HIRE: HIPPOCAMPAL REPLAY FOR TEST-TIME AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient-free test-time reinforcement learning (RL) adapts large language model (LLM) agents through experience without updating model parameters, avoiding the weight-access requirements and parameter-optimization costs of training-time RL. In this paper, we identify a *discovery–consolidation gap* in test-time RL: agents can discover high-return behavior yet fail to sustain the achieved returns in later rollouts. In existing advantage-based methods, each update reweights the same frozen policy using re-estimated advantages, which can reverse action preferences that previously led to success. Inspired by complementary learning systems theory, we observe that advantage-based adaptation statistically integrates past experience but lacks a complementary hippocampus-like mechanism for recovering successful behavior. We therefore propose **HIRE** (HIppocampal REplay), which uses replay to recover successful behavior, allowing the adapted policy to build on previously achieved progress rather than rediscover it. We prove that HIRE satisfies a return consolidation criterion, yielding an expected-return lower bound that is non-decreasing as the best verified return increases for a fixed exploration rate. Experiments on ScienceWorld and WebChoreArena show that HIRE achieves the best overall performance among the evaluated methods. Our code is available at https://anonymous.4open.science/r/HIRE-D54E.

1 INTRODUCTION

Recent advances in large language models (LLMs) have substantially improved their reasoning, language understanding, and problem-solving capabilities (OpenAI, 2023; Anthropic, 2024; Guo et al., 2025). These advances have enabled LLM agents to pursue goals through planning, tool use, and multi-step interaction with external environments (Yao et al., 2023; Shinn et al., 2023; Wang et al., 2024a). However, effective decision-making in these environments poses several challenges: agents must account for the long-term consequences of their actions, recover from unexpected outcomes, and adapt to new tasks after deployment (Wang et al., 2024b). Reinforcement learning (RL) offers a principled framework for addressing these challenges by optimizing long-term return through trial and error and refining policies as new feedback becomes available. Therefore, agentic RL has become a key paradigm for improving agent capabilities through experience (Zhou et al., 2024b; Qi et al., 2025; Zhang et al., 2026).

Conventional training-time agentic RL collects multi-step trajectories and uses estimates of returns or advantages to update the LLM policy parameters (Zhou et al., 2024b). However, two constraints limit its practical use (Zhang et al., 2026): (i) closed-source models (OpenAI, 2023; Anthropic, 2024) typically provide inference access without permitting weight updates; and (ii) parameter optimization can exceed downstream users' compute and memory budgets (Dettmers et al., 2023). These constraints have motivated early work on gradient-free test-time RL, which adapts policies from experience without updating model parameters. A notable early advance in this direction is JitRL, which adapts the policy through closed-form logit corrections based on nonparametric advantage estimates from past transitions (Li et al., 2026).

However, training-time RL can internalize acquired behavior in policy parameters (Sutton & Barto, 2018; Mnih et al., 2015), whereas test-time RL must preserve improvements through external adaptation. Although JitRL can discover high-return behavior, subsequent policy updates depend on

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: **Core idea.** (a) JitRL may discover a productive path yet lose access to its progress as action preferences change. HIRE instead preserves and replays successful behavior so that subsequent exploration builds on previously achieved progress rather than rediscovering the path. (b) On ScienceWorld's freezing-water task (Wang et al., 2022), JitRL's mean return declines after an early rise, whereas the illustrated HIRE run achieves higher later returns.

advantage estimates that evolve as new experience accumulates. These updates can shift the policy away from actions that previously led to success. Figure 1(b) illustrates this phenomenon on ScienceWorld's freezing-water task, where JitRL's mean return declines after an early rise, indicating that previously attained performance gains are not sustained. We call the gap between discovering high-return behavior and reliably realizing its return in subsequent rollouts the discovery–consolidation gap in test-time RL.

