C^3: COMPETITIVE CONTEXT COMPRESSION FOR LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language-model agents must compress growing interaction histories while retaining information needed for future decisions. We observe sublinear growth in total response length when an LLM handles several items jointly, suggesting an implicit soft output budget. We introduce ***Competitive Context Compression (C^3)***, which induces competition among jointly compressed histories by combining downstream task rewards with a validity bonus under a shared output budget. Over-budget outputs are truncated and receive no validity bonus or partial validity credit; truncation may also reduce task rewards. Individual memories have no fixed quotas. We train over compaction events at multiple depths, using a curriculum to introduce histories shaped by earlier memory decisions. Although training jointly processes multiple histories, deployment compresses one history at a time. C^3 achieves 51.40% success on SWE-bench Verified and 47.33% on BrowseComp-Plus, improving over single-history GRPO by 3.20 and 6.66 percentage points, respectively. On BrowseComp-Plus, it also reduces cumulative downstream-agent input by 7.9%, from 102.439K to 94.388K tokens per task. Ablations show that joint generation and shared-budget feedback contribute together, supporting competition during training as a useful signal for compression within individual agent trajectories.

1 INTRODUCTION

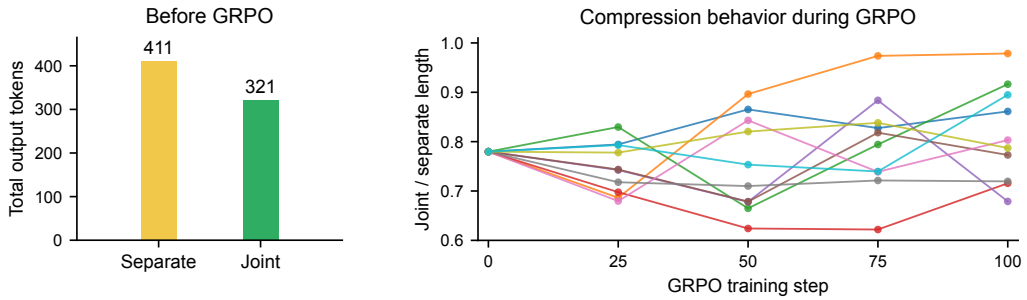
Language-model agents accumulate interaction histories as they reason and use tools (Yao et al., 2023; Schick et al., 2023; Yang et al., 2024). Growing contexts increase inference cost, can exceed context windows, and do not guarantee reliable information use (Packer et al., 2023; Liu et al., 2024; Hsieh et al., 2024; Bai et al., 2024). Compression must therefore preserve useful information within limited capacity (Ye et al., 2026; Zhong et al., 2024). Whereas prompt compression reduces static inputs (Jiang et al., 2023; 2024; Xu et al., 2024), agent memories must support future decisions and subsequent compactions (Park et al., 2023; Wu et al., 2025; Zhou et al., 2026). Their quality depends on the execution they enable after replacing the original history (Kang et al., 2026; Zhang et al., 2026).

This creates a capacity-allocation problem (Wu et al., 2026; Wei et al., 2026): histories of similar length may require different memories. A completed search may leave a few essential facts, whereas unresolved debugging may require hypotheses, file locations, and dependencies. Uniform targets do not capture this variation, and independent length penalties (Arora & Zanette, 2025) do not couple compression decisions across histories. We ask whether sharing limited capacity can improve the information selected for retention.

Our starting observation is that total response length grows sublinearly when several items share a prompt, suggesting an implicit *soft output budget*. This is consistent with the per-question self-compression reported by Deng et al. (2026). This behavior is not reliably preserved by task-reward optimization: in a three-question arithmetic diagnostic, the joint-to-separate total-length ratio is 0.78 before training but ranges from 0.68 to 0.98 across ten seeds after 100 GRPO steps (Figure 1). This motivates explicit length regularization to sustain capacity competition.

We introduce Competitive Context Compression (C^3), which jointly compresses independent histories under a shared output budget. GRPO (Shao et al., 2024) optimizes average downstream task success

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.



(a) Before GRPO, Joint answering uses fewer tokens than separate answering (ratio: 0.78)

(b) During GRPO, each curve denotes one training seed; across ten independent runs, the joint-to-separate total-length ratio drifts substantially rather than remaining fixed.

Figure 1: **Implicit capacity sharing is not reliably preserved by task-reward optimization.** We use a controlled three-question arithmetic task to isolate the effect. The ratio is defined as the total length of the joint response divided by the summed lengths of the three separate responses.

plus a validity bonus for complete, correctly formatted outputs within budget. Over-budget responses are truncated; correctly formatted items are retained, but the entire response receives zero validity reward with no partial validity credit. Truncation may also reduce task rewards. Individual memories have no fixed quotas, so allocating more tokens to one leaves fewer for others. The trained compressor processes individual histories at deployment.

Repeated compaction also changes compressor inputs: earlier memories shape subsequent interactions and later compactions. We use a curriculum (Bengio et al., 2009) that progressively covers the first one, two, and three compaction events. Later-event histories are collected with the updated compressor while earlier events remain represented, exposing training to the consequences of preceding memory decisions.

