WHICH MEMORIES ARE WORTH CONSOLIDATING? SELECTIVE MEMORY CONSOLIDATION FOR LANGUAGE MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

External memories enable large language model agents to reuse prior experience, improve exploration, and solve complex interactive tasks more effectively. However, relying on external memory makes agent performance dependent on retrieving appropriate memories, while noisy or conflicting guidance may lead to incorrect actions. Memory consolidation offers an alternative by selectively transferring experience that contributes to further policy improvement into model parameters, turning accumulated memories into reusable capabilities and reducing reliance on retrieval. The key challenge is to identify which memories remain useful for learning as training progresses, since a memory's usefulness in guiding a rollout does not necessarily translate into further improvement of the memory-free policy. We propose **Co**nsolidation of **M**emory into **P**arameters (COMP), a selective memory consolidation framework that identifies memories whose induced updates can persistently improve the memory-free policy. COMP uses memories to guide trajectory generation and reinforcement learning to consolidate useful behaviors, selecting memories based on whether their induced updates are compatible with memory-free policy improvement. Experiments on ALFWorld, WebShop, and ScienceWorld show that COMP consistently improves memory-free performance over GRPO and memory-augmented baselines. Specifically, COMP achieves relative improvements of up to 20.3% on ALFWorld and 12.7% on WebShop over GRPO across model scales, while also demonstrating improved generalization to out-of-distribution tasks.

1 INTRODUCTION

Large language model agents (Wu et al., 2023; Xi et al., 2025; Nie et al.; Wang et al., 2024; Yue, 2025) increasingly rely on external memories to reuse prior experience, procedural knowledge, and task-specific guidance (Packer et al., 2023; Wu et al., 2025; Hatalis et al., 2023). By retrieving relevant memories and incorporating them into the context, agents can avoid repeated failures, improve long-horizon decision-making, and discover successful trajectories more efficiently (Shinn et al., 2023; Zhao et al., 2024; Xu et al., 2026; Chang et al., 2026). Memories can provide reusable strategies, highlight common pitfalls, and guide exploration without requiring the model to relearn the same behaviors from scratch. Consequently, external memory has become an important mechanism for improving the adaptability and performance of LLM agents.

However, relying solely on external memory introduces several limitations that may become more pronounced as experience accumulates. As the memory bank expands, the agent must identify relevant information from an increasingly large collection of records. Its performance therefore depends heavily on whether appropriate memories are retrieved and presented at the right time, while noisy or conflicting memories may distract the policy and lead to incorrect actions (Liu et al., 2024; Shi et al., 2023; Wang et al., 2025; Snell et al., 2022). Maintaining the memory bank and injecting retrieved records into the agent's context also increase context length, inference cost, and system complexity (Yue et al., 2025b; Keles et al., 2023). These limitations motivate memory consolidation: training the policy on memory-guided rollouts so that behaviors supported by retrieved memories become part of the model's weights and no longer require retrieval at inference time. This mirrors

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.

memory consolidation in humans, where episodic memories are gradually integrated into long-term knowledge (McClelland et al., 1995; O'Reilly et al., 2014).

The key challenge is to identify which memories are worth consolidating at each stage of policy learning. Retrieval-time usefulness alone is insufficient, because a memory may improve the current rollout without inducing a policy update that is beneficial to retain (Geirhos et al., 2020; Wang et al., 2026; Mindermann et al., 2022). For example, consider an ALFWorld (Shridhar et al., 2021) task in which the agent is instructed to `put some spraybottle on toilet`, with the target spray bottle located inside a closed cabinet. A memory such as `open a closed receptacle before retrieving an object from it` can provide a useful learning signal by encouraging a reusable interaction pattern. In contrast, `the target object is in cabinet 2` may help the current rollout by reducing exploration, yet consolidating the behavior induced by this cue can introduce a spurious preference for a particular location when layouts or object placements change. Importantly, consolidation value is also policy-dependent: once the policy has already acquired a behavior supported by a memory, using that memory to drive further consolidation may provide little additional improvement. Thus, what to consolidate depends not on whether a memory helps at inference time, but on whether its induced update still benefits the current policy, motivating CoMP, which uses training signals to keep such memories and filter out the rest.

We propose **Co**nsolidation of **M**emory into **P**arameters (CoMP), a selective memory consolidation framework that identifies memories whose induced updates can yield beneficial and persistent improvements in the memory-free policy. CoMP uses external memories to guide trajectory generation and consolidates useful behaviors through reinforcement learning, whose reward-driven optimization turns successful memory-guided experience into task-adapted policy capabilities. Rather than selecting memories based solely on their retrieval-time usefulness, CoMP estimates whether the update induced by each memory is compatible with improving the current memory-free policy. Updates predicted to provide beneficial policy improvement are consolidated, while memories that do not provide a useful training signal continue to serve only as training-time guidance. For memories selected for consolidation, CoMP removes the memory from the prompt during policy optimization, encouraging the policy to reproduce the memory-guided behavior from its parameters alone. We therefore evaluate the resulting policy without external memory so that performance directly reflects the capabilities retained through consolidation, rather than gains attributable to inference-time retrieval. This evaluation setting isolates the effect of parameterized learning and provides a clear measure of whether memory-guided experience has been successfully internalized. Our contributions could be summarized as follows:

- We formulate selective memory consolidation as the problem of determining which memory-induced updates provide beneficial learning signals for the current policy and are worth retaining in model parameters, rather than treating all retrieved memories as equally suitable for consolidation.
- We propose CoMP, a reinforcement-learning framework that uses retrieved memories to guide trajectory generation and selectively chooses whether to retain or remove them during policy optimization. For memories selected for consolidation, CoMP removes the memory from the prompt while optimizing the policy on the resulting behavior, forcing the policy to reproduce the memory-guided behavior from its parameters alone. This explicitly trains a memory-free policy and therefore removes the need for external memory retrieval at deployment.
- Across ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2023), and ScienceWorld (Wang et al., 2022), CoMP consistently improves memory-free performance over standard RL and indiscriminate memory consolidation, achieving relative gains of up to 20.3% on ALFWorld and 12.7% on WebShop over Group Relative Policy Optimization(GRPO) (Guo et al., 2025). Moreover, results on ScienceWorld suggest improved generalization to out-of-distribution tasks.

2 Preliminaries

2.1 Agentic Reinforcement Learning

We consider an LLM agent π_θ parameterized by θ that interacts with an environment to solve a task $x \sim \mathcal{D}$. At interaction step t with the interaction history h_t, the agent receives an observation o_t and generates an action a_t according to its policy $a_t \sim \pi_\theta(\cdot \mid h_t)$. An episode produces a trajectory $\tau = (o_1, a_1, \ldots, o_T, a_T)$ and receives a trajectory-level reward $r(\tau)$. The objective of

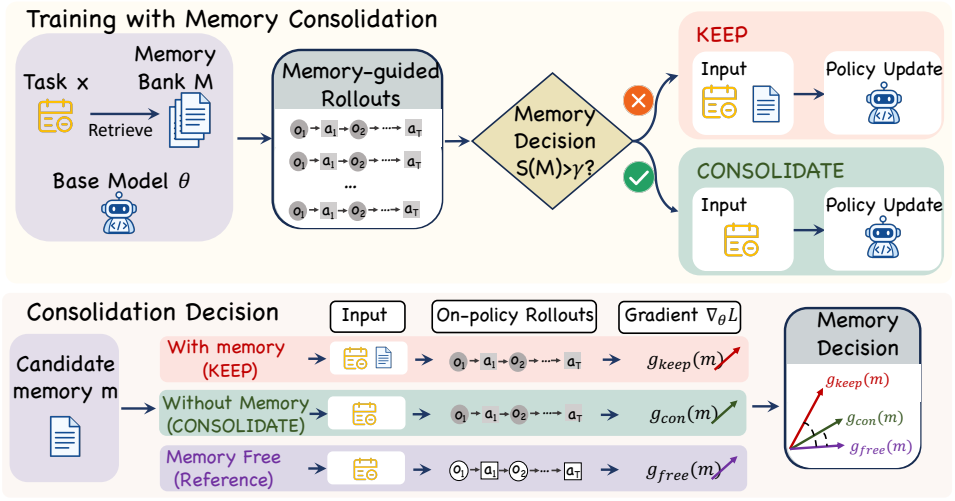


Figure 1: Overview of CoMP. The upper panel shows memory-guided training with selective consolidation. The lower panel shows how the consolidation decision is made from memory-conditioned and memory-free gradients.

agentic reinforcement learning is to maximize the expected return

$$J(\theta) = \mathbb{E}_{x\sim\mathcal{D},\,\tau\sim\pi_\theta(\cdot|x)}\left[r(\tau)\right]. \tag{1}$$

We instantiate policy optimization using Group Relative Policy Optimization (GRPO). For each task x, GRPO samples a group of G trajectories $\{\tau_i\}_{i=1}^{G}$ and computes the group-relative advantage $\widehat{A}_i = \frac{r(\tau_i)-\overline{r}}{s_r+\epsilon_A}$, where $\overline{r}$ and s_r are the mean and standard deviation of the rewards within the group. Given a policy context c, the token-level importance ratio is $\rho_{i,t}(\theta;c) = \frac{\pi_\theta(a_{i,t}|h_{i,t},c)}{\pi_{\theta_{\text{old}}}(a_{i,t}|h_{i,t},c)}$. The GRPO loss is

$$\mathcal{L}_{\text{GRPO}}(\theta;\mathcal{B},c) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\left(\rho_{i,t}(\theta;c)\widehat{A}_i,\,\text{clip}\left(\rho_{i,t}(\theta;c),1-\epsilon_c,1+\epsilon_c\right)\widehat{A}_i\right) + \beta\mathcal{L}_{\text{KL}}(\theta), \tag{2}$$

where $\mathcal{B}$ denotes the sampled trajectory group, ϵ_c is the clipping coefficient, and β controls the regularization toward a reference policy.

