AERCA: ADAPTIVE EXPERIENCE REGULATION AND CREDIT ASSIGNMENT FOR LONG-HORIZON AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliable end-to-end completion remains challenging for LLM agents in long-horizon tasks. We identify two recurring obstacles. First, retrieved experience can remain influential after becoming poorly matched to the agent's evolving state or information need, causing previously useful guidance to bias subsequent decisions. Second, long-horizon RL exhibits a credit assignment dilemma: some credit schemes improve strict completion while leaving many unresolved trajectories with limited progress, whereas milestone-aware credit can substantially improve intermediate progress without yielding correspondingly stronger completion. To address these challenges, we introduce **AERCA** (**A**daptive **E**xperience **R**egulation and Memory-Aware **C**redit **A**ssignment), which couples adaptive experience use during execution with selective credit assignment during policy learning. AERCA structures past interactions into Task-level Completion Experience for *how to complete* and Gap-level Cognition Experience for *how to know*, while dynamically regulating whether retrieved experience should be used, retained, or discarded as execution evolves. For policy learning, relative trajectory progress determines how much memory-aware credit a trajectory receives, while breakthroughs during memory use events determine where that credit is assigned. On ScienceWorld, AERCA achieves 45.08% full success with a task score of 30.70, while reaching 78.13% success and a task score of 87.83 on WebShop. More importantly, even when all compared methods use our Adaptive Experience Regulation, AERCA improves full success over GiGPO by 4.67 percentage points on ScienceWorld, isolating the contribution of Memory-Aware Credit Assignment. Ablation studies further support the complementary contributions of the two components. Our code and datasets are available at https://anonymous.4open.science/r/AERCA-6FF2/.

1 INTRODUCTION

Large language models (LLMs) are increasingly deployed as autonomous agents for complex, long-horizon tasks requiring multi-step interaction with an environment (Yao et al., 2022b; Liu et al., 2024; Zeng et al., 2024; Lin et al., 2023; Deng et al., 2023; Zhou et al., 2024a). Training such agents is challenging because task outcomes are often sparse and delayed over extended interaction horizons (Zhou et al., 2024b; Wang et al., 2025; Xi et al., 2025). Recent RL methods therefore introduce finer-grained supervision or credit over intermediate execution (Feng et al., 2026; Wang et al., 2026; Zhang et al., 2026c; Li et al., 2026), while experience-augmented agents reuse reflections, skills, or episodic memories to guide future behavior (Shinn et al., 2023; Xia et al., 2026; Zhang et al., 2026b; Liu et al., 2026). Despite these advances, reliable end-to-end completion remains challenging, especially over long interaction horizons.

Two complementary obstacles hinder reliable completion over long horizons as illustrated in Figure 1. First, experience-augmented agents increasingly reuse retrieved reflections, procedural knowledge, or episodic memories during execution (Shinn et al., 2023; Zhao et al., 2024; Xu et al., 2026; Zhang et al., 2026b). We observe that such experience can remain influential even after becoming poorly matched to the agent's current state or information need, causing previously useful guidance to bias later decisions. Second, although recent long-horizon RL methods introduce step- or milestone-level credit to improve multi-step policy learning (Feng et al., 2026; Wang et al., 2026; Zhang et al., 2026c; Li et al., 2026), they exhibit a credit assignment dilemma. For example,

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

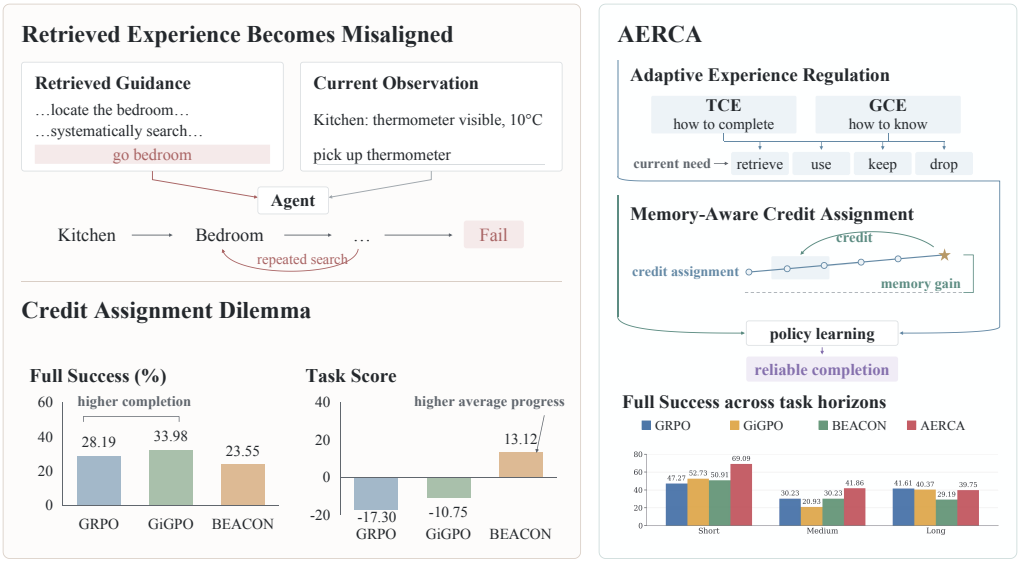


Figure 1: **Left:** Retrieved experience can become misaligned, while progress and completion may diverge. **Right:** AERCA combines Adaptive Experience Regulation and Memory-Aware Credit Assignment for reliable long horizon completion.

GRPO (Shao et al., 2024) and GiGPO (Feng et al., 2026) achieve stronger strict completion while leaving many unresolved trajectories with limited progress, whereas the milestone-aware credit used in BEACON (Wang et al., 2026) substantially improves intermediate progress without yielding correspondingly stronger completion.

To address these challenges, we introduce **AERCA** (**A**daptive **E**xperience **R**egulation and Memory-Aware **C**redit **A**ssignment), which couples adaptive regulation of retrieved experience during execution with selective credit assignment during policy learning. Specifically, the Adaptive Experience Regulation component organizes past interactions into Task-level Completion Experience (TCE), which captures how to complete a task, and Gap-level Cognition Experience (GCE), which captures how to resolve information gaps. As execution evolves, it dynamically decides whether retrieved experience should be used, retained, or discarded. In addition, the Memory-Aware Credit Assignment component converts relative trajectory progress into selective supervision for memory use, determining how much memory-aware credit each trajectory receives. It then localizes this credit to memory-use events associated with subsequent environment-verified breakthroughs.

We evaluate AERCA on ScienceWorld (Wang et al., 2022) and WebShop (Yao et al., 2022a). On ScienceWorld, AERCA achieves 45.08% full success with a task score of 30.70, attaining the strongest performance on both metrics among the evaluated methods. On WebShop, it reaches 78.13% success with a task score of 87.83. More importantly, under the same Adaptive Experience Regulation, AERCA improves full success over GiGPO by 4.67 percentage points on ScienceWorld, isolating the contribution of Memory-Aware Credit Assignment.

Our main contributions are:

- We identify two obstacles to reliable long horizon completion: retrieved experience can become poorly matched as execution evolves, and existing credit schemes can favor either intermediate progress or strict completion without improving both consistently.
- We propose **AERCA**, which organizes experience for task completion and information needs, dynamically regulates its influence during execution, and assigns memory credit using relative trajectory progress and subsequent breakthroughs during memory use.
- Experiments on ScienceWorld and WebShop show strong completion and graded progress, reaching 45.08% full success with a task score of 30.70 on ScienceWorld and 78.13%

success with a task score of 87.83 on WebShop. Under shared Adaptive Experience Regulation, AERCA further improves full success over GiGPO by 4.67 percentage points.

2 Diagnosing Obstacles to Reliable Long-Horizon Completion

We examine two obstacles to reliable long-horizon completion on ScienceWorld: retrieved experience that becomes poorly matched to the current interaction, and a credit assignment dilemma between intermediate progress and strict completion. Figure 2 summarizes the corresponding qualitative and quantitative evidence.

2.1 When Retrieved Experience No Longer Matches the Current State

Experience-based agents increasingly retrieve reflections, procedural knowledge, or learned memories during execution (Zhao et al., 2024; Fang et al., 2026; Cao et al., 2026; Xu et al., 2026; Yu et al., 2026; Yan et al., 2026; Zhang et al., 2026a;b). However, experience that is useful when retrieved may become poorly matched as the agent's state and information needs evolve.

Retrieved experience can conflict with current evidence. Figure 2(a) illustrates a representative failure from our RetroAgent (Zhang et al., 2026b) reproduction on the ScienceWorld `use-thermometer` task. The retrieved experience encourages searching the bedroom for the thermometer, while the current observation already reveals the thermometer in the kitchen. The agent nevertheless leaves the kitchen and repeatedly searches the bedroom, eventually failing the task. This example shows that effective experience reuse requires not only retrieving useful experience, but also regulating whether it should continue to influence execution as new evidence becomes available.

2.2 Credit Assignment Dilemma in Long Horizon RL

Recent long horizon RL methods increasingly introduce finer grained credit signals to improve multi-step policy learning (Feng et al., 2026; Wang et al., 2026; Zhang et al., 2026c; Li et al., 2026). Under the same strict completion criterion, we examine how different credit schemes affect intermediate task progress and eventual completion.