Experiments on SWE-bench Verified and BrowseComp-Plus (Jimenez et al., 2024; Chowdhury et al., 2024; Chen et al., 2025) show improved task success over single-history GRPO and lower cumulative agent input in long-document search. Ablations separate the effects of joint generation and budget pressure. They also show that retaining fewer tokens per compaction need not lower cumulative input, because subsequent execution determines the overall context cost. Our contributions are:

- Shared-budget compression training that couples memory allocation across histories through joint generation and length regularization.
- An RL objective over multiple compaction depths and a curriculum incorporating histories shaped by earlier memory decisions.
- Evidence of improved single-history deployment across two domains, with analyses of joint generation, training group size, and budget calibration.

2 Preliminaries

2.1 Agents with Recursive Context Compression

We consider a fixed language-model agent that interacts with an environment through actions and observations. Its working context H contains the textual history available for subsequent decisions. A separate compressor π_θ maps this history to a textual memory M, which should preserve information needed for continued execution. The downstream agent remains fixed when the compressor is trained.

Compression is triggered when the working context reaches an input-length threshold T_{ctx}. We wait for the current interaction round, including its resulting observation, to finish before replacing the accumulated history with the generated memory. The agent then resumes from this memory and appends new interactions. If k indexes compression events, this process can be written as

$$M_k \sim \pi_\theta(\cdot \mid H_k), \qquad H_{k+1} = [M_k, \Delta H_{k+1}], \tag{1}$$

where ΔH_{k+1} contains the interactions accumulated before the next compression, and brackets denote concatenation. We refer to this repeated replacement as *recursive compression*. Later compression inputs therefore contain earlier compressed memories as well as new observations.

2.2 Group Relative Policy Optimization

For an input X, GRPO samples G candidate outputs Y_j from the old policy and evaluates their rewards R_j (Shao et al., 2024). It assigns each candidate the group-relative advantage

$$\widehat{A}_j = \frac{R_j - \overline{R}}{\sigma_R + \delta}, \tag{2}$$

where $\overline{R}$ and σ_R are the group mean and standard deviation, and $\delta > 0$ ensures numerical stability.

For token $y_{j,t}$ with generated prefix $Y_{j,<t}$, define

$$\rho_{j,t}(\theta) = \frac{\pi_\theta(y_{j,t} \mid X, Y_{j,<t})}{\pi_{\theta_{\text{old}}}(y_{j,t} \mid X, Y_{j,<t})}. \tag{3}$$

GRPO minimizes the clipped loss (Shao et al., 2024; Schulman et al., 2017)

$$\begin{aligned}\mathcal{L}(\theta) = -\mathbb{E}\Big[&\min\{\rho_{j,t}(\theta)\widehat{A}_j, \text{clip}(\rho_{j,t}(\theta), 1-\epsilon, 1+\epsilon)\widehat{A}_j\} \\ &- \beta D_{\text{KL}}(\pi_\theta(\cdot \mid X, Y_{j,<t}) \| \pi_{\text{ref}}(\cdot \mid X, Y_{j,<t}))\Big].\end{aligned} \tag{4}$$

Here ϵ controls clipping, β weights KL regularization toward π_{ref}, and the expectation is over inputs and candidates sampled from the old policy.

3 Method

Figure 2 summarizes joint compression, downstream reward evaluation, and the curriculum over successive compaction depths.

3.1 Competition under an Output Budget

Our motivating observation is that total response length grows sublinearly as the number of items processed jointly in a single prompt increases. C^3 uses this implicit soft budget to induce competition among memories: each prompt X contains N histories from independent agent trajectories, and the compressor produces a joint output Y containing their memories $M_1, \ldots, M_N$. Histories are randomly ordered within each joint prompt.

The implicit budget can expand during RLVR as outputs become longer. We counteract this growth by regularizing total memory length against a fixed reference B_{mem}. A joint output earns a validity bonus only when every memory is complete and correctly formatted, and

$$\sum_{i=1}^{N} |M_i| \le B_{\text{mem}}, \tag{5}$$

where $|M_i|$ is the memory length. No memory has an individual quota, so retaining extra detail for one history leaves less room for the others if the output is to earn the bonus.