2.2 Memory-Augmented Training

Let $\mathcal{M} = \{m_i\}_{i=1}^{N}$ denotes an external memory bank, a set of relevant memories $\mathcal{M}_x$ are selected by the retrieval criterion for each task x. The retrieved memories are inserted into the policy context, resulting in a memory-augmented policy $\pi_\theta\left(a_t \mid h_t;\mathcal{M}_x\right)$. Conventional memory-augmented training uses the same retrieved memories during both trajectory generation and policy optimization:

$$\theta^{+} = \text{Update}\left(\theta;\mathcal{L}_{\text{GRPO}}\left(\theta;\mathcal{B}^{\mathcal{M}},\mathcal{M}_x\right)\right), \tag{3}$$

where $\mathcal{B}^{\mathcal{M}}$ is a batch of memory-guided trajectories. Such training directly improves the policy while the memories remain in its context. However, the resulting improvements remain tied to the retrieved memories. When the memories are removed at inference time, the policy largely loses the behaviors they supported. CoMP instead selectively consolidates memory-guided behaviors into a memory-free policy.

3 Method

CoMP uses external memories to guide trajectory generation and selectively determines whether their induced behaviors should be consolidated into the policy parameters. For each memory, we compare a memory-free update that removes it during optimization with a memory-conditioned update that retains it, using their relative benefit to the memory-free policy to guide consolidation.

3.1 CONSOLIDATION OBJECTIVE

Let $\mathcal{M}_x \subseteq \mathcal{M}$ denote the memory group retrieved for task x, and let $\mathcal{B}_x^{\mathcal{M}}$ denote the trajectories generated under the full retrieved memory context. Using the same trajectories, rewards, and advantages, we construct two counterfactual policy updates. The *memory-retained update* keeps the retrieved memory group in the policy context:

$$\theta^{+}_{\text{keep}}(\mathcal{M}_x) = \text{Update}\left(\theta; \mathcal{L}^{\mathcal{M}_x}_{\text{keep}}(\theta)\right), \tag{4}$$

where $\mathcal{L}^{\mathcal{M}_x}_{\text{keep}}(\theta) = \mathcal{L}_{\text{GRPO}}\left(\theta; \mathcal{B}_x^{\mathcal{M}}, \mathcal{M}_x\right)$.

The *consolidation update* removes the entire retrieved memory group from the policy context:

$$\theta^{+}_{\text{con}}(\mathcal{M}_x) = \text{Update}\left(\theta; \mathcal{L}^{\mathcal{M}_x}_{\text{con}}(\theta)\right), \tag{5}$$

where $\mathcal{L}^{\mathcal{M}_x}_{\text{con}}(\theta) = \mathcal{L}_{\text{GRPO}}\left(\theta; \mathcal{B}_x^{\mathcal{M}}, \varnothing\right)$.

Thus, both branches learn from the same behavior generated with the retrieved memory group, but differ in whether that group remains visible during policy optimization. Let

$$J_{\varnothing}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \tau\sim\pi_\theta(\cdot|x,\varnothing)}\left[r(\tau)\right] \tag{6}$$

denote the expected return without external memory. We define the ideal consolidation worthiness of the retrieved memory group as

$$W_\theta(\mathcal{M}_x) = J_{\varnothing}\left(\theta^{+}_{\text{con}}(\mathcal{M}_x)\right) - J_{\varnothing}\left(\theta^{+}_{\text{keep}}(\mathcal{M}_x)\right). \tag{7}$$

A positive $W_\theta(\mathcal{M}_x)$ indicates that removing the retrieved memory group during optimization is expected to produce a better memory-free policy than retaining it as external context. A negative value instead favors keeping the group during optimization. Importantly, Eq. 7 compares two update strategies: $W_\theta(\mathcal{M}_x) > 0$ does not imply improvement over the original policy, only that consolidation is preferable to retention.

Directly evaluating Eq. 7 is prohibitively expensive because it would require applying both counterfactual updates and evaluating both resulting policies without memory for every retrieved memory group. We therefore derive an efficient local surrogate.

3.2 MEMORY CONSOLIDATION DECISION

Define the parameter displacements induced by the two counterfactual updates as $\delta\theta_{\text{keep}}(\mathcal{M}_x)$ and $\delta\theta_{\text{con}}(\mathcal{M}_x)$. Let $\mathcal{L}_{\varnothing}(\theta)$ denote a differentiable memory-free policy loss whose descent direction locally improves $J_{\varnothing}(\theta)$, and define $g_{\text{free}} = \nabla_\theta \mathcal{L}_{\varnothing}(\theta)$. A first-order expansion of the two counterfactual policies gives

$$W_\theta(\mathcal{M}_x) \approx \langle g_{\text{free}}, \delta\theta_{\text{keep}}(\mathcal{M}_x) - \delta\theta_{\text{con}}(\mathcal{M}_x)\rangle. \tag{8}$$

A complete derivation and approximation-error bound are provided in Appendix A.

Algorithm 1 : Consolidation Decision

Require: Task x, retrieved memory group $\mathcal{M}_x$, policy π_θ, threshold γ

1: $\mathcal{B}_x^{\mathcal{M}} \sim \pi_\theta(\cdot \mid x, \mathcal{M}_x)$
2: Compute g^x_{keep}, g^x_{con}, and $\widetilde{g}^x_{\text{free}}$
3: $S_\theta(\mathcal{M}_x) \leftarrow \left\langle \widetilde{g}^x_{\text{free}}, g^x_{\text{con}} - g^x_{\text{keep}} \right\rangle$
4: $z(\mathcal{M}_x) \leftarrow \mathbb{I}[S_\theta(\mathcal{M}_x) > \gamma]$
5: **return** $z(\mathcal{M}_x)$

In practice, collecting fresh memory-free trajectories for every retrieved memory group remains expensive. We therefore estimate the memory-free direction using matched replay. Given the memory-guided trajectories $\mathcal{B}_x^{\mathcal{M}}$, we remove the retrieved memory group from the policy context while retaining the logged actions, observations, rewards, and group-relative advantages. This yields a memory-free replay loss $\widetilde{\mathcal{L}}^{\mathcal{M}_x}_{\text{free}}(\theta)$ and gradient $\widetilde{g}^{\mathcal{M}_x}_{\text{free}} = \nabla_\theta \widetilde{\mathcal{L}}^{\mathcal{M}_x}_{\text{free}}(\theta)$. We similarly approximate the two counterfactual parameter updates using their gradients, $g^{\mathcal{M}_x}_{\text{keep}}$ and $g^{\mathcal{M}_x}_{\text{con}}$. This yields the practical decision signal

$$S_\theta(\mathcal{M}_x) = \left\langle \widetilde{g}^{\mathcal{M}_x}_{\text{free}}, g^{\mathcal{M}_x}_{\text{con}} - g^{\mathcal{M}_x}_{\text{keep}} \right\rangle. \tag{9}$$

We convert this value into a binary consolidation decision. At decision-refresh step k, we define

$$z_k(\mathcal{M}_x) = \mathbb{I}\left[S_{\theta_k}(\mathcal{M}_x) > \gamma\right], \tag{10}$$

where γ is the consolidation threshold and is set to 0 unless otherwise specified. A retrieved memory group with $z_k(\mathcal{M}_x) = 1$ is selected for consolidation, whereas $z_k(\mathcal{M}_x) = 0$ indicates that retaining the memory context is currently preferred. Algorithm 1 summarizes the complete decision procedure.

3.3 SELECTIVE MEMORY CONSOLIDATION

Given the consolidation indicator, COMP determines how each memory-guided RL signal is incorporated during policy optimization. For a retrieved memory group $\mathcal{M}_x$, the selective objective is

$$\mathcal{L}_{\text{CoMP}}^{\mathcal{M}_x}(\theta) = z_k(\mathcal{M}_x)\mathcal{L}_{\text{con}}^{\mathcal{M}_x}(\theta) + (1 - z_k(\mathcal{M}_x))\,\mathcal{L}_{\text{keep}}^{\mathcal{M}_x}(\theta). \quad (11)$$

When $z_k(\mathcal{M}_x) = 1$, the retrieved memory group is removed from the policy context during optimization, encouraging the memory-guided behavior to be represented in the policy parameters. Otherwise, the memory group is retained as external context during the update.

The consolidation decisions are recomputed every K policy iterations to account for changes in the policy. Between refresh steps, the latest decision assigned to each observed memory group is retained. The indicators are treated as fixed during policy optimization, and gradients are not propagated through either $S_{\theta_k}(\mathcal{M}_x)$ or the indicator function. After training, the final policy is evaluated without external memory so that performance reflects capabilities retained in the policy parameters rather than gains from inference-time retrieval. The full training procedure is provided in Algorithm 2.