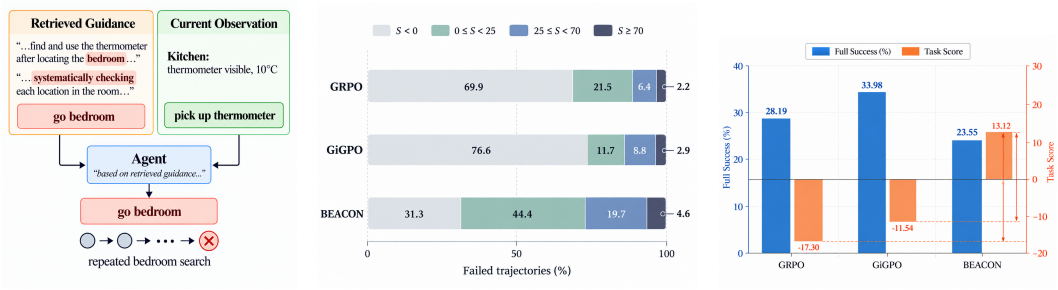
Many unresolved trajectories still make limited progress. Figure 2(b) shows the task score distribution of failed trajectories after training. For GRPO and GiGPO, 69.9% and 76.6% of failed trajectories, respectively, still have negative task scores. BEACON changes this failure profile: the fraction of negative score failures decreases to 31.3%, while 24.3% reach $S \geq 25$ and 4.6% reach $S \geq 70$. These results suggest that credit based on stronger milestone supervision can move unresolved trajectories toward greater intermediate progress.

Intermediate progress and strict completion can diverge. Figure 2(c) compares average task score with full success under the same strict completion criterion. GRPO and GiGPO achieve 28.19% and 33.98% full success, respectively, while their average task scores remain negative at -17.3 and -11.5. In contrast, BEACON reaches a positive average task score of 13.12 but only 23.55% full success. Together, these results reveal a **credit assignment dilemma**. BEACON moves many failed trajectories toward higher task scores, but this additional progress does not translate into higher full success. In contrast, GRPO and GiGPO achieve stronger completion while leaving many failed trajectories with little progress. Effective credit assignment therefore needs to exploit intermediate progress without weakening the signal for complete task satisfaction.

3 Method

3.1 Problem Formulation

Task Formulation. We consider a long horizon agent interacting with an environment with state space $\mathcal{S}$, action space $\mathcal{A}$, observation space $\mathcal{O}$, transition function F, and task reward R. At step t, the agent represents its interaction history as $s_t = (o_0, a_0, \ldots, a_{t-1}, o_t)$ and samples $a_t \sim \pi_\theta(\cdot \mid s_t)$, producing a trajectory τ. The policy parameters θ are optimized to maximize the expected task return



(a) **Mismatch between retrieved experience and current evidence** (b) **Progress among failed trajectories** (c) **Progress versus completion**

Figure 2: **Diagnosis of obstacles to reliable long-horizon completion.** **(a)** Retrieved experience conflicting with current evidence. **(b)** Task score distribution among failed trajectories. **(c)** Task progress and full success under different RL credit schemes.

$\mathbb{E}_{\tau\sim\pi_\theta,F}[R(\tau)]$. For memory-augmented agents, retrieved experience m_t is provided as additional context for action generation, $a_t \sim \pi_\theta(\cdot \mid s_t, m_t)$.

Operationalizing Adaptive Experience Use in LLM Agents. We extend the above formulation by explicitly representing how experience is managed during agent execution. At step t, the LLM agent observes the interaction state s_t together with the current experience state, consisting of an optional candidate m_{t-1}^{cand} and active experience m_{t-1}^{act}. The agent then produces a memory decision d_t^{mem}; retrieval is performed only when requested, and the experience state is updated before the agent generates its environment action:

$$\left(m_t^{\text{cand}}, m_t^{\text{act}}\right) = U\left(m_{t-1}^{\text{cand}}, m_{t-1}^{\text{act}}, s_t, d_t^{\text{mem}}\right), \qquad a_t \sim \pi_\theta(\cdot \mid s_t, m_t^{\text{act}}), \tag{1}$$

where U denotes the state update induced by the agent's memory decision; a new candidate is retrieved only when the agent chooses RETRIEVE. The resulting trajectory is $\tau^{\text{mem}} = \{(o_t, m_t^{\text{cand}}, d_t^{\text{mem}}, m_t^{\text{act}}, a_t, S_t)\}_{t=0}^{T-1}$, where S_t denotes the environment task score after step t.

3.2 ADAPTIVE EXPERIENCE REGULATION

To alleviate the adverse effect of retrieved experience that becomes inconsistent with current observations, Adaptive Experience Regulation instantiates the above formulation with complementary experience roles and lifecycle decisions made by the agent.

Experience for Complementary Roles. We organize past interactions according to two complementary roles:

$$\mathcal{M}^{\text{TCE}} = (\text{goal}, \text{protocol}, \text{summary}),$$
$$\mathcal{M}^{\text{GCE}} = (\text{trigger}, \text{protocol}, \text{expected effect}, \text{summary}).$$

Task-level Completion Experience (TCE) is distilled from successful trajectories and provides reusable experience for *how to complete*. Gap-level Cognition Experience (GCE) is extracted from locally informative segments of failed trajectories that lead to increases in task score, capturing reusable experience for *how to know*. The two forms respectively target overall completion and local information needs.

Dynamic Regulation of Experience. We instantiate the lifecycle update U in Eq. 1 through four agent decisions, $d_t^{\text{mem}} \in \{\text{RETRIEVE}, \text{USE}, \text{KEEP}, \text{DROP}\}$. At step t, TCE retrieval is mainly conditioned on the task, whereas GCE also uses the current observation and cognition gap G_t. RETRIEVE obtains a candidate experience m_t^{cand}, USE adopts it as the active experience m_t^{act}, KEEP retains the current active experience, and DROP removes its influence. A later RETRIEVE followed by USE can therefore replace outdated experience as execution evolves. We refer to the lifecycle of an adopted

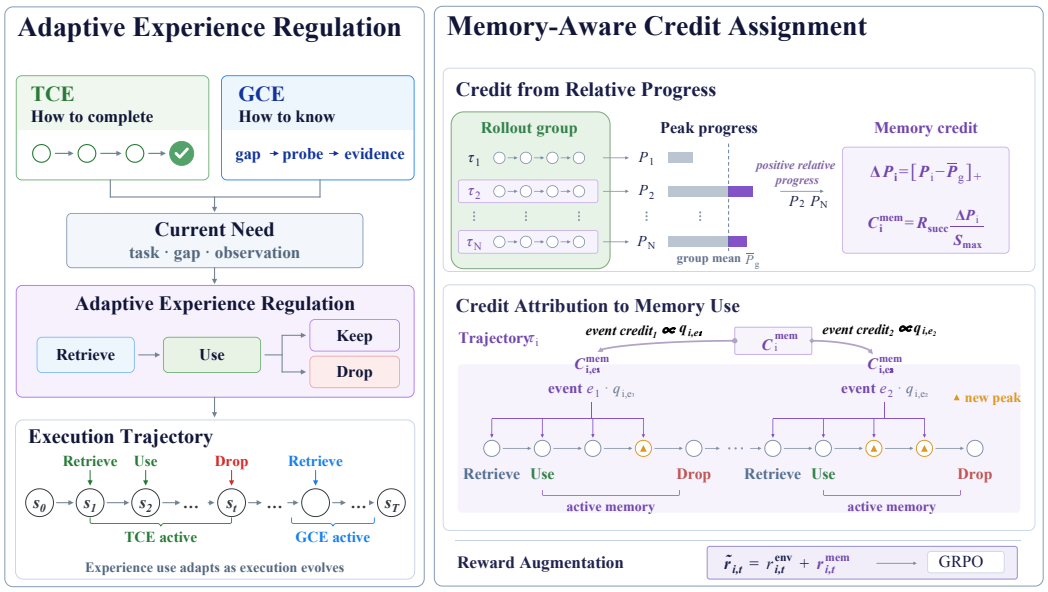


Figure 3: **Overview of AERCA. Left:** Adaptive Experience Regulation organizes Task-level Completion Experience (TCE) and Gap-level Cognition Experience (GCE) and dynamically regulates their use during execution. **Right:** Memory-Aware Credit Assignment uses relative trajectory progress to determine how much credit a trajectory receives and task breakthroughs during memory use events to determine where that credit is assigned.

candidate as a memory use event. For each event e, t_r^e, t_u^e, and t_d^e denote its retrieval, use, and ending steps, respectively. The event spans $[t_r^e, t_d^e)$, while the experience influences execution during its active interval $[t_u^e, t_d^e)$, until it is dropped, replaced, or the episode terminates.

3.3 Memory-Aware Credit Assignment

Section 2.2 shows that stronger intermediate progress does not necessarily translate into reliable completion. We therefore assign memory-aware credit in two stages: relative trajectory progress determines *how much* credit a trajectory receives, while breakthroughs during each memory use event determine *where* that credit is allocated. This turns coarse trajectory progress into selective supervision over experience use, providing a practical approximation for localizing credit according to subsequent task progress associated with memory use.

Credit from Relative Progress. For each task variation g, let $\mathcal{G}_g$ denote its rollout group and $S_{i,t}$ the task score of trajectory τ_i^{mem} after step t. We measure each trajectory by its peak score improvement and compare it with the rollout group mean:

$$P_i = \max\Big(0, \max_t S_{i,t} - S_{i,0}\Big), \qquad P_g = \frac{1}{|\mathcal{G}_g|}\sum_{j\in\mathcal{G}_g} P_j, \qquad \Delta P_i = [P_i - P_g]_+. \tag{2}$$

We then map this relative progress to the native reward scale of the environment:

$$C_i^{\text{mem}} = R_{\text{succ}}\frac{\Delta P_i}{S_{\max}}, \tag{3}$$

where $S_{\max}$ and R_{succ} are fixed quantities defined by the environment, denoting the completion score and the corresponding success reward, respectively.