For each prompt, GRPO samples G alternative joint outputs. To evaluate each candidate, we return its memories to their corresponding trajectories and resume the fixed downstream agent to termination, with further compression disabled. Let z_i indicate whether trajectory i succeeds, and let r_{valid} be one when the joint output meets the validity conditions above and zero otherwise. Each candidate receives one reward:

$$R(X, Y) = \frac{2}{3N} \sum_{i=1}^{N} z_i + \frac{1}{3} r_{\text{valid}}. \tag{6}$$

Over-budget responses are truncated and receive zero validity reward. Complete, correctly formatted memories are passed to the execution agent, including those ending exactly at the budget. If truncation

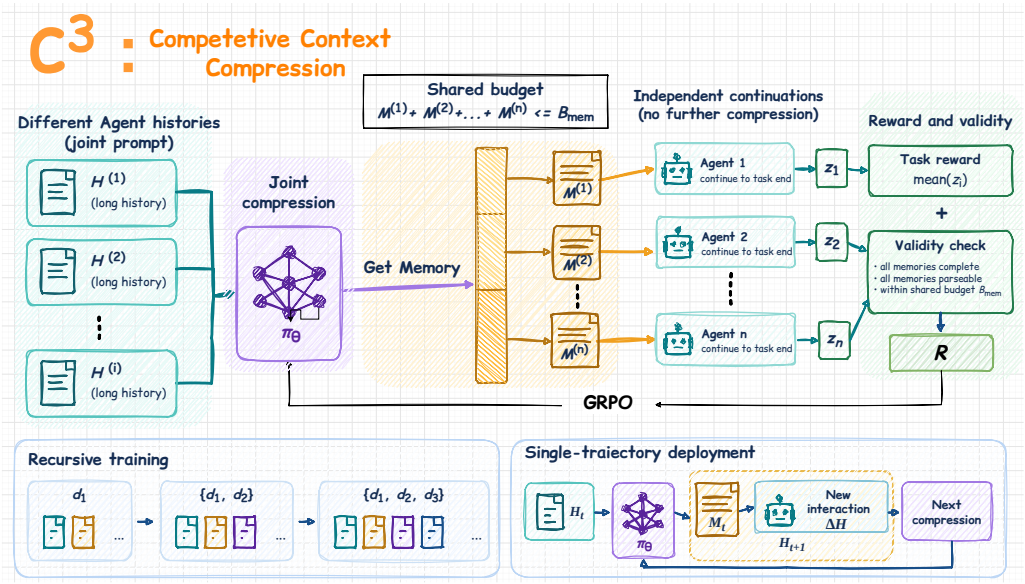


Figure 2: Training pipeline of C^3. Independent histories are jointly compressed under a shared budget. The compressor is updated with GRPO using downstream task feedback, while stage-wise construction of recursive states progressively expands the compression depths covered during training.

prevents extracting a complete memory for trajectory i, its continuation is skipped and $z_i = 0$. The length condition thus acts as a regularizer, with no token-wise penalty within the target. GRPO compares rewards across the G candidates to update the compressor (Section 2.2); Appendix A gives the full update objective.

3.2 Curriculum Learning across Multiple Compactions

Repeated compaction changes the inputs the compressor must handle. Let d_k^θ denote the history distribution at the k-th compaction event among trajectories that reach that position during recursive deployment. The first event precedes any memory replacement, while later histories depend on the preceding $k-1$ memories and the interactions they enable (Equation 1). Training only on first-event histories omits these later inputs. We take a compaction event as the training unit and progressively include later events.

At stage K, training covers the first K compaction positions. Let $\widehat{d}_k$ denote the distribution of collected histories at position k, generated using compressor checkpoints during data collection. We balance the available positions:

$$\bar{d}_K = \frac{1}{K}\sum_{k=1}^{K} \widehat{d}_k. \tag{7}$$

Each joint prompt X contains N histories $H_{1:N}$ drawn independently from this mixture, from independent trajectories. We optimize

$$J_K(\theta) = \mathbb{E}_{\substack{H_{1:N}\sim(\bar{d}_K)^N \\ Y\sim\pi_\theta(\cdot|X)}} \left[\frac{2}{3N}\sum_{i=1}^{N} z_i + \frac{1}{3} r_{\text{valid}} \right], \tag{8}$$

using the reward in Equation 6; the expectation also covers the downstream continuations used to obtain the task outcomes. The output budget determines validity-bonus eligibility and may also affect task outcomes through truncation. Balancing positions gives later compactions explicit training coverage; it does not reproduce their deployment frequencies.

The curriculum begins with the first compaction, then includes the first or second, and finally the first, second, or third. We periodically roll out the updated compressor to collect later-event histories while

retaining earlier examples. This delays deeper-state collection until the compressor has learned from earlier events, rather than constructing all depths with the initial policy.

For an example at position k, its prefix contains $k-1$ compactions, and the candidate memory is the k-th. Subsequent compaction is disabled during reward evaluation to avoid attributing errors from later memories to this candidate. The curriculum therefore trains on recursively constructed inputs, while continuation without further compression approximates the candidate's utility during recursive deployment. Deployment processes one history at a time and permits further compactions whenever its context reaches the input threshold.

4 EXPERIMENTS

We evaluate whether shared-budget training improves the quality of compressed agent histories, how joint generation and budget pressure contribute to this effect, and whether the learned behavior transfers to single-history deployment and different compression thresholds. Section 4.4 examines how budget calibration affects the learning signal. At deployment, C^3 recursively compresses one history at a time.