Algorithm 2 : COMP

Require: $\{\mathcal{X}_k\}_{k=1}^{T}$, memory bank $\mathcal{M}$, policy π_θ, refresh interval K

for $k = 1, \ldots, T$ **do**
  if $k \bmod K = 0$ **then**
    Sample candidate tasks $\mathcal{C}_k$
    for all $x \in \mathcal{C}_k$ **do**
      $\mathcal{M}_x \leftarrow \text{Retrieve}(x, \mathcal{M})$
      $z(\mathcal{M}_x) \leftarrow \text{Decide}(x, \mathcal{M}_x, \pi_\theta)$
    end for
  end if
  $\mathcal{U}_k \leftarrow \varnothing$
  for all $x \in \mathcal{X}_k$ **do**
    $\mathcal{M}_x \leftarrow \text{Retrieve}(x, \mathcal{M})$
    $\mathcal{B}_x^{\mathcal{M}} \sim \pi_\theta(\cdot \mid x, \mathcal{M}_x)$
    if $z(\mathcal{M}_x) = 1$ **then**
      $\mathcal{L}_x \leftarrow \mathcal{L}_{\text{con}}(\theta; \mathcal{B}_x^{\mathcal{M}})$
    else
      $\mathcal{L}_x \leftarrow \mathcal{L}_{\text{keep}}(\theta; \mathcal{B}_x^{\mathcal{M}}, \mathcal{M}_x)$
    end if
    $\mathcal{U}_k \leftarrow \mathcal{U}_k \cup \{\mathcal{L}_x\}$
  end for
  $\mathcal{L}_k \leftarrow \text{Aggregate}(\mathcal{U}_k)$
  $\theta \leftarrow \text{Update}(\theta; \mathcal{L}_k)$
end for
return $\pi_{\theta^\star}(\cdot \mid x, \varnothing)$

In summary, COMP first uses retrieved memories to guide trajectory generation and then decides whether those memories should remain in the policy input during optimization. It compares memory-retained and memory-free updates on the same trajectories and uses a gradient-based gate to select the update better aligned with improving the memory-free policy. These decisions are refreshed throughout training as the policy changes, enabling selective consolidation of memory-guided behaviors into the model parameters.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETTINGS

Models. We use Qwen3-8B (Yang et al., 2025) as the primary backbone for our agent and conduct additional experiments with Qwen3-4B (Yang et al., 2025) and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025) to evaluate the generality of our method across different model scales and families. Unless otherwise specified, memory generation, memory selection, and policy training are performed using the same backbone model. This setup allows us to examine whether the effectiveness of selective memory consolidation is consistent across models with different capacities and post-training recipes.

Datasets. We evaluate COMP on three interactive agent benchmarks: ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2023), and ScienceWorld (Wang et al., 2022), covering household manipulation, web navigation, and scientific reasoning, respectively. Memories are constructed from the model's own rollouts through iterative reflection and distillation (Appendix B.2). We evaluate the trained policy without memory to measure whether memory-induced behaviors are retained in the model parameters, and use ScienceWorld to further assess generalization beyond the training tasks.

Baselines. We compare COMP with two groups of baselines. **Memory-free baselines** include BASE, the backbone model without external memory or task-specific training, and GRPO (Shao et al., 2024), which optimizes the policy using environment rewards alone. **Memory-augmented baselines** include BASE W/ MEMORY, which directly retrieves external memories at inference time; EXPEL (Zhao et al.,

2024) and REFLEXION (Shinn et al., 2023), which leverage accumulated experience or reflection; and MEMRL (Zhang et al., 2026b), which learns memory utility from environmental feedback while keeping the model parameters frozen. We also compare with training-based methods, including GRPO W/ MEMORY, which retains retrieved memories during policy optimization; SKILLRL (Xia et al., 2026), which jointly evolves a skill library and the agent policy; and SDAR (Lu et al., 2026), which uses skill-conditioned guidance for on-policy self-distillation. These comparisons distinguish selective memory consolidation from inference-time retrieval, non-parametric memory learning, and alternative ways of incorporating external memory or skills during policy optimization.

4.2 MAIN RESULTS

Table 1 reports the performance on ALFWorld and WebShop using models from the Qwen family with the vairance analysis presented in Appendix C.2. Additional results on DeepSeek-R1-Distill-Llama-8B are provided in Appendix C.1. Further sensitivity analysis on the number of retrieved memories is presented in Appendix C.3.

		ALFWorld							WebShop	
Model	**Method**	**Place**	**Cool**	**Clean**	**Heat**	**Pick2**	**Light**	**Avg.**	**Task Score**	**Success Rate**
Qwen3-8B	*Inference-based*									
	Base	74.3	20.0	44.4	37.5	16.7	23.1	40.0	35.7	8.4
	Base w/ Memory	91.7(+17.4)	52.4(+32.4)	58.0(+13.6)	69.6(+32.1)	29.4(+12.7)	44.4(+21.3)	59.7(+19.7)	49.9(+14.2)	11.2(+2.8)
	ExpeL	91.7(+17.4)	38.1(+18.1)	32.3(−12.1)	52.2(+14.7)	5.9(−10.8)	44.4(+21.3)	45.5(+5.5)	42.5(+6.8)	13.0(+4.6)
	Reflexion	58.1(−16.2)	81.8(+61.8)	83.3(+38.9)	66.7(+29.2)	16.7(0.0)	46.2(+23.1)	54.3(+14.3)	42.3(+6.6)	11.8(+3.4)
	Training-based									
	GRPO	93.8(+19.5)	66.7(+46.7)	55.0(+10.6)	54.2(+16.7)	86.4(+69.7)	78.6(+55.5)	74.5(+34.5)	75.0(+39.3)	54.4(+46.0)
	GRPO w/ Memory	87.5(+13.2)	71.4(+51.4)	61.3(+16.9)	82.6(+45.1)	76.5(+59.8)	83.3(+60.2)	76.1(+36.1)	79.3(+43.6)	47.6(+39.2)
	SkillRL	91.2(+16.9)	95.2(+75.2)	77.4(+33.0)	73.9(+36.4)	64.7(+48.0)	77.8(+54.7)	80.6(+40.6)	70.5(+34.8)	49.2(+40.8)
	MemRL	79.1(+4.8)	71.4(+51.4)	74.1(+29.7)	43.4(+5.9)	88.2(+71.5)	72.2(+49.1)	70.9(+30.9)	48.2(+12.5)	18.0(+9.6)
	SDAR	95.8(+21.5)	76.2(+56.2)	61.3(+16.9)	56.5(+19.0)	23.5(+6.8)	61.1(+38.0)	64.2(+24.2)	64.8(+29.1)	44.0(+35.6)
	CoMP	**100.0**(+25.7)	81.0(+61.0)	**83.9**(+39.5)	**95.7**(+58.2)	82.4(+65.7)	**94.4**(+71.3)	**89.6**(+49.6)	**80.1**(+44.4)	**59.4**(+51.0)
Qwen3-4B	*Inference-based*									
	Base	79.2	23.8	29.0	39.1	5.9	0.0	32.1	26.8	10.6
	Base w/ Memory	83.3(+4.1)	19.0(−4.8)	52.6(+23.6)	56.5(+17.4)	0.0(−5.9)	5.6(+5.6)	40.3(+8.2)	46.9(+20.1)	18.2(+7.6)
	ExpeL	79.2(0.0)	19.1(−4.7)	41.9(+12.9)	52.2(+13.1)	79.2(+73.3)	16.7(+16.7)	38.8(+6.7)	42.5(+15.7)	11.4(+0.8)
	Reflexion	50.9(−28.3)	53.8(+30.0)	75.0(+46.0)	69.2(+30.1)	5.9(0.0)	72.2(+72.2)	53.0(+20.9)	33.1(+6.3)	18.0(+7.4)
	Training-based									
	GRPO	100.0(+20.8)	76.2(+52.4)	74.2(+45.2)	73.9(+34.8)	29.4(+23.5)	66.7(+66.7)	72.4(+40.3)	65.4(+38.6)	53.4(+42.8)
	GRPO w/ Memory	91.7(+12.5)	76.2(+52.4)	54.8(+25.8)	78.3(+39.2)	35.3(+29.4)	83.3(+83.3)	70.2(+38.1)	65.5(+38.7)	49.2(+38.6)
	SkillRL	100.0(+20.8)	85.7(+61.9)	77.4(+48.4)	60.9(+21.8)	35.3(+29.4)	61.1(+61.1)	72.4(+40.3)	71.3(+44.5)	47.4(+36.8)
	MemRL	66.7(−12.5)	66.7(+42.9)	77.4(+48.4)	60.8(+21.7)	82.3(+76.4)	50.0(+50.0)	67.9(+35.8)	25.4(−1.4)	20.0(+9.4)
	SDAR	70.8(−8.4)	28.6(+4.8)	29.0(0.0)	34.8(−4.3)	23.5(+17.6)	55.6(+55.6)	40.3(+8.2)	61.5(+34.7)	42.0(+31.4)
	CoMP	**100.0**(+20.8)	**90.5**(+66.7)	67.7(+38.7)	**78.3**(+39.2)	**64.7**(+58.8)	**94.4**(+94.4)	**82.1**(+50.0)	**73.7**(+46.9)	**58.2**(+47.6)

Table 1: Performance comparison on ALFWorld and WebShop. Small gray figures in parentheses show each method's difference from *Base* within the same model.