This gap echoes a longstanding question in cognitive science about how newly acquired experience becomes lasting knowledge. Complementary learning systems (CLS) theory explains this consolidation through the interaction of a **neocortical system** that learns shared structure across experiences and a **hippocampal system** that encodes and replays individual episodes to support their integration into cortical knowledge (McClelland et al., 1995; Kumaran et al., 2016). JitRL provides a functional analogue of neocortical integration by aggregating past outcomes, but lacks complementary hippocampus-like replay. Inspired by CLS theory, we propose HIppocampal REplay (HIRE) to address this gap. HIRE preserves successful behavior separately from policy adaptation, using replay to recover previously achieved progress in later rollouts. The adapted policy can then focus on extending that progress rather than rediscovering it, as shown in Figure 1(a).

In summary, our contributions are threefold:

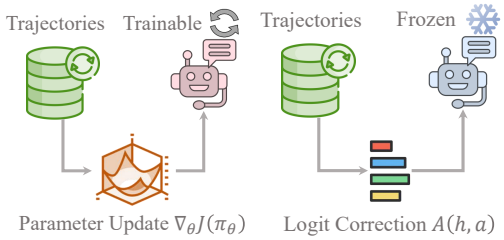
- We identify a *discovery–consolidation gap* in gradient-free test-time RL: discovering high-return behavior does not guarantee that later rollouts sustain the achieved return. We analyze how advantage re-estimation can alter action preferences without preserving the gains of earlier policies.
- We address this gap with HIRE, a CLS-inspired hippocampal replay mechanism that preserves access to successful behavior independently of advantage re-estimation. Through retention, recovery, and exploration, HIRE enables the adapted policy to build on previously achieved progress rather than rediscover it.
- We formalize the preservation of discovered progress through a return consolidation criterion that bounds the expected shortfall below the best verified return. We prove that HIRE satisfies this criterion under explicit recovery and execution conditions, yielding an expected-return lower bound that is non-decreasing as the best verified return increases for a fixed exploration rate.

2 Problem Formulation

Agentic RL. Agentic RL uses reward feedback to improve how an LLM agent performs actions over multi-step interactions with an environment (Zhou et al., 2024b). We model each task as a partially observable Markov decision process (POMDP) (Kaelbling et al., 1998) represented by the tuple $\mathcal{E} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{P}, \Omega, r, \rho_0)$, where $\mathcal{S}$, $\mathcal{O}$, and $\mathcal{A}$ are the state, observation, and action spaces; $\mathcal{P}$

and Ω are the transition and observation models; r is the reward function; and ρ_0 is the initial-state distribution.

At each step t, the agent receives $o_t \sim \Omega(\cdot|s_t)$ and samples $a_t \sim \pi(\cdot|h_t)$ from a history-conditioned LLM policy. Here $h_t = (\iota, o_0, a_0, \ldots, o_t)$ denotes the interaction history augmented with the task instruction ι. The environment then transitions to $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ and yields the reward $r_t = r(s_t, a_t, s_{t+1})$. For an interaction of T steps, the complete trajectory τ records the terminal history h_T together with rewards $(r_t)_{t=0}^{T-1}$. The goal of agentic RL is to maximize expected discounted return $J(\pi) = \mathbb{E}_{\tau\sim p_\pi}[R(\tau)]$, where $R(\tau) = \sum_{t=0}^{T-1} \gamma^t r_t \in [\underline{R}, \overline{R}]$ and $0 < \gamma \leq 1$ is the discount factor. Here $\underline{R}$ and $\overline{R}$ are finite lower and upper return bounds. For a fixed task instruction, p_π is the trajectory distribution induced by policy π in $\mathcal{E}$.



(a) Training-time RL (b) Test-time RL

Figure 2: Two diagrams of agentic RL.