4.1 EXPERIMENTAL SETUP

Training and evaluation. The compressor is initialized from Qwen3.5-9B (Qwen Team, 2026), which has a context window of 262,144 tokens, and trained with GRPO. Unless otherwise specified, training jointly processes three independent histories, uses a shared length target of 1,400 tokens for the validity reward, and expands recursive-state training to depth three. Compression is triggered when the agent's accumulated context exceeds 4,096 tokens. The input threshold determines when compression occurs; the shared output budget governs truncation and eligibility for the validity reward during training. Evaluation processes one history per compression call, without a memory-output budget or output-length cap.

Budget calibration. We calibrate budgets using natural, untruncated outputs of the initial compressor on groups of N agent histories. We define the initial *budget-hit rate* as the fraction of these outputs exceeding a candidate budget and target approximately 30%. This yields budgets of 814, 1,400, and 2,230 tokens for $N = 1, 3$, and 5, respectively. The budgets remain fixed during training; the realized hit rate may change. Section 4.4 varies the budget and its initial hit rate at fixed $N = 3$.

For software engineering, we train on approximately 2,000 SWE-smith examples (Yang et al., 2025) with DeepSeek-V4 Flash (DeepSeek-AI, 2026) as the downstream agent and evaluate on all 500 SWE-bench Verified tasks using the official evaluation criteria (Jimenez et al., 2024; Chowdhury et al., 2024). For long-document search, we train on the LART BM25 subset with Tongyi-DeepResearch-30B-A3B (Tongyi DeepResearch Team, 2025) as the downstream agent and evaluate on BrowseComp-Plus (Chen et al., 2025), using DeepSeek V3.2 (DeepSeek-AI, 2025) as the judge under the dataset's evaluation protocol. The threshold-generalization study additionally uses 150 LART test tasks. Within each domain, methods use the corresponding tool interfaces and identical execution limits. Tasks that terminate because of context-window overflow remain in the evaluation set.

Metrics. We report task success together with four measures of agent execution: (i) ***Peak context*** is the maximum input length over downstream agent calls within a task, averaged across tasks. (ii) ***Cumulative agent input*** sums input tokens across all downstream calls in a task and then averages across tasks. (iii) ***Retained fraction*** is the compressed memory length divided by the history length at the compression event. (iv) ***Turns*** counts downstream agent generations per task. Peak context measures the context capacity used by the agent, whereas cumulative input also depends on how long execution continues.

Comparisons. We compare C^3 against the following baselines: (i) ***Full-context ReAct*** retains the complete interaction history (Yao et al., 2023). (ii) ***Base compressor*** uses Qwen3.5-9B without our reinforcement learning updates. (iii) $\boldsymbol{N = 1 + \mathrm{GRPO}}$ trains on one history per input without the output-length criterion. (iv) $\boldsymbol{N = 1 + C^3}$ adds an individual memory budget during single-history training; both single-history variants use the same recursive compression procedure as the default C^3

Table 1: Main results on SWE-bench Verified and BrowseComp-Plus. Peak and Input are mean peak context and cumulative downstream agent input, respectively, in thousands of tokens; compressor calls are excluded. Turns counts agent generations per task. Retained is the fraction of history retained after compression. Shading identifies the default C^3 configuration.

Agent backbone	*SWE-bench Verified*				
DeepSeek-V4 Flash	**Success↑**	**Peak↓**	**Input↓**	**Turns**	**Retained**
Base compressor	26.40%	4.21	175.68	68.40	7.01%
$N = 1$ + GRPO	48.20%	4.16	195.65	63.38	7.76%
$N = 1 + C^3$	47.00%	4.14	205.72	64.49	6.36%
CoMem	43.40%	4.16	202.65	61.37	6.07%
CoMem K4	36.40%	4.13	198.32	67.57	7.73%
$\boldsymbol{N = 3 + C^3}$	**51.40%**	**4.18**	**199.25**	**64.07**	**4.18%**
Full-context ReAct	61.80%	27.20	949.11	54.68	N/A
Agent backbone	*BrowseComp-Plus*				
Tongyi-DeepResearch	**Success↑**	**Peak↓**	**Input↓**	**Turns**	**Retained**
Base compressor	32.00%	4.03	61.037	91.74	6.59%
$N = 1$ + GRPO	40.67%	4.09	102.439	153.12	3.65%
$N = 1 + C^3$	40.00%	4.07	101.392	153.85	3.53%
LongLLMLingua	38.00%	4.09	93.884	147.34	5.72%
VISTA	43.33%	4.10	90.247	141.69	N/A
$\boldsymbol{N = 3 + C^3}$	**47.33%**	**4.09**	**94.388**	**131.78**	**3.94%**
Full-context ReAct	28.67%*	59.70	130.929	34.45	N/A

* The Tongyi model has a maximum context window of 73,728 tokens; 64% of tasks failed because accumulated context exceeded this limit.

configuration. (v) ***CoMem*** is an external context-management baseline that uses a decoupled memory model to compress interaction histories (Zhang et al., 2026). (vi) ***CoMem K4*** uses the $k = 4$ step-off configuration of CoMem (Zhang et al., 2026). We evaluate CoMem and CoMem K4 for software engineering. (vii) ***LongLLMLingua*** applies query-aware prompt compression to long-context inputs (Jiang et al., 2024). (viii) ***VISTA*** is a training-free context-management method that exposes the agent's context state through typed, addressable blocks and a runtime dashboard, allowing the agent to manage its own working context (Xu et al., 2026). We evaluate LongLLMLingua and VISTA for long-document search. Our default model, $\boldsymbol{N = 3 + C^3}$, jointly compresses three histories under a shared memory budget. In the ablations, we additionally train joint GRPO without the output-length criterion and vary the number of jointly compressed histories under an output budget.