Consistent improvements over strong training baselines. Our method consistently outperforms vanilla GRPO and other training-based baselines, showing that the gains cannot be attributed solely to additional policy optimization. For example, on Qwen3-8B, our method achieves an ALFWorld average of 89.6, compared with 74.5 for GRPO and 80.6 for SkillRL. The improvements are observed across multiple task categories, suggesting that selective consolidation captures broadly useful behaviors rather than overfitting to a specific interaction type. **Robustness across model scales and families.** The effectiveness of selective memory consolidation remains consistent across different model capacities and architectures. In addition to Qwen3 models, we observe clear improvements on DeepSeek-R1-Distill-Llama-8B, where our method increases the ALFWorld average from 44.3 with GRPO to 54.5. This indicates that the proposed consolidation mechanism is not tied to a particular backbone and can generalize across different model families. **Generalization across agent environments.** The benefits of selective consolidation also extend beyond ALFWorld to WebShop, despite the substantial differences in task structure and interaction patterns between the two environments. For example, with Qwen3-4B, our method improves the WebShop task score from 65.4 with GRPO to 73.7, while also improving the success rate from 53.4 to 58.2. These results suggest that selectively consolidating useful memory-induced behaviors provides persistent improvements across diverse agent environments without requiring external memory at deployment.

4.3 COMPARISON WITH ALTERNATIVE CONSOLIDATION CRITERIA

Effect of memory selection strategies. We investigate whether the performance gain comes from the proposed consolidation criterion or simply from exposing the policy to additional memories.

Figure 2 compares our method with several alternative ways of constructing the cumulative consolidation set. *Random Memories* randomly selects memories for consolidation, *Frequent Memories* selects the six most frequently retrieved memories at each selection step, matching the number of memories initially selected by CoMP, and *Unconsolidated Memories* instead consolidates memories excluded by our criterion. We also consider *All Memories*, which removes selection entirely and consolidates every available memory. Our method consistently achieves the best performance. In particular, on Qwen3-8B, consolidating all memories reaches 76.1 and consolidating the complementary unconsolidated set reaches only 71.6, compared with 89.6 for our method. Similar trends are observed on Qwen3-4B, where random, frequency-based, and indiscriminate consolidation all fail to improve over GRPO. These results indicate that the benefit does not arise merely from consolidating more memories; rather, *which* memories are consolidated is critical for persistent policy improvement.

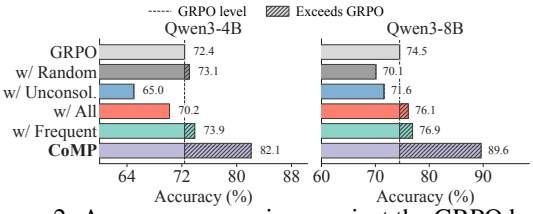
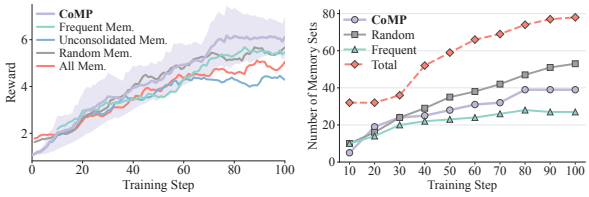


Figure 2: Accuracy comparison against the GRPO baseline on Qwen3-4B and Qwen3-8B.

Training and memory-set dynamics. Figure 3a further shows how different memory selection strategies affect the optimization process. Our method improves more rapidly and reaches a substantially higher training reward than random, frequent, and unconsolidated-memory selection, with the gap becoming particularly pronounced in the later stages of training. While the alternative strategies largely plateau at lower reward levels, our method continues to improve, suggesting that selecting consolidation-worthy memories provides more effective learning signals throughout training. Figure 3b provides a complementary view of how the corresponding memory sets evolve. As the total memory pool continuously grows, the set selected by our method remains substantially smaller than the full pool and grows more conservatively than the random selection set. By the end of training, fewer than half of the available memories are selected for consolidation, yet this compact set yields both higher training rewards and better final performance. Meanwhile, the frequent-memory set quickly saturates around a small subset of repeatedly retrieved memories, but does not provide comparable gains. Together, these results show that the effectiveness of our method is driven neither by consolidating more memories nor by simply prioritizing frequently retrieved ones, but by selectively identifying memories that induce beneficial and persistent policy updates.



(a) Training reward dynamics. (b) Memory set dynamics.

Figure 3: Comparison of different memory selection strategies on ALFWorld. (a) Training reward dynamics on Qwen3-4B. (b) Evolution of the cumulative memory sets selected by different strategies.

4.4 Ablation Study

We ablate the consolidation criterion in CoMP by replacing the proposed directional compatibility signal with two alternative selection signals while keeping the remaining training procedure unchanged. *Advantage Magnitude* selects memories associated with the highest-advantage rollouts, testing whether stronger reward-based learning signals alone are sufficient for identifying memories worth consolidating. *Gradient Magnitude* instead selects memories with the largest memory-free consolidation gradient $\|g^m_{\text{con}}\|$, testing whether a stronger parameter-update signal is sufficient without considering its direction. In contrast, the full CoMP selects memories according to the compatibility between the memory-induced update and the memory-free learning direction.

As shown in Table 2, removing directional compatibility leads to clear performance degradation. Advantage-magnitude selection achieves an average performance of 64.2, while gradient-magnitude selection reaches 72.4, compared with 82.1 for the full method. Although both alternatives capture the strength of the learning signal, neither determines whether the induced update moves the policy in a beneficial memory-free direction. This ablation therefore confirms that the directional information in our consolidation criterion is critical for identifying memories that can be effectively retained.

Method	Place	Cool	Clean	Heat	Pick2	Light	Avg.
Advantage Magnitude	100.0	57.1	54.8	56.5	23.5	88.9	64.2
Gradient Magnitude	95.8	85.7	80.6	87.0	11.8	50.0	72.4
CoMP	100.0	90.5	67.7	78.3	64.7	94.4	**82.1**

Table 2: Ablation study of different criteria components.

4.5 Adaptation on Out-of-distribution Dataset

We further evaluate whether the learned policy generalizes beyond the training task using ScienceWorld. We train on `identify-life-stages-2` and evaluate on two held-out tasks with different degrees of distribution shift. We treat `lifespan-shortest-lived` as the *Near-OOD* task because it remains within a closely related biological domain, while `test-conductivity` serves as the *Far-OOD* task because it requires substantially different scientific knowledge and interaction patterns. As shown in the results, standard GRPO improves performance on the source task from 0.2 to 20.0, but does not yield corresponding transfer, with performance decreasing from 56.3 to 50.0 on Near-OOD and from 14.5 to 7.0 on Far-OOD relative to the base model. In contrast, CoMP improves source-task performance to 36.8 while also achieving 76.3 on Near-OOD and 32.0 on Far-OOD. These results provide evidence that selective memory consolidation can promote more transferable behaviors, mitigating over-specialization to the training task and improving performance under distribution shifts.

Method	Source `identify-life-stages-2`	Near-OOD `lifespan-shortest-lived`	Far-OOD `test-conductivity`
Base	0.2	56.3	14.5
GRPO	20.0	50.0	7.0
CoMP	36.8	76.3	32.0

Table 3: OOD generalization results on ScienceWorld.

4.6 Case study

Figure 4 illustrates two examples of memories consolidated by CoMP and the corresponding policy behaviors in ALFWorld and WebShop. CoMP selects memories based on training signals that assess whether their induced updates improve the memory-free policy. In ALFWorld, the illustrated memory describes placing multiple objects one at a time and opening an enclosed target before placing objects inside. CoMP's successful trajectory follows this procedure by opening the safe and storing the two CDs sequentially, whereas GRPO fails to complete the task. In WebShop, the memory emphasizes confirming the required options before purchasing and proceeding to checkout once the constraints are satisfied. CoMP selects the required size and color and completes the purchase, while GRPO repeatedly switches between options without completing the task. These examples show how memories selected through training signals can encode useful procedural guidance, including action ordering, constraint satisfaction, and task completion. In both cases, CoMP's behavior reflects the guidance in the selected memories. These examples illustrate how memory selection based on training signals can help agents learn useful routines and execute them without external memory.

5 Related Work

Group Relative Policy Optimization. Group Relative Policy Optimization (GRPO) was introduced by DeepSeekMath as a critic-free alternative to Proximal Policy Optimization (PPO) for LLM post-training (Schulman et al., 2017; Shao et al., 2024) to improve the model's reasoning ability (Yue et al., 2025a; Xie et al., 2025) and has since been widely adopted across a range of applications (Hua et al., 2026; Gao et al., 2026; Jacob Dineen et al., 2025; Zhang et al., 2026c; Li et al., 2026). For each prompt, GRPO samples a group of responses and estimates their advantages by normalizing rewards within the group, thereby avoiding a separately trained value model and reducing the memory overhead of policy optimization. DeepSeek-R1-Zero subsequently demonstrated that large-scale GRPO-based reinforcement learning with verifiable rewards can elicit complex reasoning behaviors without