Credit Attribution to Memory Use. Credit derived from relative progress determines which trajectories receive memory-aware credit, but not which memory use events are associated with the observed progress. We therefore attribute this credit using task breakthroughs that occur during

memory use. Let $\mathcal{E}_i$ denote the set of memory use events in τ_i^{mem}. For each event, we measure the breakthrough progress established during its active interval:

$$\delta P_{i,t} = \left[S_{i,t+1} - \max_{0 \le v \le t} S_{i,v} \right]_+, \qquad q_{i,e} = \sum_{t=t_u^e}^{t_d^e - 1} \delta P_{i,t}. \tag{4}$$

When $\sum_{e \in \mathcal{E}_i} q_{i,e} > 0$, we attribute trajectory credit to memory use events in proportion to their breakthrough scores and distribute each event's credit across its event interval $[t_r^e, t_d^e)$:

$$C_{i,e}^{\text{mem}} = \frac{q_{i,e}}{\sum_{e' \in \mathcal{E}_i} q_{i,e'}} C_i^{\text{mem}}, \qquad r_{i,t}^{\text{mem}} = \sum_{\substack{e \in \mathcal{E}_i: \\ t_r^e \le t < t_d^e}} \frac{C_{i,e}^{\text{mem}}}{t_d^e - t_r^e}. \tag{5}$$

This preserves the total credit, $\sum_t r_{i,t}^{\text{mem}} = C_i^{\text{mem}}$. If no valid breakthrough occurs during memory use, we apply the credit at the trajectory level without event attribution.

Finally, the memory credit is added to the environment reward, $\tilde{r}_{i,t} = r_{i,t}^{\text{env}} + r_{i,t}^{\text{mem}}$, and the resulting policy is optimized using standard GRPO (Shao et al., 2024).

4 Experiments

4.1 Experimental Setup

Benchmarks and Metrics. We evaluate on two long-horizon benchmarks: ScienceWorld (Wang et al., 2022) and WebShop (Yao et al., 2022a). ScienceWorld contains 30 scientific reasoning task types requiring multi-step exploration, manipulation, and reasoning. We report **Full Success (Full Succ.)** and **Task Score (Score)**. Full success is the percentage of episodes with a final score of 100, while task score is the average final environment score across evaluation episodes. We use "full" because some prior ScienceWorld implementations count any positive final score as success (Wang et al., 2026; Zhang et al., 2026c). Detailed metric definitions and comparison with the released BEACON success criterion are provided in Appendix A.2.

WebShop evaluates multi-step product search and purchase under compositional natural-language requirements, for which we report success rate and task score following the standard evaluation protocol. For AERCA, results are reported as mean and standard deviation over five runs. Additional benchmark details are provided in Appendix A.1.

Compared Methods. We compare against two groups of learning-based baselines. **(i) RL-Based Agentic Learning** includes GRPO as the standard group-based RL baseline; GiGPO (Feng et al., 2026), which introduces step-level relative advantages through anchor-state grouping; BEACON (MiGPO) (Wang et al., 2026), which incorporates milestone-based trajectory decomposition and multi-scale credit; and RLVMR (Zhang et al., 2026c), which augments outcome rewards with verifiable process-level supervision. **(ii) Memory-Augmented RL Agents** include RetroAgent (Zhang et al., 2026b), which combines retrospective feedback with intrinsic progress rewards; EMPO^2 (Liu et al., 2026), which reuses memories distilled from previous rollouts with hybrid policy optimization; and MemRL (Zhang et al., 2026a), which learns environment-grounded utility for episodic memories. Together, these methods cover representative RL-based and memory-augmented approaches to long-horizon agent learning.

Implementation. We use Qwen2.5-7B-Instruct as the primary policy backbone. Generalization results with Qwen3-4B-Instruct are provided in Appendix D.2. All RL methods use identical training and test splits and follow the same evaluation protocol. Detailed hyperparameter settings and training configurations are provided in Appendix A.

4.2 Main Results

Overall Performance. Table 1 summarizes the main results. On ScienceWorld, AERCA achieves 45.08% full success with a task score of 30.70, attaining the strongest performance on both metrics.

Table 1: **Main results on long horizon interactive benchmarks.** We compare AERCA with competitive RL and memory baselines. Best and second best results are shown in **bold** and underlined, respectively. * denotes results reported in prior work (Xia et al., 2026; Zhang et al., 2026b); all other results are reproduced under our experimental setup.

	ScienceWorld		WebShop	
Method	**Full Succ. ↑**	**Score ↑**	**Succ. ↑**	**Score ↑**
Qwen2.5-7B-Instruct				
Zero-Shot	$11.29_{\pm 0.86}$	$-30.42_{\pm 3.56}$	$7.8_{\pm 0.0}$	$26.4_{\pm 1.2}$
RL-Based Agentic Learning				
GRPO (Shao et al., 2024)	$28.19_{\pm 0.39}$	$-7.94_{\pm 9.66}$	$63.67_{\pm 1.65}$	$78.51_{\pm 1.39}$
GiGPO (Feng et al., 2026)	$32.24_{\pm 2.46}$	$-10.75_{\pm 1.12}$	$69.53_{\pm 1.10}$	$80.96_{\pm 0.97}$
BEACON (Wang et al., 2026)	$21.67_{\pm 1.88}$	$26.06_{\pm 1.98}$	$72.92_{\pm 0.90}$	$79.98_{\pm 2.48}$
RLVMR (Zhang et al., 2026c)	$\underline{35.91_{\pm 0.67}}$	$21.74_{\pm 5.19}$	$65.96_{\pm 1.06}$	$82.08_{\pm 3.44}$
Memory-Augmented RL Agents				
Mem0 (Chhikara et al., 2025)+GRPO	–	–	37.5^{*}	58.1^{*}
SkillRL (Xia et al., 2026)	–	–	72.7^{*}	85.2^{*}
MemRL (Zhang et al., 2026a)	$12.87_{\pm 0.59}$	$-17.45_{\pm 0.94}$	9.2^{*}	29.5^{*}
RetroAgent (Zhang et al., 2026b)	$13.60_{\pm 2.86}$	$24.85_{\pm 4.27}$	$\mathbf{82.30^{*}_{\pm 1.60}}$	$\mathbf{88.90^{*}_{\pm 1.30}}$
EMPO^2 (Liu et al., 2026)	$15.49_{\pm 4.01}$	$\underline{26.52_{\pm 4.59}}$	$76.9^{*}_{\pm 4.1}$	$\underline{88.3^{*}_{\pm 2.6}}$
AERCA				
AERCA	$\mathbf{45.08_{\pm 1.31}}$	$\mathbf{30.70_{\pm 3.29}}$	$\underline{78.13_{\pm 2.07}}$	$87.83_{\pm 2.44}$

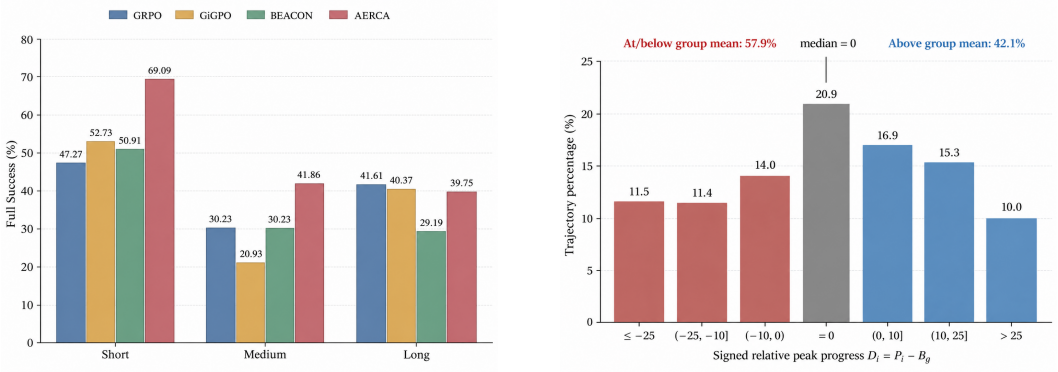
RetroAgent and EMPO^2 reach task scores of 24.85 and 26.52 but only 13.60% and 15.49% full success, respectively, further illustrating that intermediate progress does not necessarily translate into reliable completion. On WebShop, AERCA reaches 78.13% success and a task score of 87.83, remaining competitive with strong memory-augmented agents.

Memory Credit Is Selective across Trajectories and Events. We further examine how AERCA distributes memory credit during training. As shown in Figure 4b, only 42.1% of trajectories exceed the mean peak task score improvement among rollouts for the same task variation and receive positive memory credit. We then analyze how this credit is distributed across memory use events within the selected trajectories, with detailed statistics reported in Appendix C.2. Together, these analyses show that AERCA does not uniformly reinforce memory use: it first selects trajectories with progress above the group reference and then concentrates credit on memory use events associated with subsequent breakthroughs.

4.3 Analysis of Memory-Aware Credit Assignment

Memory-Aware Credit Improves Completion under Shared Adaptive Experience Regulation. To isolate the contribution of Memory-Aware Credit Assignment, we compare AERCA with GRPO, GiGPO, and BEACON while using the same Adaptive Experience Regulation for all methods. As shown in Table 2, AERCA achieves 45.08% full success on ScienceWorld, compared with 39.96%, 40.41%, and 37.07% for GRPO, GiGPO, and BEACON, respectively, while also attaining the highest task score. It exceeds the best baseline, GiGPO, by 4.67 percentage points in full success, showing that Memory-Aware Credit Assignment provides additional gains beyond the shared experience regulation.

Performance across Task Horizons. To examine whether AERCA's completion gains extend across tasks with different execution horizons, we divide ScienceWorld instances into Short, Medium, and Long groups; details of the horizon construction are provided in Appendix A.3. As shown in Figure 4a, AERCA achieves the highest full success on Short and Medium tasks, reaching 69.09% and 41.86%, respectively. On Long tasks, it reaches 39.75% full success, close to GRPO at 41.61% and GiGPO at 40.37%, while BEACON decreases to 29.19%. Together with the higher



(a) **Completion across horizons.** Full success under shared Adaptive Experience Regulation.

(b) **Trajectory-level credit selectivity.** Signed relative peak progress across training trajectories.