4.2 Task Success and Context Use

Across both domains, C^3 achieves the highest task success among the evaluated compression methods (Table 1). Since the compressed configurations operate under similar peak-context limits, the substantial differences in success rate indicate that compression quality depends not only on how much context is retained, but on whether the retained state preserves the information needed for subsequent execution.

On SWE-bench Verified, $\boldsymbol{N = 3 + C^3}$ reaches 51.40% success, improving over single-history GRPO by 3.20 percentage points and outperforming all other compressed baselines. Full-context ReAct achieves a higher success rate of 61.80%, but requires a mean peak context of 27.20K tokens and 949.11K cumulative input tokens, compared with 4.18K and 199.25K for C^3. Thus, C^3 achieves the highest downstream task success among the evaluated compression methods.

On BrowseComp-Plus, the role of compression is more fundamental. Full-context execution frequently becomes infeasible because accumulated histories exceed the agent's 73,728-token context window, causing 64% of tasks to terminate from context overflow. Compression is therefore essential for sustaining execution rather than merely reducing inference cost. Under this constraint, C^3

Config.	Succ.↑	Peak↓	Input↓	Ret.
$N = 1$ + GRPO	48.20%	4.16	195.65	7.76%
$N = 1 + C^3$	47.00%	4.14	205.72	6.36%
$N = 3$ + GRPO	48.20%	4.24	303.95	7.02%
$\boldsymbol{N = 3 + C^3}$	**51.40%**	**4.18**	**199.25**	**4.18%**
$N = 5 + C^3$	50.60%	4.12	298.11	4.14%

Table 2: Extended SWE-bench Verified ablations. Peak and Input are in thousands of tokens; Ret. is the retained fraction. All compressors are evaluated on one history at a time.

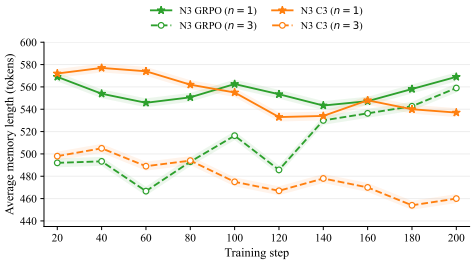


Figure 3: Memory-length dynamics for models trained with $N = 3$ histories. At each checkpoint, $n = 1$ and $n = 3$ denote single-history and joint evaluation, respectively.

achieves 47.33% success, exceeding the strongest alternative compression baseline, VISTA, by 4.00 percentage points and single-history GRPO by 6.66 percentage points. Thus, C^3 achieves the highest task success among the evaluated compression strategies.

The efficiency metrics further show that retained fraction alone does not determine compression quality. On BrowseComp-Plus, our method retains slightly more history than single-history GRPO (3.94% vs. 3.65%), yet requires fewer agent turns (131.78 vs. 153.12), less cumulative input (94.388K vs. 102.439K), and achieves substantially higher success. Moreover, $\boldsymbol{N = 3 + C^3}$ retains a larger fraction of history than $\boldsymbol{N = 1 + C^3}$ (3.94% vs. 3.53%), while still achieving a higher success rate, suggesting that overly aggressive compression may itself degrade compression quality.

4.3 Joint Generation and Shared-Budget Competition

Table 2 isolates the effects of joint generation and shared-budget training. Joint generation alone does not improve success: moving from $N = 1$ GRPO to $N = 3$ GRPO keeps success at 48.20% while increasing cumulative input from 195.65K to 303.95K tokens. Likewise, imposing a budget on a single history reduces the retained fraction from 7.76% to 6.36% but does not improve task success. Relative to joint GRPO without a budget, their combination improves both outcomes: $N = 3 + C^3$ reaches 51.40% success while reducing cumulative input to 199.25K. These ablations support the combination of joint generation and shared-budget training; neither component alone reproduces its task-success benefit.

Figure 3 shows how this difference emerges during training. Joint evaluation initially produces shorter memories per history, but under ordinary $N = 3$ GRPO the $n = 3$ length progressively approaches the $n = 1$ length. With C^3, the separation remains substantially larger, indicating that the shared budget preserves the compression pressure induced by joint generation. Mixed-budget candidate groups occur across all three training stages (Figure 5). They can provide a relative validity signal when within-budget candidates also satisfy the format and completeness requirements.