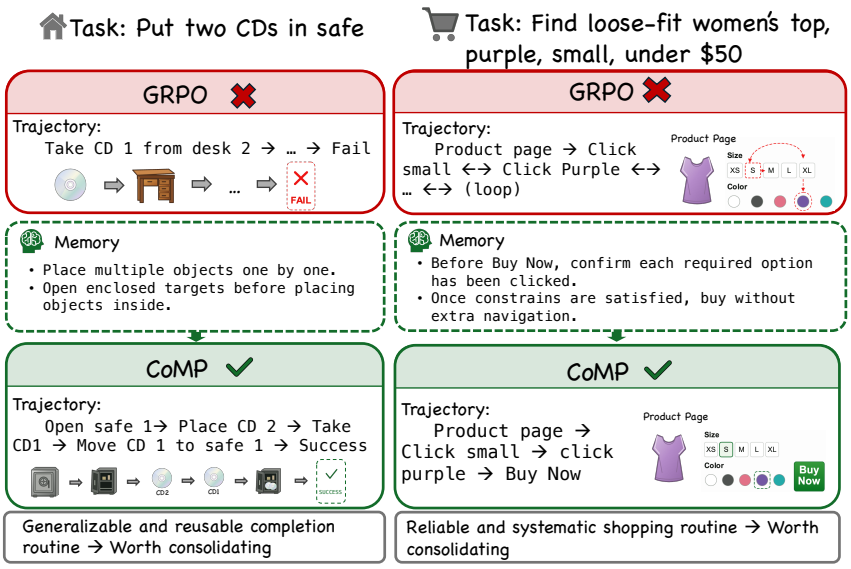


Figure 4: Case study of consolidated memories in Alfworld and Webshop dataset.

supervised reasoning trajectories (Guo et al., 2025). Follow-up work has improved the scalability and credit assignment of group-based policy optimization. DAPO introduces decoupled clipping, dynamic sampling, token-level policy gradients, and overlong-reward shaping for stable large-scale reasoning training (Yu et al., 2026a), while GiGPO constructs both episode-level and step level relative advantages to support fine-grained credit assignment in multi-turn agent environments (Feng et al., 2026). These studies primarily focus on improving the policy optimization objective or its advantage estimator. In contrast, CoMP adopts GRPO as the underlying optimizer and studies an orthogonal problem: how memory-conditioned trajectories should be used to train a policy that no longer requires external memory at inference time.

Memory-Augmented LLM Agents. Existing memory-based LLM agents can be broadly divided into methods that keep the model parameters fixed and methods that incorporate external memory or skills into policy training (Zhang et al., 2026a; Fang et al., 2026; Yu et al., 2026b). In the former category, Reflexion (Shinn et al., 2023) stores verbal self-feedback in an episodic memory buffer, ExpeL (Zhao et al., 2024) extracts transferable insights from past trajectories, and Mem0 (Chhikara et al., 2025) dynamically extracts, organizes, and retrieves salient long-term memories. MemRL (Zhang et al., 2026b) further learns utility values over episodic memories through non-parametric runtime reinforcement learning while keeping the underlying LLM frozen (Zhang et al., 2026b). These approaches improve agent behavior through external retrieval rather than persistent changes to the model parameters. Training-based approaches instead couple memory or skill use with policy optimization. SkillRL recursively co-evolves a retrieved skill bank and the agent policy (Xia et al., 2026). CoMP explicitly compares the policy-update consequences of removing versus retaining each memory and selectively consolidates the memories predicted to better support the final memory-free policy.

6 Conclusion

We introduced CoMP, a selective memory consolidation framework that transfers reusable experience into model parameters. Our central contribution is a selection principle aligned with the goal of consolidation: choosing memories whose induced updates improve the memory-free policy. This criterion connects memory selection directly to learned behavior, enabling agents to internalize useful experience without requiring continued retrieval. Across ALFWorld, WebShop, and ScienceWorld, CoMP outperforms GRPO and indiscriminate consolidation, demonstrating that selecting which memories to consolidate is essential to realizing their learning value. These findings establish selective consolidation as a promising approach to turning accumulated experience into lasting agent capabilities without continued access to external memory.

AI DISCLOSURE

In this work, we used generative AI tools to polish the language, improve the clarity of the manuscript, and assist with the literature survey. We reviewed all AI-assisted revisions and manually verified the relevant literature to assess the originality of our research ideas and ensure appropriate attribution. We take responsibility for the final content of this work, including all text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Yurui Chang, Yongkang Du, Yuanpu Cao, Jinghui Chen, and Lu Lin. Forecastcompass: Guiding agentic forecasting with adaptive factor memory. *arXiv preprint arXiv:2605.30858*, 2026.

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

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 17490–17502, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *Advances in Neural Information Processing Systems*, 38:46375–46408, 2026.

Zijun Gao, Zhikun Xu, Xiao Ye, and Ben Zhou. Core: Concept-oriented reinforcement for bridging the definition–application gap in mathematical reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 78478–78499, 2026.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, November 2020. ISSN 2522-5839. doi: 10.1038/s42256-020-00257-z. URL http://dx.doi.org/10.1038/s42256-020-00257-z.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Kostas Hatalis, Despina Christou, Joshua Myers, Steven Jones, Keith Lambert, Adam Amos-Binks, Zohreh Dannenhauer, and Dustin Dannenhauer. Memory matters: The need to improve long-term memory in llm-agents. In *Proceedings of the AAAI Symposium Series*, volume 2, pp. 277–280, 2023.

Harper Hua, Zhen Han, Zhengyuan Shen, Meng-Chieh Lee, Sheng Guan, Qi Zhu, Sullam Jeoung, Yueyan Chen, Yunfei Bai, Shuai Wang, Vassilis N. Ioannidis, and Huzefa Rangwala. SQL-trail: Multi-turn reinforcement learning with interleaved feedback for text-to-SQL. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 36224–36246, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1677. URL https://aclanthology.org/2026.acl-long.1677/.

Aswin RRV Jacob Dineen, Qin Liu, Zhikun Xu, Xiao Ye, Ming Shen, Zhaonan Li, Shijie Lu, Chitta Baral, Muhao Chen, and Ben Zhou. Qa-lign: Aligning llms through constitutionally decomposed qa. *arXiv preprint arXiv:2506.08123*, 2025.

Feyza Duman Keles, Pruthuvi Mahesakya Wijewardena, and Chinmay Hegde. On the computational complexity of self-attention. In *International conference on algorithmic learning theory*, pp. 597–619. PMLR, 2023.

Haoyuan Li, Zhengyuan Shen, Sullam Jeoung, Yueyan Chen, Jiayu Li, Qi Zhu, Shuai Wang, Vassilis Ioannidis, and Huzefa Rangwala. Boundrl: Efficient structured text segmentation through reinforced boundary generation, 2026. URL https://arxiv.org/abs/2510.20151.

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.

Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, et al. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*, 2026.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Sören Mindermann, Jan Brauner, Muhammed Razzak, Mrinank Sharma, Andreas Kirsch, Winnie Xu, Benedikt Höltgen, Aidan N. Gomez, Adrien Morisot, Sebastian Farquhar, and Yarin Gal. Prioritized training on points that are learnable, worth learning, and not yet learnt, 2022. URL https://arxiv.org/abs/2206.07137.

Fan Nie, Junlin Wang, Harper Hua, Federico Bianchi, Yongchan Kwon, Zhenting Qi, Owen Queen, Shang Zhu, and James Zou. Dsgym: A standardized and holistic framework for evaluating and training data science agents. In *Forty-third International Conference on Machine Learning*.

Randall C O'Reilly, Rajan Bhattacharyya, Michael D Howard, and Nicholas Ketz. Complementary learning systems. *Cognitive science*, 38(6):1229–1248, 2014.

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.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International conference on machine learning*, pp. 31210–31227. PMLR, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context, 2022. URL https://arxiv.org/abs/2209.15189.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Retrieval-augmented generation with conflicting evidence. *arXiv preprint arXiv:2504.13079*, 2025.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. A survey on large language model based autonomous agents. *Frontiers of computer science*, 18(6):186345, 2024.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader?, 2022. URL https://arxiv.org/abs/2203.07540.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. Don't tell the answer, truly guide the reasoning during rl rollouts, 2026. URL https://arxiv.org/abs/2510.09388.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv preprint arXiv:2504.15965*, 2025.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, et al. The rise and potential of large language model based agents: A survey. *Science China information sciences*, 68(2):121101, 2025.

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. *arXiv preprint arXiv:2602.08234*, 2026.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing llm reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2502.14768*, 2025.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2023. URL https://arxiv.org/abs/2207.01206.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026a.

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

Murong Yue. A survey of large language model agents for question answering, 2025. URL https://arxiv.org/abs/2503.19213.

Murong Yue, Zhiwei Liu, Liangwei Yang, Jianguo Zhang, Zuxin Liu, Haolin Chen, Ziyu Yao, Silvio Savarese, Caiming Xiong, Shelby Heinecke, and Huan Wang. Toollibgen: Scalable automatic tool creation and aggregation for llm reasoning, 2025a. URL https://arxiv.org/abs/2510.07768.

Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. In *International Conference on Learning Representations*, volume 2025, pp. 72914–72938, 2025b.

Guibin Zhang, Muxin Fu, Kun Wang, Frank Wan, Miao Yu, and Shuicheng Yan. G-memory: Tracing hierarchical memory for multi-agent systems. *Advances in Neural Information Processing Systems*, 38:12988–13018, 2026a.

Shengtao Zhang, Jiaqian Wang, Ruiwen Zhou, Junwei Liao, Yuchen Feng, Zhuo Li, Yujie Zheng, Weinan Zhang, Ying Wen, Zhiyu Li, et al. Memrl: Self-evolving agents via runtime reinforcement learning on episodic memory. *arXiv preprint arXiv:2601.03192*, 2026b.