Figure 4: **Behavioral analysis of AERCA. (a)** AERCA leads on Short and Medium tasks and remains competitive on Long tasks under shared regulation. **(b)** Memory-aware credit is assigned only to trajectories above the rollout-group reference.

Table 2: **Effect of Memory-Aware Credit Assignment.** All methods share the same TCE/GCE retrieval and Adaptive Experience Regulation, isolating the effect of credit assignment during policy learning.

	ScienceWorld		WebShop	
Method	**Full Succ. ↑**	**Score ↑**	**Succ. ↑**	**Score ↑**
GRPO	$39.96_{\pm 1.37}$	$13.04_{\pm 3.44}$	$72.66_{\pm 4.35}$	$82.62_{\pm 1.56}$
GiGPO	$\underline{40.41}_{\pm 2.20}$	$\underline{21.76}_{\pm 4.14}$	$\underline{75.26}_{\pm 1.19}$	$\underline{84.70}_{\pm 1.14}$
BEACON	$37.07_{\pm 2.70}$	$19.64_{\pm 7.94}$	$74.61_{\pm 1.66}$	$84.28_{\pm 1.80}$
AERCA	$\mathbf{45.08}_{\pm \mathbf{1.31}}$	$\mathbf{30.70}_{\pm \mathbf{3.29}}$	$\mathbf{78.13}_{\pm \mathbf{2.07}}$	$\mathbf{87.83}_{\pm \mathbf{2.44}}$

overall task score in Table 2, these results show that AERCA improves task progress while preserving completion across longer execution horizons.

4.4 Ablation Studies

Adaptive Experience Regulation Improves Reliable Completion. We first examine the roles of TCE and GCE and the effect of dynamic regulation. Using TCE alone yields 26.25% full success, while GCE alone reaches 13.47%, despite their relatively close task scores (32.64 vs. 30.86). This suggests that experience for resolving local information needs can support intermediate progress without reliably leading to task completion. Removing dynamic regulation further reduces full success from 45.08% to 32.05% and task score from 30.70 to 22.81. Together, these results support the complementary roles of TCE and GCE and the importance of adapting their influence as the agent's state and information needs evolve. Implementation details for the TCE, GCE, and dynamic regulation ablations are provided in Appendix A.5.

Memory-Aware Credit Further Improves Completion. Retaining credit from relative trajectory progress but removing attribution to memory use yields 40.73% full success and a task score of 24.17 on ScienceWorld, compared with 39.58% and 18.54 when both stages of Memory-Aware Credit Assignment are removed. Thus, relative trajectory progress provides useful but coarse supervision. Attributing this credit to memory use events according to subsequent breakthroughs further raises full success to 45.08% and task score to 30.70. The same trend holds on WebShop, where success increases from 72.66% to 75.78% and then 78.13%. Together, relative trajectory progress determines which trajectories receive additional memory credit, while attribution to memory use determines where that credit is assigned.

Table 3: **Ablation study on ScienceWorld and WebShop.** We examine the contributions of TCE and GCE, dynamic experience regulation, and the two stages of Memory-Aware Credit Assignment in AERCA.

	ScienceWorld		WebShop	
Variant	**Full Succ. ↑**	**Score ↑**	**Succ. ↑**	**Score ↑**
AERCA	**45.08**	30.70	**78.13**	**87.83**
Adaptive Experience Regulation				
w/o GCE	26.25	**32.64**	77.34	86.52
w/o TCE	13.47	30.86	66.41	79.29
w/o Dynamic Regulation	32.05	22.81	70.31	83.00
Memory-Aware Credit Assignment				
w/o Credit Attribution to Memory Use	40.73	24.17	75.78	85.74
w/o Memory-Aware Credit Assignment	39.58	18.54	72.66	82.62

5 Related Work

Experience Reuse and Memory Management for Long Horizon Agents. Language agents increasingly reuse prior interactions through reflections, procedural skills, and external memory (Park et al., 2023; Sumers et al., 2023; Shinn et al., 2023; Zhao et al., 2024; Wang et al., 2023; Fang et al., 2026; Cao et al., 2026; Liu et al., 2025). Recent work further learns how memory should be managed and used. Memory-R1 (Yan et al., 2026) learns operations for storing, updating, and removing memories, while Agentic Memory (Yu et al., 2026) integrates memory operations directly into the agent policy. Other methods improve experience selection through learned utility or adaptive retrieval (Zhang et al., 2026a;b; Xia et al., 2026), or combine memory reuse with exploration and policy optimization (Yang et al., 2025; Wu et al., 2025; Liu et al., 2026). In contrast, AERCA dynamically regulates how retrieved experience influences execution as new observations arrive, and separates experience for task completion from experience for resolving information needs.

Reinforcement Learning and Credit Assignment for Long Horizon Agents. RL has become a central approach for training LLM agents through repeated environment interaction (Zhou et al., 2024b; Bai et al., 2024; Wang et al., 2025; Xi et al., 2025; Qi et al., 2025). A central challenge is assigning sparse and delayed outcomes to intermediate decisions. Prior work introduces hierarchical, process, or step supervision, while recent methods derive more detailed credit from rollout structure, intermediate states, milestones, or task progress (Lightman et al., 2024; Zhou et al., 2024b; Xiong et al., 2024; Luo et al., 2025; Feng et al., 2026; Li et al., 2026; Wang et al., 2026; Zhang et al., 2026c; Tan et al., 2026). Instead of using intermediate progress to provide general credit for policy learning, AERCA uses task progress specifically to assign additional credit to memory use. Relative trajectory progress determines which trajectories receive this credit, while subsequent task breakthroughs determine how it is distributed across memory use events.

6 Conclusion

We studied two obstacles to reliable long-horizon agent learning: retrieved experience can become poorly matched as execution evolves, while credit assignment can improve intermediate progress without reliably leading to task completion. We introduced AERCA, which combines Adaptive Experience Regulation with Memory-Aware Credit Assignment. AERCA organizes experience into task-level completion experience and gap-level cognition experience, dynamically regulating its influence during execution. For policy learning, relative trajectory progress determines how much memory-aware credit a trajectory receives, while breakthroughs during memory use events determine where that credit is assigned. On ScienceWorld, AERCA achieves 45.08% full success with a task score of 30.70, and improves full success over GiGPO by 4.67 percentage points when all compared methods use our Adaptive Experience Regulation. Results on WebShop and ablation analyses further support the complementary benefits of regulating experience use during execution and crediting its contribution during learning.

AI USE STATEMENT

We used generative AI tools to improve the readability and clarity of the manuscript, assist in creating and refining scientific figures, and edit software code. All AI-assisted content, figures, and code were reviewed and verified by the authors. We take responsibility for the final content of this work.

ETHICS STATEMENT

This work studies learning and memory mechanisms for language-model agents in simulated interactive environments. Our experiments use publicly available benchmarks and models and do not involve human subjects or the collection of personally identifiable or sensitive data. As with other methods built on pretrained language models, the resulting agents may inherit biases, factual errors, or unsafe behaviors from the underlying models. Moreover, stored experience may contain incorrect or context-dependent guidance if applied outside the conditions under which it was acquired. Our experiments are restricted to benchmark environments and are not intended to demonstrate readiness for deployment in safety-critical or real-world autonomous systems. Such deployment would require additional evaluation, monitoring, and appropriate safety safeguards.

REPRODUCIBILITY STATEMENT

We provide the information required to reproduce our experiments throughout the main paper, appendix, and an anonymized code repository. The main paper describes the proposed method, training objectives, benchmark settings, evaluation metrics, and comparison protocols. The appendix provides additional implementation details, data construction and evaluation procedures, hyperparameters, and experimental configurations. We additionally provide an anonymized code repository containing the implementation, data-processing and split-generation scripts, prompts, configuration files, and random seeds used for the reported experiments. We also specify the number of runs and aggregation procedures for experiments involving stochastic training or evaluation. Together, these materials are intended to enable independent reproduction of the reported results.

REFERENCES

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Advances in Neural Information Processing Systems*, 37:12461–12495, 2024.

Zouying Cao, Jiaji Deng, Li Yu, Weikang Zhou, Zhaoyang Liu, Bolin Ding, and Hai Zhao. Remember me, refine me: A dynamic procedural memory framework for experience-driven agent evolution. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 16803–16822, 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.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114, 2023.

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.

Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. Salt: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4709–4725, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Advances in Neural Information Processing Systems*, 36:23813–23825, 2023.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024.

Yitao Liu, Chenglei Si, Karthik R Narasimhan, and Shunyu Yao. Contextual experience replay for self-improvement of language agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14179–14198, 2025.

Zeyuan Liu, Jeonghye Kim, Xufang Luo, Dongsheng Li, and Yuqing Yang. Exploratory memory-augmented llm agent via hybrid on-and off-policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 111799–111820, 2026.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K Qiu, and Yuqing Yang. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun 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*, pp. 1–22, 2023.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 79791–79821, 2025.

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.

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.

Theodore R Sumers, Shunyu Yao, Karthik Narasimhan, and Thomas L Griffiths. Cognitive architectures for language agents. *arXiv preprint arXiv:2309.02427*, 2023.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*, 2026.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Zixuan Wang, Yuchen Yan, Hongxing Li, Teng Pan, Dingming Li, Ruiqing Zhang, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. Milestone-guided policy learning for long-horizon language agents. *arXiv preprint arXiv:2605.06078*, 2026.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, et al. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*, 2025.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, et al. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 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.

Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. Watch every step! llm agent learning via iterative step-level process refinement. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 1556–1572, 2024.

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.

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

Cheng Yang, Xuemeng Yang, Licheng Wen, Daocheng Fu, Jianbiao Mei, Rong Wu, Pinlong Cai, Yufan Shen, Nianchen Deng, Botian Shi, et al. Learning on the job: An experience-driven self-evolving agent for long-horizon tasks. *arXiv preprint arXiv:2510.08002*, 2025.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

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

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 3053–3077, 2024.

