Performance Trade-offs with Improved Context Utilization

Method	Ctx	2 Obj			8 Obj			16 Obj			32 Obj		
		D	P	S	D	P	S	D	P	S	D	P	S
Summary	16k	664.469	1517.570	0.661	1451.613	3871.188	2.127	1658.180	4938.746	2.447	2894.227	6306.773	0.656
Search-R1	16k	306.648	792.320	0.768	1481.602	3769.055	2.016	2280.371	5729.789	1.643	3669.387	6810.332	0.372
Mem1	16k	**304.439**	**632.912**	0.838	**652.492**	**914.818**	2.345	**876.083**	**1096.275**	2.391	**1046.315**	**1361.961**	1.210
Ours	16k	445.133	2032.777	**0.925**	1227.461	2986.828	**2.756**	1919.848	3537.504	**4.114**	7600.406	8795.160	**3.249**
Search-R1	8k	304.152	793.641	0.762	1332.250	3445.043	1.805	1920.008	4952.941	1.612	2556.023	5065.641	0.461
Summary	8k	409.352	1234.168	0.677	758.180	2780.047	2.371	1027.000	4160.898	2.628	1508.281	4497.906	0.491
Mem1	8k	**304.439**	**632.912**	0.838	**652.492**	**914.818**	2.345	**876.083**	**1096.275**	2.391	**1046.315**	**1361.961**	1.210
Ours	8k	482.633	2069.656	**0.909**	1030.082	2767.555	**2.790**	1848.777	3442.949	**4.011**	5224.945	6436.852	**2.938**
Search-R1	4k	298.672	763.586	0.739	910.277	2849.121	1.056	1145.297	3522.727	0.600	1486.727	3317.695	0.069
Summary	4k	508.785	2710.801	1.479	858.246	3087.055	0.508	305.969	1192.008	0.704	701.258	3448.832	1.109
Mem1	4k	**304.439**	**632.912**	0.838	**652.492**	**914.818**	2.345	**876.083**	**1096.275**	2.391	**1046.315**	**1361.961**	1.210
Ours	4k	403.254	1985.531	**0.870**	942.934	2659.348	**2.576**	1584.344	3129.082	**3.801**	2862.035	4001.288	**2.061**

Table 4: mean dependent cost (D), mean peak tokens (P), and summed F1 (S) across context budgets and objective counts.

We compare methods under multiple context budgets (4k, 8k, 16k) and varying objective counts, reporting mean dependent cost (D), peak context tokens (P), and summed F1 (S). As shown in Table 4, the proposed method adapts token usage to both task complexity and available context, and consistently achieves the highest performance.

For simple tasks (2 objectives), token usage remains stable across context budgets (e.g., Peak: 2033 at 16k vs. 1986 at 4k) while yielding higher scores, indicating that additional context does not introduce unnecessary token growth. As task complexity increases (8–16 objectives), the proposed method maintains superior performance with competitive or lower Peak tokens compared to methods without effective context control (e.g., 16 objectives: 3538 vs. 4939 for Summary at 16k; 3443 vs. 4161 at 8k), which reflects more efficient utilization of context. For more complex settings (32 objectives), Peak tokens increase with larger context budgets (4339 $\rightarrow$ 6437 $\rightarrow$ 8795 from 4k to 16k), leading to consistent performance improvements. In contrast, Mem1 maintains minimal token usage (1362) but exhibits limited performance gains, primarily because it does not preserve sufficient task-relevant information within the available context.

Overall, these results indicate that effective context management requires adaptive allocation of tokens rather than aggressive compression or delayed summarization.

A.4 Answer rate analysis across context lengths

Figure 7 shows answer rates across context lengths (4K–16K tokens) and task complexities (2/8/16/32 objectives) on the 7B model.

In simple scenarios (2–8 objectives), all methods exhibit answer rates above 90%. At higher complexities (16–32 objectives), baseline methods (ReACT, Search-R1, Summary) show lower answer rates, particularly at shorter context lengths. Mem1 maintains higher answer rates across settings (61.7% at 32 objectives).

In scenarios with 2–16 objectives, our method exhibits answer rates comparable to Mem1. At 32 objectives, our method shows a lower answer rate than Mem1, while corresponding to higher task performance due to retaining more information; compared to other baselines, our method maintains higher answer rates. Task performance results are reported in the main experiments.