Zhi Zhang, Zhen Han, Costas Mavromatis, Qi Zhu, Yunyi Zhang, Sheng Guan, Dingmin Wang, Xiong Zhou, Shuai Wang, Soji Adeshina, Vassilis Ioannidis, and Huzefa Rangwala. Train less, learn more: Adaptive efficient rollout optimization for group-based reinforcement learning, 2026c. URL https://arxiv.org/abs/2602.14338.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

A Derivation of the Memory Consolidation Gate

This appendix derives the local surrogate used by CoMP. Unlike and absolute memory-benefit score, our target compares two counterfactual policy updates: removing a candidate memory during policy optimization and retaining that memory as external training context. Both resulting policies are evaluated without memory.

A.1 Relative Consolidation Objective

For a retrieved memory group $\mathcal{M}_x$, define the two counterfactual parameter displacements as

$$\delta\theta_{\mathrm{con}}(\mathcal{M}_x) = \theta^{+}_{\mathrm{con}}(\mathcal{M}_x) - \theta \tag{12}$$

and

$$\delta\theta_{\mathrm{keep}}(\mathcal{M}_x) = \theta^{+}_{\mathrm{keep}}(\mathcal{M}_x) - \theta. \tag{13}$$

The exact consolidation worthiness is

$$W_\theta(\mathcal{M}_x) = J_\varnothing\left(\theta + \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\right) - J_\varnothing\left(\theta + \delta\theta_{\mathrm{keep}}(\mathcal{M}_x)\right). \tag{14}$$

Let $\mathcal{L}_\varnothing(\theta) = -J_\varnothing(\theta)$ denote the ideal memory-free loss. Equation 14 can equivalently be written as

$$W_\theta(\mathcal{M}_x) = \mathcal{L}_\varnothing\left(\theta + \delta\theta_{\mathrm{keep}}(\mathcal{M}_x)\right) - \mathcal{L}_\varnothing\left(\theta + \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\right). \tag{15}$$

A.2 First-Order Approximation

Proposition 1. *Suppose that $\mathcal{L}_\varnothing$ has an L-Lipschitz continuous gradient in a neighborhood containing $\theta + \delta\theta_{\mathrm{con}}(\mathcal{M}_x)$ and $\theta + \delta\theta_{\mathrm{keep}}(\mathcal{M}_x)$. Let*

$$g_{\mathrm{free}} = \nabla_\theta \mathcal{L}_\varnothing(\theta). \tag{16}$$

Then

$$W_\theta(\mathcal{M}_x) = \langle g_{\mathrm{free}}, \delta\theta_{\mathrm{keep}}(\mathcal{M}_x) - \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\rangle + \mathcal{R}_{\mathcal{M}_x}, \tag{17}$$

where

$$|\mathcal{R}_{\mathcal{M}_x}| \le \frac{L}{2}\Big(\|\delta\theta_{\mathrm{keep}}(\mathcal{M}_x)\|_2^2 + \|\delta\theta_{\mathrm{con}}(\mathcal{M}_x)\|_2^2 \Big). \tag{18}$$

Proof. By the first-order Taylor expansion of $\mathcal{L}_\varnothing$ around θ,

$$\begin{aligned}\mathcal{L}_\varnothing\left(\theta + \delta\theta_{\mathrm{keep}}(\mathcal{M}_x)\right) =& \mathcal{L}_\varnothing(\theta) + \langle g_{\mathrm{free}}, \delta\theta_{\mathrm{keep}}(\mathcal{M}_x)\rangle \\ &+ r_{\mathrm{keep}}(\mathcal{M}_x),\end{aligned} \tag{19}$$

where

$$|r_{\mathrm{keep}}(\mathcal{M}_x)| \le \frac{L}{2}\,\|\delta\theta_{\mathrm{keep}}(\mathcal{M}_x)\|_2^2. \tag{20}$$

Similarly,

$$\begin{aligned}\mathcal{L}_\varnothing\left(\theta + \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\right) =& \mathcal{L}_\varnothing(\theta) + \langle g_{\mathrm{free}}, \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\rangle \\ &+ r_{\mathrm{con}}(\mathcal{M}_x),\end{aligned} \tag{21}$$

where

$$|r_{\mathrm{con}}(\mathcal{M}_x)| \le \frac{L}{2}\,\|\delta\theta_{\mathrm{con}}(\mathcal{M}_x)\|_2^2. \tag{22}$$

Subtracting Eq. 21 from Eq. 19 and using Eq. 15 gives

$$\begin{aligned}W_\theta(\mathcal{M}_x) =& \langle g_{\mathrm{free}}, \delta\theta_{\mathrm{keep}}(\mathcal{M}_x) - \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\rangle \\ &+ r_{\mathrm{keep}}(\mathcal{M}_x) - r_{\mathrm{con}}(\mathcal{M}_x).\end{aligned} \tag{23}$$

Setting

$$\mathcal{R}_{\mathcal{M}_x} = r_{\mathrm{keep}}(\mathcal{M}_x) - r_{\mathrm{con}}(\mathcal{M}_x) \tag{24}$$

and applying the triangle inequality yields Eq. 18. □

Proposition 1 motivates the optimizer-aware surrogate

$$\widehat{W}^{\mathrm{upd}}_\theta(\mathcal{M}_x) = \langle g_{\mathrm{free}}, \delta\theta_{\mathrm{keep}}(\mathcal{M}_x) - \delta\theta_{\mathrm{con}}(\mathcal{M}_x)\rangle. \tag{25}$$

A positive value predicts that the consolidation update yields a better memory-free policy than the memory-retained update.

A.3 GRADIENT-BASED CONSOLIDATION GATE

COMP approximates the two counterfactual updates using their local loss gradients:

$$g_{\text{keep}}^{\mathcal{M}_x} = \nabla_\theta \mathcal{L}_{\text{keep}}^{\mathcal{M}_x}(\theta), \qquad g_{\text{con}}^{\mathcal{M}_x} = \nabla_\theta \mathcal{L}_{\text{con}}^{\mathcal{M}_x}(\theta). \tag{26}$$

Corollary 1. *Suppose both counterfactual updates are approximated by one gradient descent step with the same step size $\eta > 0$:*

$$\delta\theta_{\text{keep}}(\mathcal{M}_x) = -\eta g_{\text{keep}}^{\mathcal{M}_x}, \qquad \delta\theta_{\text{con}}(\mathcal{M}_x) = -\eta g_{\text{con}}^{\mathcal{M}_x}. \tag{27}$$

Then the first-order surrogate in Eq. 25 becomes

$$\widehat{W}_\theta^{\text{grad}}(\mathcal{M}_x) = \eta \left\langle g_{\text{free}}, g_{\text{con}}^{\mathcal{M}_x} - g_{\text{keep}}^{\mathcal{M}_x} \right\rangle. \tag{28}$$

Proof. Substituting the two gradient descent updates into Eq. 25 gives

$$\begin{aligned} \widehat{W}_\theta^{\text{upd}}(\mathcal{M}_x) &= \left\langle g_{\text{free}}, -\eta g_{\text{keep}}^{\mathcal{M}_x} + \eta g_{\text{con}}^{\mathcal{M}_x} \right\rangle \\ &= \eta \left\langle g_{\text{free}}, g_{\text{con}}^{\mathcal{M}_x} - g_{\text{keep}}^{\mathcal{M}_x} \right\rangle. \end{aligned} \tag{29}$$

□

Since $\eta > 0$ is shared by the two counterfactual updates, it does not affect the sign of the comparison. We therefore use

$$S_\theta(\mathcal{M}_x) = \left\langle g_{\text{free}}, g_{\text{con}}^{\mathcal{M}_x} - g_{\text{keep}}^{\mathcal{M}_x} \right\rangle \tag{30}$$

as the local decision statistic for the consolidation gate. Equivalently,

$$S_\theta(\mathcal{M}_x) = \left\langle g_{\text{free}}, g_{\text{con}}^{\mathcal{M}_x} \right\rangle - \left\langle g_{\text{free}}, g_{\text{keep}}^{\mathcal{M}_x} \right\rangle. \tag{31}$$

Thus, the gate compares whether the consolidation update is more locally compatible with improving the memory-free policy than the memory-retained update.

Interpretation. Since g_{free}, $g_{\text{con}}^{\mathcal{M}_x}$, and $g_{\text{keep}}^{\mathcal{M}_x}$ are loss gradients, gradient descent moves in their negative directions. A larger $\langle g_{\text{free}}, g_{\text{con}}^{\mathcal{M}_x} \rangle$ indicates stronger local compatibility between the consolidation update and reduction of the memory-free loss. The difference in Eq. 31 therefore determines which of the two update strategies is preferred by the gate.