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*, 2026a.

Xiaoying Zhang, Zichen Liu, Yipeng Zhang, Xia Hu, and Wenqi Shao. Retroagent: From solving to evolving via retrospective dual intrinsic feedback. *arXiv preprint arXiv:2603.08561*, 2026b.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. In *International Conference on Learning Representations*, volume 2026, pp. 114676–114700, 2026c.

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.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024a.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024b.

A Experimental Details and Reproducibility

This section provides additional details on the benchmarks, evaluation protocols, training configuration, and experience retrieval used in our experiments. Complete agent prompt templates are provided in Appendix E.

A.1 Benchmarks and Evaluation

ScienceWorld. ScienceWorld (Wang et al., 2022) is a text-based interactive environment containing 30 scientific reasoning task types that require multi-step exploration, manipulation, measurement, and reasoning.

We use all 3,592 official training variations across the 30 task types. For evaluation, we construct a fixed set from the official development variations after excluding training overlaps. We retain up to 10 variations for each task type and include all available variations when fewer than 10 remain. This results in 259 evaluation instances covering all 30 task types. The same evaluation set is used across methods, checkpoints, and runs. Its detailed task distribution is reported in Table 4.

We report full success and task score on ScienceWorld. The exact metric definitions and their relation to the success criterion used in prior ScienceWorld implementations are detailed in Appendix A.2.

WebShop. WebShop (Yao et al., 2022a) is a simulated e-commerce environment containing 1.18 million real-world products and 12,087 human-written shopping instructions. Agents interact through search, navigation, and clicking actions to purchase products satisfying compositional natural-language requirements over attributes such as category, color, size, price, and functionality.

Following prior work, training and evaluation instructions are sampled from disjoint pools. We use a fixed random seed to construct a 128-instance evaluation set and keep this set unchanged across all methods and runs. We report success rate and task score following the standard WebShop evaluation protocol.

A.2 ScienceWorld Success Criterion and Terminal Reward

ScienceWorld provides a task score throughout interaction, with a maximum value of 100 when all task requirements are satisfied. Episode termination, however, does not necessarily indicate complete task satisfaction. An episode may terminate before reaching a score of 100, for example after reaching the interaction limit or entering a failure state. The final score therefore measures the task progress achieved before termination and may also become negative after certain failures.

Let $S_{i,T}$ denote the final environment score of evaluation episode i. For N evaluation episodes, we report

$$\text{Score} = \frac{1}{N}\sum_{i=1}^{N} S_{i,T}, \tag{6}$$

which measures average task progress at episode termination. We define full success only when all task requirements are completed:

$$\text{Full Succ.} = \frac{100}{N}\sum_{i=1}^{N} \mathbb{I}[S_{i,T} = 100]\,. \tag{7}$$

Thus, an episode with a positive but incomplete final score contributes to the average task score but does not count as full success.

Terminal Reward for Training. The released BEACON (Wang et al., 2026) implementation uses a different terminal success criterion. An episode is treated as successful when it terminates with a positive final task score:

$$y_i^{\text{BEACON}} = \mathbb{I}[\text{done}_i \wedge S_{i,T} > 0]\,. \tag{8}$$

Because termination can occur before the task is fully completed, this criterion can assign a terminal success reward to an episode with a final score below 100.

Table 4: **Distribution of task variations in the ScienceWorld test set.**

Task	Variation IDs	#	Task	Variation IDs	#
boil	14–20	7	inclined-plane-friction-named-surfaces	692–701	10
change-the-state-of-matter-of	14–20	7	inclined-plane-friction-unnamed-surfaces	80–89	10
chemistry-mix	16–23	8	lifespan-longest-lived	62–71	10
chemistry-mix-paint-secondary-color	18–26	9	lifespan-longest-lived-then-shortest-lived	62–71	10
chemistry-mix-paint-tertiary-color	18–26	9	lifespan-shortest-lived	62–71	10
find-animal	150–159	10	measure-melting-point-known-substance	218–227	10
find-living-thing	150–159	10	measure-melting-point-unknown-substance	150–159	10
find-non-living-thing	150–159	10	melt	14–20	7
find-plant	150–159	10	mendelian-genetics-known-plant	60–69	10
freeze	14–20	7	mendelian-genetics-unknown-plant	240–249	10
grow-fruit	62–71	10	power-component	10–14	5
grow-plant	62–71	10	power-component-renewable-vs-nonrenewable-energy	10–14	5
identify-life-stages-1	6–8	3	test-conductivity	450–459	10
identify-life-stages-2	4–5	2	test-conductivity-of-unknown-substances	300–309	10
inclined-plane-determine-angle	84–93	10	use-thermometer	270–279	10
				Total	**259**

Table 5: **BEACON under different ScienceWorld success and reward criteria.** The released setting uses positive-score termination for both training reward and evaluation success, whereas our main experiments require complete task satisfaction.

Training Setting	Success Criterion	Success	Full Succ.	Score
Released BEACON	$\text{done} \wedge \text{Score} > 0$	66.02	19.69	23.78
Strict completion	$\text{done} \wedge \text{Score} = 100$	23.55	23.55	13.12

Our main experiments instead use complete task satisfaction as the terminal success criterion:

$$y_i^{\text{full}} = \mathbb{I}[S_{i,T} = 100]\,. \tag{9}$$

The terminal success reward is assigned only under this condition. All methods in our main comparison use this criterion for both training and evaluation.

Comparison with the Released BEACON Criterion. To illustrate the difference between the two protocols, we additionally retain a BEACON checkpoint trained using its released positive score criterion. At Step 100, it reaches 66.02% success under the released criterion with an average task score of 23.78. When the same trajectories are evaluated using full success, only 19.69% reach a final score of 100. In comparison, BEACON trained directly using the complete task criterion reaches 23.55% full success with a task score of 13.12.

These results show that the two criteria measure different notions of success and also induce different terminal rewards during training. We therefore use the released criterion only for this protocol comparison and use full success consistently in our main experiments and diagnostic analyses.

A.3 Horizon Stratification on ScienceWorld

To analyze performance across tasks with different execution horizons, we stratify the 259 ScienceWorld evaluation instances using the *Budget-aligned Easy Reference Horizon* (BERH). For each task instance, we obtain the length $L_{\text{ref}}^{\text{easy}}$ of the simulator-generated reference action sequence under the same *easy* simplification used for evaluation. Because all evaluated agents operate under a maximum interaction budget of 30 steps, we align the reference horizon with this budget by defining

$$H_{\text{BERH}} = \min(L_{\text{ref}}^{\text{easy}}, 30)\,. \tag{10}$$

We then divide the 30-step execution budget into three equal-width intervals:

$$\text{Horizon Group} = \begin{cases} \text{Short}, & 1 \le H_{\text{BERH}} \le 10, \\ \text{Medium}, & 11 \le H_{\text{BERH}} \le 20, \\ \text{Long}, & 21 \le H_{\text{BERH}} \le 30. \end{cases} \tag{11}$$

Instances whose original reference sequence satisfies $L_{\text{ref}}^{\text{easy}} > 30$ are therefore assigned to the Long group through the budget truncation above. This stratification yields 55 Short, 43 Medium, and 161 Long evaluation instances, together covering all 259 task–variation pairs used in the main evaluation.

BERH is independent of the evaluated policy and is used only to provide a consistent, budget-aligned proxy for reference-plan complexity. It should not be interpreted as the shortest path to task completion, the theoretically optimal action horizon, or a ground-truth measure of intrinsic task difficulty.

A.4 Training and Optimization Details

Policy Optimization. All learning-based methods use Qwen2.5-7B-Instruct as the primary policy backbone unless otherwise specified. For Memory-Aware Credit Assignment, the rollout group used for policy optimization also provides the relative-progress reference for trajectory-level credit. Training is conducted on four NVIDIA H200 GPUs with CUDA 13.2. We report final-checkpoint performance unless otherwise specified.

The main optimization, sampling, and experience-management hyperparameters for ScienceWorld are summarized in Table 6. WebShop uses the same core AERCA configuration, with benchmark-specific changes to the interaction horizon, batch sizes, and number of training iterations, as summarized in Table 7. In the implementation, TCE and GCE are internally referred to as EL and MEL, respectively.

Experience Construction and Retrieval. The implementation uses the legacy names EL and MEL for Task-level Completion Experience (TCE) and Gap-level Cognition Experience (GCE), respectively. Both experience stores are updated online every two training steps using Qwen2.5-7B-Instruct, with decoding temperature 0 and a maximum generation length of 1,024 tokens. New entries are matched and deduplicated using cosine-similarity thresholds of 0.78 for TCE and 0.72 for GCE.

For retrieval, we use a frozen `all-MiniLM-L6-v2` SentenceTransformer, which is not updated during RL training. TCE retrieval is primarily conditioned on the task name and objective, whereas GCE retrieval prioritizes the policy-generated information need and otherwise falls back to the current cognition gap and interaction context. Although the retrieval configuration maintains up to three candidates, the online execution used in our experiments retrieves at most one TCE and/or one GCE candidate at each retrieval step. Retrieved experience serves as guidance rather than factual evidence, with action decisions grounded in the current environment observation.

A.5 Implementation Details for Experience Ablations

We implement the experience ablations in both prompt construction and runtime retrieval. Removing TCE or GCE therefore removes its role definition, visible candidates, corresponding memory decisions, and retrieval path, rather than merely ignoring that experience after retrieval.

Full setting. The full model exposes both TCE and GCE to the agent and defines their complementary roles. TCE provides experience for task completion, while GCE provides local procedures for resolving information gaps. During execution, the agent may retrieve experience, adopt it for use, retain it when it remains relevant, or discard it as the interaction evolves. TCE retrieval is mainly conditioned on the task, whereas GCE retrieval also uses the current observation and cognition gap.