Training-time RL. Conventional training-time agentic RL maximizes $J(\pi_\theta)$ by updating the policy parameters θ through gradient ascent (Figure 2(a)). The policy gradient is

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim p_{\pi_\theta}}\left[R(\tau)\sum_{t=0}^{T-1}\nabla_\theta \log \pi_\theta(a_t|h_t)\right].$$

In practice, proximal policy optimization (PPO) implements policy-gradient updates with a clipped surrogate objective and typically uses a learned value function to estimate advantages (Schulman et al., 2017). Group relative policy optimization (GRPO) retains this clipped objective but estimates advantages from rewards normalized within a group of rollouts for the same input, eliminating the need for a separate value network (Shao et al., 2024).

Test-Time RL. In the gradient-free setting studied here, test-time RL pursues the same objective $J(\pi)$ as training-time RL but adapts the policy without changing the LLM parameters θ (Figure 2(b)). Given past interaction experience $\mathcal{H}$, an adaptation rule $\mathcal{U}$ produces an updated policy π':

$$\pi' = \mathcal{U}(\pi_\theta, \mathcal{H}).$$

The update requires neither access to θ nor computation of $\nabla_\theta J$. It is therefore compatible with inference-only models (OpenAI, 2023) and avoids backpropagation overhead on resource-constrained devices (Liu et al., 2024).

As a representative example, JitRL (Li et al., 2026) produces the updated policy $\pi' = \mathcal{U}_{\text{Jit}}(\pi_\theta, \mathcal{H})$ by reweighting the frozen reference policy:

$$\pi'(a|h) \propto \pi_\theta(a|h)\exp(\beta A(h,a)), \tag{1}$$

where $A(h,a)$ is an advantage estimate computed from past experience in $\mathcal{H}$, and $\beta > 0$ balances advantage-guided adaptation against adherence to the frozen reference policy. Both policies are normalized over the same candidate action set. Writing $\pi_\theta = \text{softmax}(z_\theta)$, this update is algebraically equivalent to an additive logit correction, $z'(h,a) = z_\theta(h,a) + \beta A(h,a)$.

Problem Analysis. Figure 1(b) shows that JitRL's early return gains are not sustained. Whereas training-time RL can encode learned preferences in θ, JitRL relies on $A(h,a)$ to maintain them against a fixed π_θ. For an action a^* from a successful trajectory and a competing action b, (1) gives

$$\pi'(b|h) > \pi'(a^*|h) \iff \beta\big[A(h,b) - A(h,a^*)\big] > \log\frac{\pi_\theta(a^*|h)}{\pi_\theta(b|h)}.$$

The estimated gap $A(h,b) - A(h,a^*)$ can thus outweigh the reference preference for a^*. JitRL estimates A from average historical returns in similar contexts (Li et al., 2026). Limited test-time interaction leaves sparse observations of each (h,a), while differences in context and earlier policies can bias these averages. Errors in the estimated gap can therefore satisfy the condition above even when b is worse. Although estimation error affects both RL settings, JitRL's learned preferences