A.4 REPLAY-BASED REFERENCE GRADIENT

The ideal memory-free gradient $g_{\text{free}} = \nabla_\theta \mathcal{L}_\varnothing(\theta)$ would require fresh trajectories sampled from the memory-free policy. Repeating this procedure for every retrieved memory group would be computationally expensive. COMP instead constructs a matched replay gradient from the same memory-guided trajectories by removing the retrieved memory group from the policy context while retaining the logged actions, observations, rewards, and advantages:

$$\widetilde{g}_{\text{free}}^{\mathcal{M}_x} = \nabla_\theta \widetilde{\mathcal{L}}_{\text{free}}^{\mathcal{M}_x}(\theta). \tag{32}$$

The practical decision statistic is therefore

$$S_\theta(\mathcal{M}_x) = \left\langle \widetilde{g}_{\text{free}}^{\mathcal{M}_x}, g_{\text{con}}^{\mathcal{M}_x} - g_{\text{keep}}^{\mathcal{M}_x} \right\rangle, \tag{33}$$

which induces the consolidation gate

$$z(\mathcal{M}_x) = \mathbb{I}\left[S_\theta(\mathcal{M}_x) > \gamma\right]. \tag{34}$$

Proposition 2. *Let*

$$d_{\mathcal{M}_x} = \delta\theta_{\text{keep}}(\mathcal{M}_x) - \delta\theta_{\text{con}}(\mathcal{M}_x). \tag{35}$$

Under the smoothness assumption of Proposition 1, define the replay-based estimator

$$\widetilde{W}_\theta(\mathcal{M}_x) = \left\langle \widetilde{g}_{\text{free}}^{\mathcal{M}_x}, d_{\mathcal{M}_x} \right\rangle. \tag{36}$$

Then

$$\left|W_\theta(\mathcal{M}_x) - \widetilde{W}_\theta(\mathcal{M}_x)\right| \leq \frac{L}{2}\Big(\|\delta\theta_{\text{keep}}(\mathcal{M}_x)\|_2^2 + \|\delta\theta_{\text{con}}(\mathcal{M}_x)\|_2^2 \Big) + \left\| g_{\text{free}} - \widetilde{g}_{\text{free}}^{\mathcal{M}_x} \right\|_2 \|d_{\mathcal{M}_x}\|_2 \,. \tag{37}$$

Proof. From Proposition 1,

$$W_\theta(\mathcal{M}_x) = \langle g_{\text{free}}, d_{\mathcal{M}_x} \rangle + \mathcal{R}_{\mathcal{M}_x}. \tag{38}$$

Therefore,

$$W_\theta(\mathcal{M}_x) - \widetilde{W}_\theta(\mathcal{M}_x) = \left\langle g_{\text{free}} - \widetilde{g}_{\text{free}}^{\mathcal{M}_x}, d_{\mathcal{M}_x} \right\rangle + \mathcal{R}_{\mathcal{M}_x}. \tag{39}$$

Applying the Cauchy–Schwarz inequality together with Eq. 18 gives Eq. 37. □

The first term in Eq. 37 captures the higher-order error of the local approximation, while the second captures the error introduced by replacing the ideal memory-free gradient with matched replay. Reusing the same tasks, trajectories, rewards, and advantages across the three gradients also avoids additional environment interactions and controls variation due to different sampled experiences.

A.5 Scope of the Consolidation Gate

The consolidation gate compares two ways of applying the same memory-guided learning signal:

$$\text{remove memory during optimization} \quad \text{vs.} \quad \text{retain memory during optimization}. \tag{40}$$

Accordingly,

$$z(\mathcal{M}_x) = 1 \tag{41}$$

indicates that the local comparison favors the consolidation update over the memory-retained update. Under the approximation above, this corresponds to predicting

$$J_\varnothing\left(\theta_{\text{con}}^+(\mathcal{M}_x)\right) > J_\varnothing\left(\theta_{\text{keep}}^+(\mathcal{M}_x)\right). \tag{42}$$

It does not imply

$$J_\varnothing\left(\theta_{\text{con}}^+(\mathcal{M}_x)\right) > J_\varnothing(\theta). \tag{43}$$

Thus, the gate should be interpreted as a relative decision between two memory-conditioned update strategies, rather than as a predictor of absolute post-update improvement.

B Experimental Details

B.1 Baselines

We provide additional details on the memory-augmented baselines used in our experiments. These baselines cover different ways of leveraging external experience, ranging from inference-time retrieval to joint memory-augmented policy optimization and distillation.

ExpeL. ExpeL (Zhao et al., 2024) learns from previously collected agent experience without updating the model parameters. It collects trajectories from training tasks and extracts reusable natural-language insights from these experiences. At inference time, relevant insights and past experiences are retrieved and included in the context to guide decision making on new tasks. Thus, the knowledge acquired from experience remains external to the policy and must be accessed through retrieval during deployment.

Reflexion. Reflexion (Shinn et al., 2023) improves an agent through verbal self-reflection rather than parameter updates. After receiving task feedback, the agent generates natural-language reflections that summarize observed failures and possible improvements, and stores them in an episodic memory buffer. These reflections are supplied to the agent in subsequent attempts to help it avoid previously observed mistakes. The resulting improvement therefore relies on maintaining reflective memories in the inference context.

GRPO w/ Memory. We construct a direct memory-augmented variant of GRPO (Shao et al., 2024) using the same external memory bank and retrieval mechanism as CoMP. Relevant memories are retrieved and retained in the policy context during reinforcement learning, allowing the policy to optimize its actions while directly conditioning on external guidance. Memory retrieval is also retained at evaluation time. Unlike CoMP, this baseline does not selectively remove memories during optimization or explicitly encourage their behaviors to persist without memory. It therefore isolates the benefit of simply combining external memory retrieval with reinforcement learning.

SkillRL. SkillRL (Xia et al., 2026) combines reinforcement learning with a dynamically evolving external skill library. It first distills successful and failed interaction trajectories into compact reusable skills and organizes them into a hierarchical SkillBank containing general and task-specific guidance. Relevant skills are adaptively retrieved to condition agent execution, while the skill library is further refined from newly observed failures during reinforcement learning. As a result, the policy and the external skill library co-evolve, with retrieved skills continuing to provide contextual guidance for decision making.

SDAR. SDAR (Lu et al., 2026) uses retrieved skills as privileged training-time guidance for on-policy self-distillation. Specifically, the student policy operates without privileged context, while a teacher branch instantiated from the same policy additionally conditions on retrieved skills. SDAR measures the token-level discrepancy between the student and skill-conditioned teacher and uses a gated self-distillation objective together with the standard GRPO objective to transfer useful teacher guidance into the student policy. Unlike methods that directly retain skills in the inference context, the privileged skills are available only to the teacher during training, and the resulting student is evaluated without them.

B.2 Memory construction

Following prior work on reflection- and experience-based memory construction (Shinn et al., 2023; Xia et al., 2026; Zhao et al., 2024), we build the external memory bank from the model's own task executions. For each task, the model first completes an initial attempt and summarizes the resulting experience into a compact memory card that captures reusable guidance, key procedural steps, and common failure patterns. We then validate the generated memory by retrying the same task with the memory provided in context. If the retry fails, the model reflects on the failure and revises or replaces the current memory card, which is subsequently used for another retry. This iterative validation and refinement process continues for a fixed number of rounds or until the task is successfully completed. Finally, the refined experience is distilled into a compact memory representation, removing instance-specific details while retaining broadly applicable guidance. These memories are retrieved as the top-3 entries whose text embeddings are most similar to the question, using the Qwen3-Embedding-0.6B model, and serve as the external memory bank for subsequent training.

Algorithm 3 Iterative Memory Construction

Require: Task x, policy π_θ, maximum refinement rounds R
Ensure: Distilled memory card $m^\star$

```
Execute π_θ on x to obtain the initial trajectory τ_0
Generate an initial memory card m ← GenerateMemory(τ_0)
for r = 1, ..., R do
    Retry x with π_θ conditioned on m to obtain trajectory τ_r
    if τ_r is successful then
        break
    end if
    Generate reflection e_r ← Reflect(τ_r, m)
    Update the memory card m ← UpdateMemory(m, e_r, τ_r)
end for
m* ← Distill(m)
return m*
```

B.3 TRAINING CONFIGURATION

We summarize the main training hyperparameters for ALFWorld and WebShop in Table 4. Unless otherwise specified, all methods use the same optimization configuration for a fair comparison. We train for 100 optimizer steps with a batch size of 16 and a group size of 8. Each episode is limited to 50 environment steps, with a maximum response length of 512 tokens. We use a learning rate of 1×10^{-6} and a KL coefficient of 0.01. For CoMP, the consolidation criterion is refreshed every 10 optimizer steps, with no memory-free warmup. The same configuration is used for WebShop, except that we set the maximum prompt length to 1,024 tokens.

Hyperparameter	**Value**
Optimizer steps	100
Training batch size	16
Group size	8
PPO mini-batch size	128
Maximum response length	512
Maximum environment steps	50
Learning rate	1×10^{-6}
KL coefficient	0.01
Memory-free warmup steps	0
Screening refresh frequency	Every 10 steps

Table 4: Training configuration

C MORE EXPERIMENTAL RESULTS

C.1 ADDITIONAL RESULTS ON OTHER MODEL FAMILIES

Method	**Place**	**Cool**	**Clean**	**Heat**	**Pick2**	**Light**	**Avg.**
Base	41.4	0.0	0.0	0.0	0.0	38.1	14.9
GRPO	70.8	42.9	54.8	30.4	0.0	44.4	44.3
CoMP	**79.2**	**52.4**	**71.0**	**34.8**	**5.9**	**66.7**	**54.5**

Table 5: Performance comparison on ALFWorld with DeepSeek-R1-Distill-Llama-8B.

Generalization across model families. We further evaluate our method on DeepSeek-R1-Distill-Llama-8B to examine whether the benefits of selective memory consolidation transfer beyond the Qwen model family. On ALFWorld, our method improves the average performance from 44.3 with GRPO to 54.5, with consistent gains across nearly all task categories. This result shows that the proposed consolidation strategy is not specific to a particular backbone and can effectively transfer useful memory-induced behaviors into policies with different model architectures and training recipes.