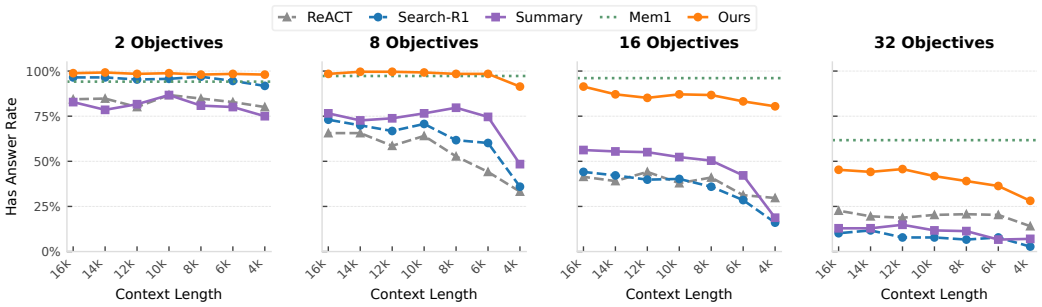

Figure 7: Answer rate comparison across context lengths and objectives.

	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5
Step Range	1–60	61–120	121–180	181–240	241–300
Max Model Length (tokens)	8,192	7,168	6,144	5,120	4,096

Table 5: Curriculum Schedule for `max_model_len`

A.5 Implementation Details and Hyperparameters

A.5.1 Training infrastructure and Core Hyperparameters

All experiments are conducted using the `verl` framework. Our method and the Search-R1 baseline share a unified RL configuration for consistency. We employ Group Relative Policy Optimization (GRPO) with a total of 16 nodes, each equipped with 8 GPUs. The training is performed for a total of 300 steps with a global batch size of 128. For the actor model, we use a cosine learning rate schedule starting at with 100 warmup steps. Detailed hyperparameters are summarized in Table 6.

A.5.2 Curriculum Reinforcement Learning Implementation

To enhance the model's context-management capabilities under progressively constrained resources, our method adopts a curriculum-based reinforcement learning strategy. Training is partitioned into $K = 5$ stages, each consisting of exactly 60 steps (yielding a total of 300 steps).

The maximum model length (`max_model_len`) follows the curriculum schedule detailed in Table A.5.2. It starts at 8,192 tokens in Stage 1 and is linearly reduced to 4,096 tokens in Stage 5 (a uniform decrease of 1,024 tokens per stage).

Unlike prior approaches that utilize a fixed maximum context length, our curriculum gradually tightens the context-window budget $B_{\max}^{(k)}$ across K training stages, where $k \in \{1, \ldots, K\}$ denotes the current stage. This mechanism forces the agent to internalize increasingly sophisticated management behaviors as it transitions from ample capacity to extreme scarcity.

To ensure fair comparison, the total training steps for our method as well as all baselines are uniformly fixed at 300 steps.

A.5.3 Search-R1 Implementation

For the Search-R1 baseline, we strictly follow the official implementation and training data construction as described in Jin et al. (2025). The model is trained to utilize external search tools within the reasoning chain. The environment interface and multi-turn interaction format (Hermes) are kept identical to our proposed method to ensure a fair comparison of context management capabilities. All training is performed for exactly 300 steps under

Hyperparameter	Value
Optimizer	AdamW
Actor Learning Rate	1×10^{-6}
LR Schedule	Cosine
Warmup Steps	100
Rollout Group Size (G)	5
KL Coefficient (β)	0.001
GRPO Mini-batch Size	128
Max Model Length	8,192 (initial; curriculum reduces to 4,096 for our method)

Table 6: Global Hyperparameters for RL Training

the shared RL configuration (with fixed `max_model_len`=8,192), by which point training has largely converged.

A.5.4 MEM1 Implementation

The MEM1 baselines utilize the official implementation provided by Zhou et al. (2025). For the Qwen2.5-7B-Instruct model, we use the provided weights and prompt templates. We found that the 30B-A3B model responded more effectively to modified special tokens, thus, we replaced the standard `<think>` tags with `<internal_state>` tags in the prompt to ensure the model correctly triggered its reasoning and memory behaviors.

A.6 Budget-Aware Context Management Prompt

The prompt variables are computed from the current agent state before the pending tool response is appended. Here, `current_ctx_len` is the token length of the current prompt buffer, and `tool_response_len` is the token length of the deferred tool response text. We conservatively set `usable_limit` to `max_model_len` minus a 1,000-token safety margin. The remaining budget is then `usable_limit` minus the projected post-load length, i.e., `current_ctx_len` + `tool_response_len`, and `remaining_pct` is its normalized percentage. This budget prompt is used only when a later tool response is deferred, so folding decisions are made before the new observation is loaded.