4.4 Budget Calibration

We next examine how the shared budget controls compression strength. With $N = 3$ and all other settings fixed, we compare $B_{\text{mem}} \in \{2500, 1800, 1400, 840\}$, corresponding to initial budget-hit rates of 1%, 10%, 30%, and 50%, respectively. As shown in Figure 4, moderate budget pressure performs best: $B_{\text{mem}} = 1400$ reduces mean memory length from 314.5 to 225.3 tokens relative to $B_{\text{mem}} = 1800$, while increasing success from 38.00% to 51.40%. A tighter budget is not necessarily better: at $B_{\text{mem}} = 840$, mean length rises to 308.9 tokens and success drops to 28%. The shared budget therefore acts as a calibration parameter rather than a monotonic compression knob.

The training dynamics provide a complementary explanation. The validity bonus changes relative advantages only when candidates differ in full validity; truncation may separately change task outcomes. Figure 5 tracks mixed-budget groups as a diagnostic, whose prevalence varies across training stages and budget settings. Together, these results motivate choosing a budget that creates sustained capacity pressure without making the constraint overwhelmingly difficult to satisfy.

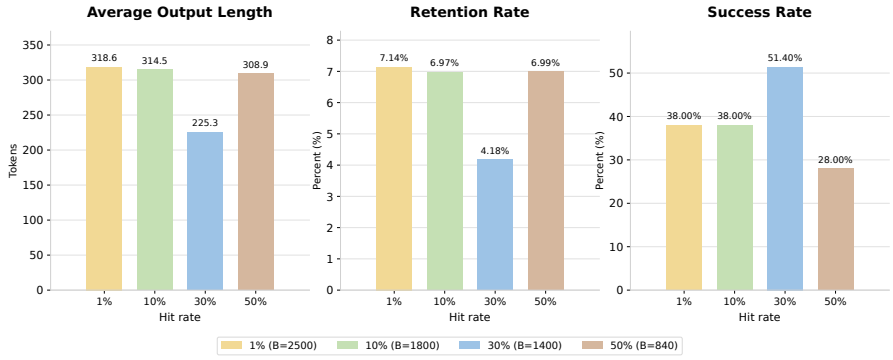


Figure 4: Effect of shared-budget strength on single-history deployment. Initial budget-hit rates of 1%, 10%, 30%, and 50% correspond to $B_{\text{mem}} = 2500, 1800, 1400$, and 840, respectively. Panels report mean memory length, retained fraction, and task success on SWE-bench Verified.

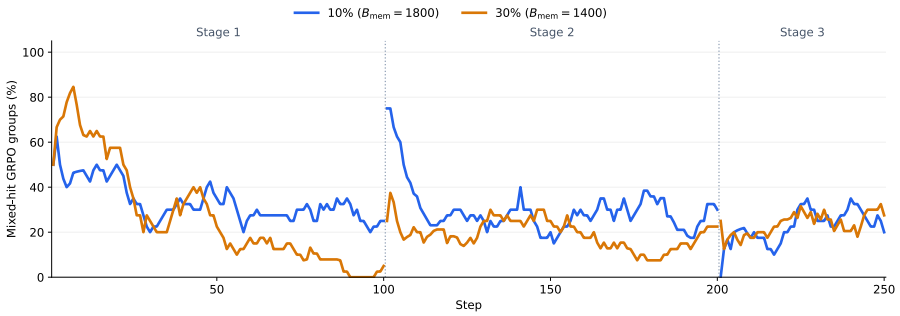


Figure 5: Fraction of GRPO groups containing both over-budget and within-budget candidates at $N = 3$. Initial hit rates of 10% and 30% correspond to budgets of 1,800 and 1,400 tokens. Ratios use known-status groups and a trailing ten-update window within each stage; dashed lines mark stage boundaries.

4.5 Transfer across Training Group Sizes and Compression Thresholds

Training group size. We train with one, three, or five histories per input and evaluate all models on individual histories. For each group size, the budget targets an initial hit rate of approximately 30%: 814, 1,400, and 2,230 tokens for $N = 1, 3$, and 5, respectively (Section 4.1). Table 2 shows that the benefit of joint training survives single-history deployment: success increases from 47.00%

Table 3: Generalization across compression thresholds on 150 LART BM25 test tasks. Success and Retained are percentages; Comp. is the average number of compression events, and Input is cumulative downstream-agent input in thousands of tokens.

Threshold	Configuration	Success↑	Comp.	Retained	Input↓
T_{ctx}	Base	58.00%	3.75	18.07%	45.310
	$N = 1$ + GRPO	64.00%	4.59	8.80%	61.780
	$\boldsymbol{N = 3 + C^3}$	**72.00%**	**4.01**	**7.09%**	**54.942**
$1.25T_{\text{ctx}}$	Base	52.00%	2.56	13.31%	50.347
	$N = 1$ + GRPO	68.00%	3.43	4.81%	70.635
	$\boldsymbol{N = 3 + C^3}$	**73.33%**	**2.77**	**6.00%**	**58.316**
$1.5T_{\text{ctx}}$	Base	52.00%	2.05	10.01%	61.672
	$N = 1$ + GRPO	66.00%	2.26	4.01%	68.953
	$\boldsymbol{N = 3 + C^3}$	**68.00%**	**2.47**	**4.65%**	**73.898**

with $N = 1$ to 51.40% with $N = 3$, while $N = 5$ achieves a similar 50.60%. Increasing the group size beyond three provides little additional task benefit and raises cumulative input from 199.25K to 298.11K, indicating that stronger competition does not monotonically improve deployment efficiency.