C.2 VARIANCE ANALYSIS

To assess whether the observed performance differences are statistically reliable, we conduct a bootstrap significance test using GRPO as the reference method on Qwen3-8B. As shown in Table 6, our method achieves statistically significant improvements over GRPO on both ALFWorld and WebShop. On ALFWorld, our method improves the success rate by 15.0 points, with a 95% confidence interval of [7.56, 22.49] and $p < 0.001$. On WebShop, it yields a 5.1-point improvement, with a 95% confidence interval of [2.31, 8.00] and $p < 0.001$. In comparison, SkillRL does not significantly outperform GRPO on ALFWorld and performs significantly worse on WebShop, while ExpeL is significantly worse than GRPO on both benchmarks. These results provide additional evidence that the gains from our method are robust and unlikely to arise from sampling variability.

C.3 SENSITIVITY ANALYSIS

Benchmark	Method	Accuracy	Δ vs. GRPO	95% CI	p-value
ALFWorld	GRPO	74.5	–	–	–
	SkillRL	80.6	6.1	[-2.14, 14.08]	0.179
	ExpeL	45.5	-29.0	[-38.71, -19.30]	< 0.001
	CoMP	89.6	15.0	[7.56, 22.49]	< 0.001
WebShop	GRPO	75.0	–	–	–
	SkillRL	70.5	-4.5	[-7.90, -1.10]	0.009
	ExpeL	43.5	-31.5	[-34.10, -28.80]	< 0.001
	CoMP	80.1	5.1	[2.31, 8.00]	< 0.001

Table 6: Bootstrap significance test results on ALFWorld and WebShop.

Sensitivity to the number of retrieved memories. We further study the effect of the number of retrieved memories k used during training. Accuracy refers to the accuracy rate for AlfWorld and task score for WebShop dataset. As shown in Table 7, increasing k from 1 to 3 substantially improves performance on ALFWorld, while further increasing it to 5 yields a smaller additional gain. This suggests that exposing the policy to a richer set of relevant memories provides more useful consolidation opportunities, but the marginal benefit diminishes as k grows. Overall, the method remains effective across different retrieval sizes, indicating that its performance is not overly sensitive to a particular choice of k.

k	ALFWorld
1	74.6
3	82.1
5	**84.3**

Table 7: Sensitivity analysis of k on ALFWorld with Qwen3-4B.

D MORE CASE STUDIES

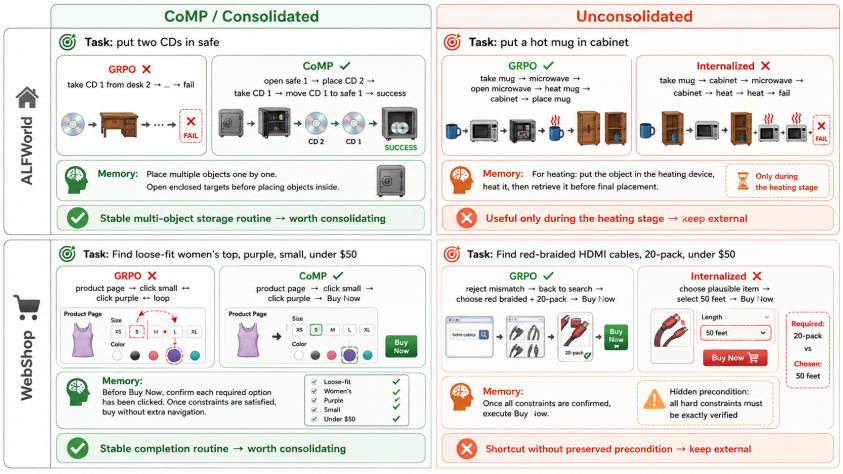


Figure 5: Case study of consolidated and unconsolidated memories.

Figure 5 illustrates four representative memory internalization behaviors across ALFWorld and WebShop. The first and third cases show desirable internalization: the learned memory provides reusable procedural structure that corrects GRPO failures, such as decomposing multi-object placement into sequential actions or completing purchase once all required options are verified. The second case highlights a stage-dependent memory whose usefulness is restricted to a specific subgoal; internalizing it as a general action tendency can disrupt the original task progression. The fourth case provides a counterexample in which an overly simplified purchasing heuristic causes the agent to commit before verifying all hard constraints. Together, these examples show that effective consolidation

requires not only identifying useful memories, but also preserving the conditions under which their guidance is valid.

E PROMPTS

Here is the prompt used to reflect on trajectories to extract raw memories and subsequently summarize them into high-level skills.

```
You are an expert Strategic Analyst for AlfWorld Agent. Your goal is
to extract high-level strategic_guidelines focusing on the execution
skeleton and error avoidance. First check the trajectory Outcome.

Case 1: SUCCESS

planning_pattern: Abstract the successful trajectory into a
high-level logical chain using -> to separate major steps. Never use
specific object or location names. Replace objects with [Object_1],
[Object_2], search areas with [Location], and fixed destinations with
[Target_Location].

Before interacting with any object, explicitly include its search phase
using the form "Search [Location] where possibly having [Object_X] ->
[Action] [Object_X]". Prefer standard actions such as Acquire, Use,
Heat, Cool, Clean, and Place. If multiple objects are in the same
location, combine their search phase; otherwise, represent their
searches separately. Use "Navigate to [Target_Location]" only when
moving to a known appliance or placement destination after obtaining
the required object.

mistakes_to_avoid: Return an empty list [].

Case 2: FAILURE

planning_pattern: Return null.

mistakes_to_avoid: Identify the root cause of failure and express it
as a generalizable rule. Never use specific object or location names;
instead use abstract terms such as [Target_Object], [Similar_Object],
[Container], and [Location].

Each item should contain only:
- trigger_condition: the abstract context where the error occurred.
- bad_action: the abstract incorrect action taken.

Output Structure:

{
"planning_pattern": "String or null",
"mistakes_to_avoid": [
{
"trigger_condition": "Abstract context where the error occurred.",
"bad_action": "Abstract incorrect action."
}
]
}

Output ONLY the JSON object with keys planning_pattern and
mistakes_to_avoid.
```

Table 8: Prompt template for extracting reusable planning patterns and failure-avoidance memories from ALFWorld trajectories.

You are an expert **Strategic Analyst for a WebShop Autonomous Agent.**

Your goal is to extract high-level strategic_guidelines from a WebShop trajectory. First inspect the trajectory outcome.

Case 1: Successful Trajectory

planning_pattern: Abstract the successful trajectory into a reusable high-level logical chain, using -> to separate major stages. Use generalized action types such as Form Constraint-Aware Search, Inspect Candidate Products, Verify Required Attributes, Select Required Options, Confirm Constraint Satisfaction, and Purchase Product.

When an attribute is verified or selected, annotate the stage with the ATTRIBUTE TYPE (e.g., size, color, material, wattage), but never include specific attribute values, product names, brands, IDs, or exact search queries. Preserve only stages supported by the trajectory. Also preserve step economy: if the agent succeeds after inspecting only a few candidates, do not add unnecessary verification or browsing stages.

mistakes_to_avoid: Return an empty list: []

Case 2: Failed Trajectory

planning_pattern: Return null.

mistakes_to_avoid: Identify the root cause of failure as a reusable negative constraint. Each item must contain:

- trigger_condition: The abstract context in which the error occurred.
- bad_action: The incorrect or missing action that caused the failure.

Check step-budget failure first. If the agent exhausts its step budget without ever reaching click[buy now] or completing a purchase, explicitly report that it continued searching, browsing, or re-verifying instead of committing to a reasonable candidate before the budget expired. Do not describe this only as generic "looping."

For other failures, identify concrete causes such as missing required attribute verification, selecting the wrong option, relying only on search-result snippets, failing to verify price, using a poor search query, or purchasing the wrong product variant.

Generalize all failures: never include specific product names, brands, IDs, or attribute values. Do preserve the relevant ATTRIBUTE TYPE (e.g., size, color, material, connector type) so that the resulting guideline remains category-specific and reusable.

Output Structure

```
{
"planning_pattern": "String or null",
"mistakes_to_avoid": [
{
"trigger_condition": "Abstract context where the error occurred.",
"bad_action": "Abstract incorrect action or missing verification."
}
]
}
```

Output ONLY the JSON object with exactly the two keys planning_pattern and mistakes_to_avoid.

Table 9: Prompt template for extracting reusable strategic guidelines from WebShop trajectories.

```
You are an expert agent-memory summarizer.

You will be given a list of planning_pattern strings extracted from
WebShop trajectories.  Your job is to produce 5-7 high-level GENERAL
MEMORIES that capture common strategies transferable across product
types.  Each memory should be concise, actionable, and suitable for
inclusion in a system prompt.

Output format (strict JSON list, no preamble, no markdown fences):

[
{
"title":  "<Title Case short name, 3-5 words>",
"principle":  "<one or two actionable sentences ending with a period>",
"when_to_apply":  "<one sentence describing when to apply the memory>"
},
...
]

Constraints:

- Produce 6-8 self-contained memories without bracketed placeholders.
- Focus on constraint-aware search, candidate selection, option
 configuration, attribute and price verification, navigation efficiency,
 and purchasing decisions.
- WebShop has a hard step limit shared across searching, browsing,
 verification, and purchasing.  At least two memories must enforce
 concrete decisiveness rules that lead to click[buy now] on the
 best available candidate rather than unnecessary re-searching or
 re-verification.
- Use plain action verbs such as "click", "buy", "purchase", and
"select"; avoid tool-call terminology such as "execute", "invoke",
"function", "trigger", or "call".
```

Table 10: Prompt template for summarizing successful WebShop trajectories into reusable high-level memories.