Budget State Prompt

```
[Remaining context capacity prior to receiving tool call results]
Please decide whether to summarize or fold the upcoming tool response to stay within limits.
- Current prompt length: {current_ctx_len} tokens
- Estimated tool response length: {tool_response_len} tokens
- Remaining tokens for next turn: {remaining_budget} tokens ({remaining_pct:.1f}% of usable
context)
- Usable context limit (max length minus 1,000 safety margin): {usable_limit} tokens
Decide whether to fold previous commits to save context budget.
OPTIONS:
- "NONE": Keep all commits (default)
- "ALL": Fold everything (only if context is low)
- "c0001,c0002": Fold specific old commits
RULE: Don't fold unless necessary. Preserve user requirements and errors.
OUTPUT format:
<tool_call>
{"name": "summarize", "arguments": {"fold_commit_ids": "NONE", "merged_commit": ""}}
</tool_call>
Examples:
- Safe: fold_commit_ids="NONE", merged_commit=""
- Low context: fold_commit_ids="ALL", merged_commit="[key points from session]"
- Selective: fold_commit_ids="c0001,c0002", merged_commit="[merged content]"
```

Ours (w/o B) uses the learned compression policy without explicit budget signals. Compared to the Budget State Prompt, it removes all budget-related information.

Compression Prompt (w/o Budget)

```
"Please choose the most appropriate folding strategy"
"The options include: Partial (e.g., c0001,c0002) / ALL / NONE."
"'c0001,c0002' means folding those commits and replacing them with merged_commit
information."
"'ALL' means folding all commits and replacing them with merged_commit."
"'NONE': Keep all commits (default)"
OUTPUT format:
<tool_call>
{"name": "summarize", "arguments": {"fold_commit_ids": "NONE", "merged_commit": ""}}
</tool_call>
```

A.7 LLM-as-a-Judge Evaluation Details

We follow the BrowseComp-Plus evaluation protocol and implement the judge using the Qwen3-32B model. The judge receives the question, the model-generated response, and the ground-truth answer, and determines whether the predicted answer is semantically equivalent to the reference answer.

Following the benchmark design, two independent judges with identical prompts are used to assess answer correctness, and their agreement differs by less than 1%, indicating high evaluation reliability. The BrowseComp-Plus authors further report that LLM-judged results closely match human annotations.

The judge prompt used in our evaluation is shown in Figure A.7.

Judge Prompt

Judge whether the following response to a question is correct based on the provided correct answer.
Question: {question}
Response: {response}
Correct Answer: {correct_answer}
Instructions:
1. Extract the final answer from the response.
2. Determine whether the extracted answer is semantically equivalent to the correct answer. Minor wording differences are acceptable. The extracted answer may be more detailed than the correct answer as long as the additional information is correct.
3. Output the following fields:

`extracted_final_answer`: the final answer extracted from the response
`correct_answer`: repeat the provided correct answer
`reasoning`: explanation of whether the extracted answer matches the correct answer
`correct`: yes or no
`confidence`: value between 0 and 100

A.8 Training Dynamics Analysis

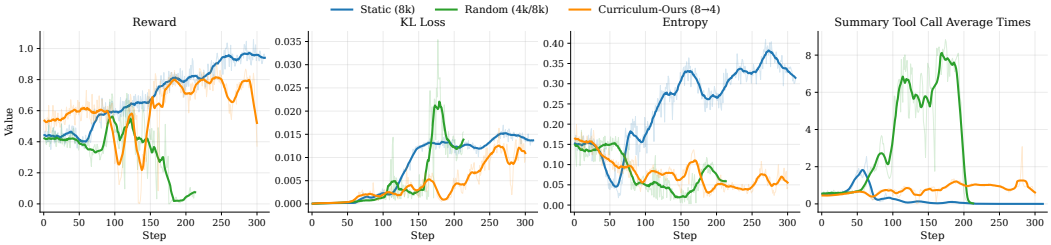

Figure 8: Training dynamics of Qwen2.5-7B-Instruct across Static (8k), Random (4k/8k), and Curriculum-Ours (8→4): Reward, KL Loss, Entropy, and Summary Tool Call Average Times.

Training setup. Figure 8 shows the training dynamics of Qwen2.5-7B-Instruct under three context-budget schedules. The Static (8k) baseline keeps the maximum context budget fixed at 8k throughout training, representing the original setting. Our Curriculum (8→4) progressively reduces the maximum context budget from 8k to 4k as training proceeds, with the budget tightened every 60 steps. To control for the effect of merely exposing the model to a 4k budget, we further introduce a Random (4k/8k) schedule, in which the maximum context budget at each step is randomly sampled from $\{4\text{k}, 8\text{k}\}$.

The Static (8k) baseline achieves stable reward improvement throughout training, while the Random (4k/8k) strategy fails to converge, exhibiting reward collapse after step 180 accompanied by KL divergence spikes. In contrast, our progressive curriculum maintains consistent reward gains even during steps 200–300 under tightening constraints. The summary-tool call frequency reveals distinct compression behaviors: Static training shows near-zero compression after step 100, as abundant context removes compression pressure, whereas our curriculum progressively increases compression frequency as the budget decreases from 8k to 4k. This confirms that gradually tightening context budgets effectively induces adaptive context-management behaviors, enabling the model to learn compression strategies rather than avoiding them when resources are plentiful.