Compression threshold. We further evaluate the long-document compressor on 150 LART BM25 test tasks at its training threshold and two larger deployment thresholds, without retraining (Table 3). C^3 maintains the highest success rate among the compared compressors at all three thresholds, achieving 72.00%, 73.33%, and 68.00%. Moderately delaying compression reduces the number of compression events, but larger thresholds eventually increase cumulative input and erode the efficiency advantage. Thus, the learned compression behavior transfers across context lengths, while the deployment threshold remains a separate efficiency–performance tradeoff.

5 Related Work

Prompt compression selects useful tokens from static inputs (Li et al., 2023; Jiang et al., 2023; 2024; Pan et al., 2024), while ReSum and MEM1 maintain compact states during interaction (Wu et al., 2025; Zhou et al., 2026). Active context managers fold or reorganize histories (Sun et al., 2026; Ye et al., 2026; Liu et al., 2026; Li et al., 2026), with ContextBudget and PACE adapting retention to capacity and anticipated relevance (Wu et al., 2026; Wei et al., 2026). ACON refines compression guidelines from execution failures; CoMem trains a separate memory model through next-action agreement (Kang et al., 2026; Zhang et al., 2026). Memory-R1 learns external-memory operations and utilization through outcome-driven RL (Yan et al., 2026). Reward shaping also controls reasoning length (Yeo et al., 2025). ConPress uses multi-question prompting to obtain concise reasoning for distillation (Deng et al., 2026). C^3 instead couples independent histories under a shared length limit, supervises memories with terminal task outcomes, and trains across recursive compaction depths for single-history deployment.

6 Discussion and Limitations

Shared-budget training improves single-history deployment, but aggregate outcomes do not establish how capacity is allocated across histories. Our evaluation covers two domains and does not separately isolate the compaction curriculum's contribution. Reward evaluation disables subsequent compaction, leaving a mismatch with recursive deployment that staged training does not fully resolve. In addition, cumulative input measures downstream-agent context use and excludes compressor calls and training costs. These results therefore do not establish end-to-end compute savings. Shorter memories can alter subsequent actions and the number of interaction steps, so local compression and cumulative context use need not improve together. The deployment threshold also remains a separate design choice: our transfer experiments show that delaying compression can improve success while increasing cumulative input. Future work should directly analyze memory allocation, isolate curriculum effects, and evaluate total execution cost across broader tasks.

7 Conclusion

We introduced Competitive Context Compression (C^3), a framework for training agent memory compressors through competition among independent histories under a shared output budget. By combining terminal task feedback, a budget-dependent validity reward, and staged training over recursive compaction depths, C^3 learns compact memories that support subsequent agent execution. The resulting compressor operates on a single history at deployment without an output budget or length cap. Across SWE-bench Verified and BrowseComp-Plus, C^3 achieves the highest task success among the evaluated compression methods, improving over single-history GRPO by 3.20 and 6.66 percentage points, respectively. Budget and group-size ablations further show that compression length alone is insufficient to predict task success or cumulative context use. Together, these findings support shared-budget competition as an effective training approach for task-oriented context compression in long-horizon agents. They further demonstrate that coupling independent histories during training can benefit later single-history execution. This provides a practical direction for learning agent memories whose quality is assessed by their contribution to task completion.

AI USE STATEMENT

We used generative AI tools to assist with language polishing and figure preparation. The authors take full responsibility for the final manuscript, including its text, figures, and scientific claims.

REFERENCES

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2032. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/579b5b84311e122584dedab1d3b7613f-Abstract-Conference.html.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.172. URL https://aclanthology.org/2024.acl-long.172/.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 41–48, 2009. doi: 10.1145/1553374.1553380. URL https://ronan.collobert.com/pub/matos/2009_curriculum_icml.pdf.

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. *arXiv preprint arXiv:2508.06600*, 2025. URL https://arxiv.org/abs/2508.06600.

Neil Chowdhury, James Aung, Chan Jun Shern, Oliver Jaffe, Dane Sherburn, Giulio Starace, Evan Mays, Rachel Dias, Marwan Aljubeh, Mia Glaese, Carlos E. Jimenez, John Yang, Leyton Ho, Tejal Patwardhan, Kevin Liu, and Aleksander Madry. Introducing SWE-bench Verified. OpenAI, 2024. URL https://openai.com/index/introducing-swe-bench-verified/.

DeepSeek-AI. DeepSeek-V3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025. URL https://arxiv.org/abs/2512.02556.

DeepSeek-AI. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026. URL https://arxiv.org/abs/2606.19348.