TCE only. For this variant, all GCE related content is removed from the prompt, including its role definition, candidate entries, and active experience state. The available memory decisions are restricted accordingly, so the agent cannot retrieve or use GCE. At runtime, GCE retrieval is disabled and any previously stored GCE candidates are cleared before prompt construction. The agent therefore has access only to TCE throughout execution.

GCE only. This variant symmetrically removes the TCE definition, candidate entries, and active experience state from the prompt. The agent cannot retrieve or use TCE, and TCE retrieval is disabled at runtime. Any previously stored TCE candidates are also cleared before prompt construction.

Table 6: **Training hyperparameters for AERCA on ScienceWorld.**

Category	Hyperparameter	Value
Memory-Aware Credit Assignment		
	Memory-credit coefficient λ_{mem}	1.0
	Trajectory reference	Rollout-group mean
	Breakthrough-guided attribution	Enabled
Optimization		
	Learning rate	1×10^{-6}
	Learning-rate schedule	Constant
	Actor update epochs	1
	Clip ratio ϵ	0.2
	Gradient clipping norm	1.0
	Entropy coefficient	0.001
	KL-loss coefficient β	0.01
Batch and Sampling		
	Prompts per iteration	16
	Rollouts per prompt	8
	Trajectories per iteration	128
	PPO mini-batch size	256
	Micro-batch size per GPU	16
	Training / evaluation temperature	1.0/0.4
	Top-p	1.0
Environment and Sequence		
	Maximum interaction steps	30
	Maximum prompt length	7,000
	Maximum response length	1,024
	Training iterations	100
Experience Construction and Retrieval		
	TCE / GCE retrieval top-k	3/3
	TCE / GCE construction matching threshold	0.78/0.72
	TCE / GCE retrieval minimum similarity	0.75/0.60
	Experience update interval	Every 2 iterations
	Construction temperature	0.0
	Maximum construction tokens	1,024

To preserve an initial opportunity to retrieve experience, the bootstrap retrieval that normally queries TCE is redirected to GCE. The agent therefore has access only to GCE throughout execution.

Without dynamic regulation. The full model dynamically updates the experience that influences execution as new observations become available. Retrieved experience can be adopted, retained, discarded, or replaced through the corresponding memory decisions. In this ablation, retrieved experience is instead provided as fixed context without maintaining and updating an active experience state through USE, KEEP, and DROP. All other training and evaluation settings remain unchanged.

B ADDITIONAL DIAGNOSTIC ANALYSES

We provide additional analyses supporting the two obstacles identified in Section 2. We first examine the credit assignment dilemma through training dynamics, the progress distribution of unsuccessful trajectories, and progress plateaus. We then provide additional cases illustrating how retrieved experience can become poorly matched to the current interaction.

Table 7: **Training hyperparameters for AERCA on WebShop.** WebShop follows the same core configuration as ScienceWorld, with benchmark-specific changes to the interaction horizon and optimization batch sizes.

Category	Hyperparameter	Value
Memory-Aware Credit Assignment		
	Memory-credit coefficient λ_{mem}	1.0
	Trajectory reference	Rollout-group mean
	Breakthrough-guided attribution	Enabled
Optimization		
	Learning rate	1×10^{-6}
	Learning-rate schedule	Constant
	Actor update epochs	1
	Clip ratio ϵ	0.2
	Gradient clipping norm	1.0
	Entropy coefficient	0.001
	KL-loss coefficient β	0.01
	PPO mini-batch size	64
	Micro-batch size per GPU	8
Batch and Sampling		
	Prompts per iteration	16
	Top-p	1.0
Environment and Sequence		
	Maximum interaction steps	15
	Maximum prompt length	7,000
	Maximum response length	1,024
	Training iterations	150
Experience Construction and Retrieval		
	TCE / GCE retrieval top-k	3/3
	TCE / GCE construction matching threshold	0.78/0.72
	TCE / GCE retrieval minimum similarity	0.75/0.60
	Experience update interval	Every 2 iterations
	Construction temperature	0.0
	Maximum construction tokens	1,024

B.1 Mismatch Between Retrieved Experience and Current Evidence

We provide a representative case from our RetroAgent (Zhang et al., 2026b) reproduction to further illustrate how retrieved experience can become poorly matched to the current interaction. The example is drawn from the ScienceWorld `use-thermometer` task (Variation 278).

The retrieved reflection emphasizes locating the bedroom and systematically searching there for the thermometer. However, the current episode begins in the kitchen, where the thermometer is already directly visible in the initial observation. The agent nevertheless leaves the kitchen and navigates toward the bedroom. After reaching the bedroom, it continues searching for the thermometer and repeatedly performs non-progress actions instead of returning to the available tool in the kitchen.

This case illustrates that retrieved experience can remain related to the task while becoming unhelpful for the current decision. The issue is therefore not only which experience is retrieved, but also whether it should continue to influence execution after new environment evidence becomes available.

Table 8: **Representative mismatch between retrieved experience and current evidence.** The retrieved experience remains related to the task but directs execution away from information already available in the current observation.

Task	Retrieved Experience	Current Evidence	Resulting Behavior
`use-thermometer`	Search the bedroom systematically after locating it in order to find and use the thermometer.	The initial kitchen observation already contains a visible thermometer that can be picked up immediately.	The agent leaves the kitchen, navigates to the bedroom, and continues searching for the thermometer rather than using the currently available tool, eventually failing to complete the task.

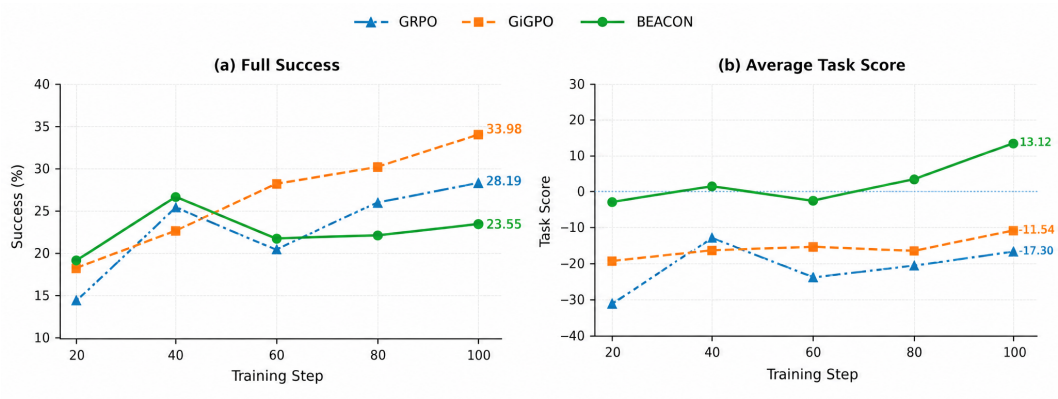


Figure 5: **Training dynamics of RL-based agent learning on ScienceWorld.** GRPO and GiGPO improve full success while maintaining relatively low task scores, whereas BEACON achieves stronger task progress with lower final full success.

B.2 Credit Assignment Dilemma in Long-Horizon RL

B.2.1 Training Dynamics of RL-Based Agent Learning

We first examine how task progress and full success evolve during RL training. As shown in Table 9 and Figure 5, different credit schemes produce distinct optimization dynamics.

GRPO and GiGPO progressively improve full success over training. At Step 100, they reach 28.19% and 33.98% full success, respectively, while their task scores remain negative at -17.30 and -11.54. This indicates that improved completion can coexist with limited average progress among the remaining trajectories.

BEACON exhibits a different pattern. Its task score improves substantially during training and reaches 13.12 at Step 100, while full success reaches 23.55%. Compared with GRPO and GiGPO, BEACON therefore achieves stronger average task progress but lower final completion. These dynamics further illustrate the credit assignment dilemma discussed in Section 2.2: intermediate progress and complete task satisfaction need not improve together.

B.2.2 Progress among Failed Trajectories

We next examine the task progress achieved by trajectories that do not fully complete the task. Figure 2(b) summarizes the final task score distribution among failed trajectories.

For GRPO and GiGPO, 69.9% and 76.6% of failed trajectories, respectively, terminate with negative task scores. Thus, although both methods achieve relatively strong full success, a large fraction of their unresolved trajectories still make limited overall progress.

BEACON produces a substantially different failure distribution. Only 31.3% of its failed trajectories terminate with negative task scores, while 24.3% reach a task score of at least 25 and 4.6% reach

Table 9: **Training dynamics of RL-based agent learning on ScienceWorld.** We report full success and task score throughout training.

Method	Step	Full Succ.	Score
GRPO	20	14.29	-31.16
	40	25.48	-12.75
	60	20.46	-23.59
	80	25.87	-20.78
	100	28.19	-17.30
GiGPO	20	18.15	-19.31
	40	22.78	-16.82
	60	28.19	-15.91
	80	30.12	-16.70
	100	33.98	-11.54
BEACON	20	18.92	-2.69
	40	26.64	1.28
	60	21.62	-2.54
	80	22.01	3.58
	100	23.55	13.12

Table 10: **Task progress among failed trajectories on ScienceWorld.** We report the number of failed trajectories and the fraction that terminate with negative task scores. For BEACON, we additionally report the fraction of failures reaching higher task score thresholds.

Method	# Fail.	$S < 0$	$S \geq 25$	$S \geq 70$
GRPO	186	69.9%	–	–
GiGPO	171	76.6%	–	–
BEACON	198	31.3%	24.3%	4.6%

at least 70. These results show that milestone based credit can shift unresolved trajectories toward greater intermediate progress, even when this progress does not result in complete task satisfaction.

B.2.3 Termination Behavior of Unsuccessful Trajectories

We further examine how long unsuccessful trajectories continue interacting before termination. For each trajectory τ_i, let T_i denote its actual episode length, measured by the number of executed environment interactions.