Jie Deng, Shining Liang, Jun Li, Hongzhi Li, and Yutao Xie. ConPress: Learning efficient reasoning from multi-question contextual pressure. *arXiv preprint arXiv:2602.01472*, 2026. URL https://arxiv.org/abs/2602.01472.

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? In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kIoBbc76Sy.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 13358–13376. Association for Computational Linguistics, 2023. 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 *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1658–1677. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.91. URL https://aclanthology.org/2024.acl-long.91/.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=VTF8yNQM66.

Minki Kang, Wei-Ning Chen, Dongge Han, Huseyin Inan, Lukas Wutschitz, Yanzhi Chen, Robert A Sim, and Saravanakumar Rajmohan. ACON: Optimizing Context Compression for Long-horizon LLM Agents. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=5EmOOLtH5P.

Mo Li, L. H. Xu, Qitai Tan, Long Ma, Hongyong Song, Ting Cao, and Yunxin Liu. Sculptor: Empowering LLMs with Cognitive Agency via Active Context Management. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=HPeiH7da0Z.

Yucheng Li, Bo Dong, Frank Guerin, and Chenghua Lin. Compressing context to enhance inference efficiency of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6342–6353. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.391. URL https://aclanthology.org/2023.emnlp-main.391/.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

Shukai Liu, Bo Jiang, Jian Yang, Yizhi LI, Jinyang Guo, Xianglong Liu, and Bryan Dai. Context as a tool: Context management for long-horizon SWE-agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 20604–20617. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-acl.1032. URL https://aclanthology.org/2026.findings-acl.1032/.

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

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

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*. Association for Computing Machinery, 2023. doi: 10.1145/3586183.3606763. URL https://doi.org/10.1145/3586183.3606763.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551. Curran Associates, Inc., 2023. doi: 10.52202/075280-2997. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

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

Weiwei Sun, Miao Lu, Zhan Ling, Kang Liu, Xuesong Yao, Yiming Yang, and Jiecao Chen. Scaling Long-Horizon Agent via Context Folding. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=lNRgWoGfYg.

Tongyi DeepResearch Team. Tongyi DeepResearch technical report. *arXiv preprint arXiv:2510.24701*, 2025. URL https://arxiv.org/abs/2510.24701.

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 *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27184–27199. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1252. URL https://aclanthology.org/2026.acl-long.1252/.

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. *arXiv preprint arXiv:2509.13313*, 2025. URL https://arxiv.org/abs/2509.13313.

Yong Wu, YanZhao Zheng, TianZe Xu, ZhenTao Zhang, YuanQiang Yu, JiHuai Zhu, Chao Ma, BinBin Lin, BaoHua Dong, HangCheng Zhu, RuoHui Huang, and Gang Yu. ContextBudget: Budget-Aware Context Management for Long-Horizon Search Agents. *arXiv preprint arXiv:2604.01664*, 2026. URL https://arxiv.org/abs/2604.01664.

Binyan Xu, Haitao Li, and Kehuan Zhang. LLM agents are latent context managers: Eliciting self-managed context via state proprioception. *arXiv preprint arXiv:2606.30005*, 2026. URL https://arxiv.org/abs/2606.30005.

Fangyuan Xu, Weijia Shi, and Eunsol Choi. RECOMP: Improving retrieval-augmented LMs with context compression and selective augmentation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=mlJLVigNHp.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.583. URL https://aclanthology.org/2026.acl-long.583/.

John Yang, Carlos Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering. In *Advances in Neural Information Processing Systems*, volume 37, pp. 50528–50652. Curran Associates, Inc., 2024. doi: 10.52202/079017-1601. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/5a7c947568c1b1328ccc5230172e1e7c-Abstract-Conference.html.

John Yang, Kilian Lieret, Carlos E. Jimenez, Alexander Wettig, Kabir Khandpur, Yanzhe Zhang, Binyuan Hui, Ofir Press, Ludwig Schmidt, and Diyi Yang. SWE-smith: Scaling data for software engineering agents. *arXiv preprint arXiv:2504.21798*, 2025. URL https://arxiv.org/abs/2504.21798.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik 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, 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 *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=IuZoTgsUws.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in LLMs. *arXiv preprint arXiv:2502.03373*, 2025. URL https://arxiv.org/abs/2502.03373.

Yuwei Zhang, Chengyu Dong, Shuowei Jin, Changlong Yu, Hejie Cui, Hongye Jin, Xinyang Zhang, Hamed Bonab, Colin Lockard, Jianshu Chen, Zhenyu Shi, Jingbo Shang, Xian Li, and Bing Yin. CoMem: Context management with a decoupled long-context model. *arXiv preprint arXiv:2605.30842*, 2026. URL https://arxiv.org/abs/2605.30842.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024. doi: 10.1609/aaai.v38i17.29946. URL https://ojs.aaai.org/index.php/AAAI/article/view/29946.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Bryan Kian Hsiang Low, and Paul Liang. MEM1: Learning to Synergize Memory and Reasoning for Efficient Long-Horizon Agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XY8AaxDSLb.