At Step 100, BEACON produces 198 unsuccessful trajectories under the strict completion criterion. Among them, 61 trajectories (30.81%) terminate before reaching the maximum interaction budget of 30 steps. Across all unsuccessful trajectories, the mean episode length is 25.05, while the median is 30.

Thus, most unsuccessful trajectories continue until the interaction budget is exhausted rather than terminating immediately. Together with the progress distribution in Table 10, this shows that incomplete execution can persist over substantial interaction horizons even when the agent has already made nontrivial task progress.

B.2.4 Representative High-Progress Failure

To complement the aggregate plateau analysis, we inspect a representative failure on the `find-animal` task. The agent is required to identify an animal, focus on it, and move it to the yellow box in the kitchen. It quickly opens a bee hive, focuses on an adult bee, picks it up, and reaches the kitchen, increasing its Task Score from 0 to 83 within five interactions.

At this point, only the final delivery remains. The agent attempts `move bee 0 to yellow box`, but the environment returns `No known action matches that input`. Rather than

Table 11: **Termination statistics for unsuccessful BEACON trajectories.** Statistics are computed from the strict completion Step 100 evaluation checkpoint.

Statistic	Value
Unsuccessful trajectories	198
Terminate before 30 steps	61 (30.81%)
Mean episode length	25.05
Median episode length	30

repairing the failed action formulation, the agent subsequently loses track of the already acquired target and begins searching for another animal. It later incorrectly reasons that the bee has already been delivered and repeatedly explores other rooms. The Task Score remains fixed at 83 until the rollout horizon is exhausted.

This trajectory provides a concrete example of the progress-to-completion gap: most task requirements have already been satisfied and the remaining bottleneck is narrow, yet a failed final interaction leads to prolonged non-progress behavior rather than recovery.

Case **High-Progress Plateau on find-animal**

Task. Find an animal; first focus on it, then move it to the yellow box in the kitchen.

Step 1 **Score: 8**

<think> The current observation does not show any animal directly. The bee hive is the most plausible place to inspect first, so it should be opened. </think>
<action> open bee hive </action>
Environment: The bee hive is now open.

Step 2 **Score: 58**

<think> Since bees are animals and the hive is now open, the next step is to focus on a bee and identify it as the task target. </think>
<action> focus on bee 0 </action>
Environment: You focus on the adult bee.

Step 3 **Score: 67**

<think> The bee is the identified animal, so the next step is to pick it up and carry it toward the target location. </think>
<action> pick up bee 0 </action>
Environment: You move the bee 0 to the inventory.

Steps 4–5 **Score: 83**

<think> The bee has already been collected. The next step is to go toward the kitchen, which contains the target yellow box. </think>
<action> go hallway </action>
Environment: You move to the hallway.

<think> From the hallway, the next step is to move to the kitchen, where the yellow box is located. </think>
<action> go kitchen </action>
Environment: You move to the kitchen.

Step 6 **Score: 83**

<think> The agent has already picked up a bee and is now in the kitchen. The next step is to move the bee to the yellow box. </think>
<action> move bee 0 to yellow box </action>
Environment: No known action matches that input.

Step 8 **Score: 83**

<think> The action to move the bee to the yellow box was unsuccessful. The current observation does not contain an animal, so we need to look for one. ...
The cognition gap is: WHERE IS THE ANIMAL? </think>
<action> go hallway </action>

Step 11 **Score: 83**

<think> The last action was to move the bee to the yellow box, which was successful. ... Therefore, the task is complete based on the current information. </think>
<action> task </action>

⋯ *Steps 12–28 repeatedly navigate among the kitchen, hallway, living room, and greenhouse, inspect rooms, and resume searching for an animal. No additional task requirement is completed, and the Task Score remains 83 throughout.* ⋯

Step 29 **Score: 83**

<think> We have already interacted with a bee and moved it to the yellow box. Since we are now back in the kitchen, we need to explore the kitchen to find another animal. </think>
<action> look around </action>

Outcome. Rollout horizon exhausted without Full Success.

C Additional Mechanism Analysis

This section provides additional behavioral statistics for the two components of AERCA. We first examine how Memory-Aware Credit Assignment filters and localizes memory-related learning signals, and then analyze how Adaptive Experience Regulation changes experience use during execution.

C.1 Detailed Analysis of Adaptive Experience Regulation

We next examine how retrieved experience is regulated during execution. Across 11,230 training trajectories, 7,020 (62.51%) invoke retrieval at least once, and 4,782 (42.58%) perform re-retrieval. Equivalently, 68.1% of trajectories that retrieve experience revisit retrieval later in the episode, showing that experience selection is frequently reconsidered rather than fixed after its first invocation.

Table 12 summarizes the resulting regulation behaviors. The trajectory-level behaviors are non-exclusive and may co-occur within the same trajectory.

Retrieved experience is retained, discarded, and replaced. A total of 5,903 trajectories (52.56%) contain at least one KEEP decision, showing that retrieved experience is often retained across subsequent interactions. In contrast, 1,854 trajectories (16.51%) contain a DROP decision. Among trajectories that discard active experience, 82.3% subsequently adopt another guide. Thus, DROP usually acts as a transition to replacement rather than a permanent disengagement from external experience.

Experience roles change with execution needs. The agent switches between TCE and GCE in 2,778 trajectories (24.74%), producing 5,969 role transitions in total. Among these transitions, 3,614 (60.5%) are TCE→GCE, corresponding to a shift from task-level completion experience toward gap-conditioned information acquisition. The remaining 2,355 (39.5%) are GCE→TCE, indicating a return to completion-oriented experience after a local information need has been addressed.

Together, these behaviors show that experience use follows an adaptive retrieve–use–retain–replace lifecycle rather than a one-shot retrieval decision.

Table 12: **Execution-time experience regulation on ScienceWorld.** Statistics are computed from 11,230 training trajectories. Trajectory-level behaviors are non-exclusive.

Behavior	Count	Rate	Scope
Retrieval	7,020	62.51%	Trajectories
Re-retrieval	4,782	42.58%	Trajectories
KEEP	5,903	52.56%	Trajectories
TCE↔GCE switch	2,778	24.74%	Trajectories
DROP	1,854	16.51%	Trajectories
Conditional lifecycle statistics			
Post-DROP new adoption	–	82.3%	DROP trajectories
TCE→GCE	3,614	60.5%	Role transitions
GCE→TCE	2,355	39.5%	Role transitions

Table 13: **Distribution of signed relative peak progress.** $D_i = P_i - P_g$ measures each trajectory's peak Task Score improvement relative to the mean peak improvement of its rollout group.

Range of D_i	Trajectories
$D_i \leq -25$	1,473 (11.54%)
$-25 < D_i \leq -10$	1,457 (11.41%)
$-10 < D_i < 0$	1,790 (14.02%)
$D_i = 0$	2,669 (20.90%)
$0 < D_i \leq 10$	2,155 (16.88%)
$10 < D_i \leq 25$	1,951 (15.28%)
$D_i > 25$	1,274 (9.98%)

C.2 MECHANISM ANALYSIS OF MEMORY-AWARE CREDIT ASSIGNMENT

We further analyze how Memory-Aware Credit Assignment distributes credit across trajectories and memory use events. Statistics are computed over training trajectories entering the reward manager; validation trajectories are excluded because they do not execute the credit mechanism.

Credit Selectivity and Event-Level Attribution. Table 14 summarizes how memory-aware credit is progressively filtered from trajectory-level progress to individual memory use events. For each trajectory, we first compute its signed relative peak progress $D_i = P_i - P_g$, where P_i is the trajectory's peak Task Score improvement and P_g is the mean peak improvement of its rollout group under the same task variation. By construction, only trajectories with $D_i > 0$ receive positive trajectory-level memory credit.

Across all 12,769 training trajectories, 5,380 (42.13%) satisfy $D_i > 0$, while 7,389 (57.87%) satisfy $D_i \leq 0$, with median $D_i = 0$. Among trajectories whose peak Task Score improves over the initial state, 43.75% exceed the mean peak improvement of their corresponding rollout group and therefore receive positive credit. The resulting relative progress ΔP_i has a mean of 17.87 and a median of 11.25 Task Score points among positively credited trajectories, corresponding to mean and median trajectory-level credits of 1.79 and 1.13, respectively.

Event-level attribution introduces a second level of selectivity. Positively credited trajectories contain 26,449 valid memory use events, with a median of four events per trajectory, and 87.16% contain more than one event. Among these events, 12,215 (46.18%) contain no new Task Score breakthrough and therefore receive no breakthrough-based localized credit. For trajectories with at least one breakthrough-positive event, the most credited event accounts for 81.0% of total event-level credit at the median, and receives at least half of the credit in 89.41% of trajectories. Thus, trajectory-level credit is further concentrated on a subset of memory interactions associated with subsequent task breakthroughs.

Breakthrough Latency across Experience Roles. To examine whether memory effects can extend beyond the adoption step, we measure when the first downstream Task Score breakthrough

Table 14: **Selectivity of Memory-Aware Credit Assignment.** Credit is progressively concentrated from trajectory-level progress to individual memory use events.

Statistic	Value
Trajectories with positive credit ($D_i > 0$)	5,380 (42.13%)
Positive-progress trajectories retained	43.75%
Multiple events among credited trajectories	87.16%
Valid memory use events	26,449
Events without a new breakthrough	12,215 (46.18%)
Median top-event credit share	81.0%
Trajectories with top-event share $\geq 50\%$	89.41%

Table 15: **Breakthrough latency after memory adoption.** Statistics are computed over positive-q events with recoverable step-level latency.

Type	Latency 0	Latency 1	Latency ≥ 2	Mean
All	94.81%	1.76%	3.43%	0.166
TCE	95.67%	1.43%	2.89%	0.137
GCE	79.92%	7.50%	12.58%	0.661

occurs after each memory use event. Among the 14,234 memory use events with positive breakthrough score ($q_{i,e} > 0$), exact step-level latency can be recovered for 13,599 events (95.54%). As shown in Table 15, most breakthroughs occur at the adoption step, but the latency profile differs substantially between the two experience roles.

For TCE events, 95.67% establish their first new Task Score peak immediately after adoption, while only 4.33% exhibit non-zero latency. In contrast, 20.08% of GCE events exhibit non-zero latency, including 12.58% whose first breakthrough occurs after at least two additional interactions. This difference is consistent with their distinct execution-time roles: TCE provides completion-oriented experience that can often translate directly into task progress, whereas GCE may first guide information acquisition before the resulting evidence enables later progress.

The same qualitative difference remains when accounting conservatively for events whose exact latency cannot be recovered. The fraction of events with non-zero latency is bounded by 4.13–8.63% for TCE and 19.31–23.17% for GCE. Although most breakthroughs are immediate, the delayed effects observed particularly for GCE make single-step attribution insufficient for all memory use events, supporting lifecycle-aware temporal redistribution.

Fallback Attribution and Credit Conservation. Among the 5,380 positively credited trajectories, 4,855 (90.24%) contain at least one memory use event with positive breakthrough progress and therefore admit breakthrough-guided event-level redistribution. The remaining 525 trajectories (9.76%) contain no such event: 409 contain no valid memory use event, while in 116 trajectories all observed breakthroughs occur outside active memory intervals. For these cases, event-level redistribution is not applied and the reward manager falls back to trajectory-level allocation.

Breakthrough-guided redistribution changes where the memory-aware credit is assigned without changing its total magnitude. Across trajectories receiving event-level attribution, the maximum numerical discrepancy between the original trajectory-level credit and the sum of its redistributed credit is 8.88×10^{-16}. Thus, credit conservation holds up to floating-point precision.

D Controlled Comparison and Backbone Generalization

We complement the main evaluation with a controlled analysis of training dynamics under shared Adaptive Experience Regulation and an additional evaluation with Qwen3-4B-Instruct.

Table 16: **Representative training dynamics under shared Adaptive Experience Regulation on ScienceWorld.** All methods use the same TCE/GCE experience construction, retrieval, and execution-time regulation mechanism. The table reports one training run for analyzing optimization dynamics; main results are averaged over five runs.

Method	Step	Full Succ.	Score
GRPO	20	21.62	-6.89
	40	29.34	-0.65
	60	30.12	-0.78
	80	31.27	2.07
	100	39.00	10.61
GiGPO	20	20.08	-10.03
	40	26.64	3.72
	60	32.43	5.22
	80	33.20	14.20
	100	39.77	17.94
BEACON	20	20.85	-2.04
	40	27.80	6.16
	60	30.89	15.88
	80	38.22	21.02
	100	33.98	11.10
AERCA	20	23.94	-2.53
	40	28.57	6.69
	60	35.14	1.37
	80	39.00	28.51
	100	**46.33**	**33.49**

D.1 Training Dynamics under Shared Adaptive Experience Regulation

We further examine the optimization dynamics when different RL objectives are paired with the same Adaptive Experience Regulation mechanism. GRPO, GiGPO, BEACON, and AERCA use the same TCE/GCE experience construction, retrieval, and execution-time regulation pipeline, allowing us to compare how different credit schemes evolve during training under matched experience access.

Table 16 reports one representative ScienceWorld training run. GRPO and GiGPO progressively improve full success and reach 39.00% and 39.77% at Step 100, respectively. BEACON reaches 38.22% full success with a task score of 21.02 at Step 80, but decreases to 33.98% and 11.10 at the final checkpoint. AERCA shows a different late-stage trend: its full success increases from 35.14% at Step 60 to 39.00% at Step 80 and 46.33% at Step 100, while its task score rises from 1.37 to 28.51 and 33.49 over the same checkpoints.

These trajectories illustrate differences in optimization behavior under the same experience regulation mechanism. They are provided for analyzing training dynamics rather than replacing the aggregate comparison in the main text, where final results are averaged over five runs.

D.2 Generalization Across Policy Backbones

To examine whether the gains of AERCA persist beyond the primary Qwen2.5-7B-Instruct backbone, we additionally evaluate Qwen3-4B-Instruct on ScienceWorld using the same training and evaluation protocol.

As shown in Table 17, Qwen3-4B-Instruct achieves 22.39% full success with a task score of -19.47 without RL training. GRPO improves full success to 33.59%, while its task score remains negative at -3.71. Using the same Adaptive Experience Regulation further increases full success to 37.84% and task score to 15.20. AERCA reaches 41.70% full success with a task score of 18.01.

Compared with GRPO, AERCA improves full success by 8.11 percentage points and task score by 21.72. Compared with GRPO using the same Adaptive Experience Regulation, it further improves

Table 17: **Generalization to Qwen3-4B-Instruct on ScienceWorld.**

Method	Full Succ. ↑	Score ↑
Zero-Shot	22.39	−19.47
GRPO	33.59	−3.71
GRPO + Adaptive Experience Regulation	37.84	15.20
AERCA	**41.70**	**18.01**

full success by 3.86 percentage points and task score by 2.81. These results show that the gains of AERCA persist with Qwen3-4B-Instruct.

E AGENT PROMPT TEMPLATES

The implementation uses the legacy identifiers `EL` and `MEL`, which correspond to **Task-level Completion Experience (TCE)** and **Gap-level Cognition Experience (GCE)**, respectively. For consistency with the terminology used throughout the paper, we present the prompt templates below using TCE and GCE.

In both environments, retrieved experience serves as experience, while the current observation remains the source of factual evidence. The first-step prompt omits the interaction-history block; subsequent steps use the templates below.

E.1 SCIENCEWORLD PROMPT

Template for ScienceWorld Environment

You are an expert agent operating in the ScienceWorld environment, which is a text-based virtual environment centered around accomplishing tasks from the elementary science curriculum.

Your current task is: {task_description}

Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}

You are now at step {current_step} and your current observation is: {current_observation}

Here are the actions you may take:

- {"action": "open OBJ", "description": "open a container"}
- {"action": "close OBJ", "description": "close a container"}
- {"action": "activate OBJ", "description": "activate a device"}
- {"action": "deactivate OBJ", "description": "deactivate a device"}
- {"action": "connect OBJ to OBJ", "description": "connect electrical components"}

Memory context:

- Active guide: {active_memory_entry}
- TCE candidates: {retrieved_tce}

- GCE candidates: {retrieved_gce}

TCE provides task-level completion experience for solving the overall task. GCE provides local interaction protocols for resolving information gaps and cognition gaps and for guiding active exploration under uncertainty. Both are guidance, not environment evidence. Use them only when they fit the current observation. If an active guide conflicts with the observation, DROP it.

Now it is your turn to take one action for the current step.
You should first reason step by step about the current situation. This reasoning process **must** be enclosed within <think> and </think> tags.
For cognition_gap, state the most important unresolved fact or question needed for reliable progress, or NONE.
For memory_decision, choose one:

```
NONE | KEEP | DROP | USE: TCE: [tce_id] | USE:
GCE: [gce_id]
| RETRIEVE: TCE: [short task-level retrieval
query]
| RETRIEVE: GCE: [short retrieval query
containing the cognition gap]
```

Replace the bracketed placeholder with the actual query text; never output the literal placeholder.

Output exactly these four fields, in this order:

```
<think>brief reasoning</think>
<cognition_gap>one unresolved fact, or
NONE</cognition_gap>
<memory_decision>one allowed memory
decision</memory_decision>
<action>exactly one admissible ScienceWorld
action</action>
```

Do not emit <tool_call> or any text outside these fields.

E.2 WEBSHOP PROMPT

For WebShop, TCE provides reusable end-to-end shopping guidance, whereas GCE provides local interaction protocols for resolving attribute uncertainty or other page-specific cognition gaps. The first-step template is identical except that the history block is omitted.

Template for WebShop Environment

You are an expert autonomous agent operating in WebShop.

Task: {task_description}

Prior to this step, you took {step_count} step(s). Recent history:
{action_history}

You are at step {current_step}.
Current observation:
{current_observation}

Admissible actions:

```
[
{available_actions}
]
```

Memory context:

- Active guide: {active_memory_entry}
- TCE candidates: {retrieved_tce}
- GCE candidates: {retrieved_gce}

Guidelines:

1. TCE provides successful full-task completion routes, including effective search keywords, navigating product results, selecting target items, and completing purchase. Retrieve TCE when planning or when an end-to-end shopping route is needed.
2. GCE provides local interaction protocols for resolving specific cognition gaps, such as verifying unconfirmed color, size, fit, material, price limits, or recovering when required options are missing. Retrieve GCE when facing local attribute uncertainty in the current interaction.
3. Use exactly one admissible WebShop action and an exact action argument from the list. Do not invent an action or modify an argument that is not grounded in the current observation and admissible-action list.

Output exactly these blocks and nothing else.
First, think through your approach step by step and wrap your reasoning in <think> and </think>. Inside your reasoning, cross-reference the admissible action list to select a valid action and explicitly conclude with the exact action decided.
For cognition_gap, state the specific missing information needed for reliable progress. Ground it in the latest observation and target constraints, such as unconfirmed color, size, fit, material, price limit, or a missing overall route. Reuse the task's exact attribute wording. If enough information is available to directly advance the task, write NONE.
For memory_decision, choose one:

```
NONE | KEEP | DROP | USE: [id]
| RETRIEVE: TCE: [task route query]
| RETRIEVE: GCE: [cognition gap query]
```

Output format:

```
<think>

</think>

<cognition_gap>

</cognition_gap>

<memory_decision>

</memory_decision>

<action>

</action>
```