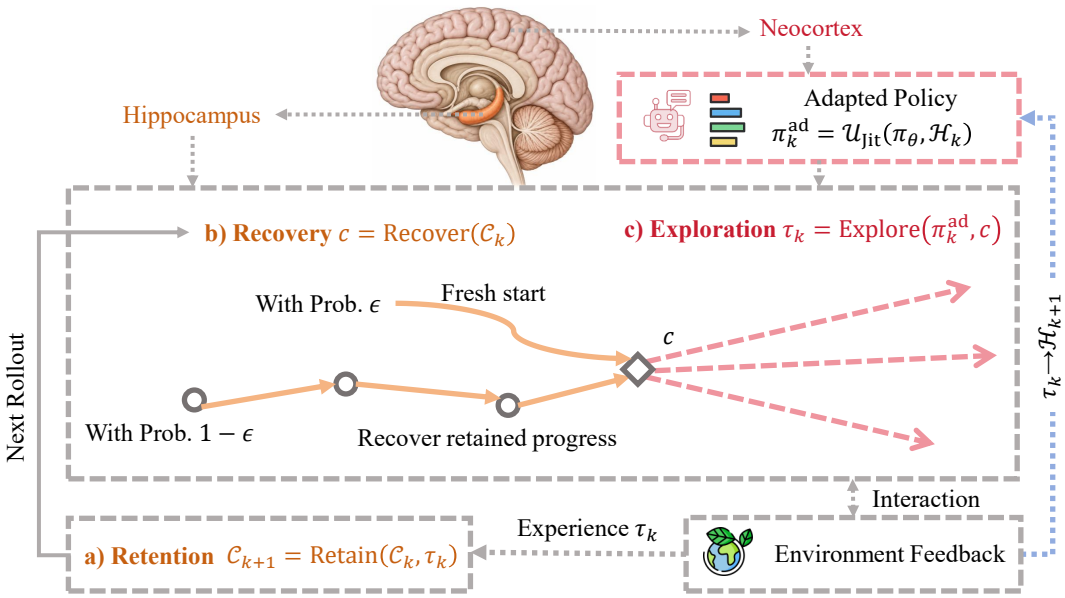


Figure 3: Overview of HIRE and its CLS-inspired design. Retention stores successful behavior, recovery selects a fresh or recovered context, and exploration proceeds with the adapted policy. New experience updates both the adaptation history and the consolidation record for subsequent rollouts.

remain external to the frozen parameters and depend on the evolving advantage. In fact, even with exact advantages, fixed-reference updates can reduce expected return (Appendix A). We therefore seek a consolidation mechanism that preserves discovered gains while supporting further policy improvement.

3 Method

The preceding analysis motivates preserving discovered gains without relying solely on the evolving advantage estimates $A(h, a)$. To this end, we draw on CLS theory, which couples statistical learning with the storage and replay of individual experiences. We translate this inspiration into a return consolidation criterion and establish its connection to the objective $J(\pi)$. This criterion then guides the design of the HIRE method, whose interaction cycle is shown in Figure 3.

3.1 Complementary Learning Systems

CLS theory explains how general knowledge and specific experiences jointly guide behavior through two complementary systems (McClelland et al., 1995). The neocortical system gradually integrates regularities across experiences into structured knowledge that supports generalization. The hippocampal system rapidly stores individual episodes and reactivates them through replay, allowing specific experiences to inform subsequent learning. Replay can prioritize rewarding or otherwise significant episodes, giving rare but valuable experiences greater influence than their frequency alone would suggest (Kumaran et al., 2016).

In our setting, the pretrained LLM supplies general knowledge, while JitRL aggregates past outcomes to estimate advantages and adapt action preferences. Together, these components provide a functional analogue of the neocortical system's knowledge and statistical integration, but lack a separate mechanism for reinstating verified behavior. CLS thus motivates a complementary, hippocampus-inspired mechanism designed to make discovered progress available for later recovery and further exploration. Its design requires a criterion for how well subsequent policies preserve that progress.

3.2 Return Consolidation Criterion

We formalize this requirement as return consolidation, measured by the expected shortfall below a previously verified return. Rather than requiring every update to improve $J(\pi)$, the criterion limits how much verified progress subsequent policies may lose.

At update k, let $\pi_k = \mathcal{U}(\pi_\theta, \mathcal{H}_k)$ be the test-time policy induced by cumulative experience $\mathcal{H}_k$, and let R_k^* be the largest verified achievable return in that experience, or $\underline{R}$ if none is available. With experience held fixed, consolidation bounds the expected shortfall below this level.

Definition 1 (Return consolidation)**.** For $0 \le \delta < 1$, an adaptation rule $\mathcal{U}$ satisfies δ-consolidation if, at every update k, the induced policy π_k satisfies

$$\mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R_k^* - R(\tau_k))_+] \le \delta(R_k^* - \underline{R}), \tag{2}$$

where $(x)_+ := \max\{x, 0\}$.

Definition 1 bounds the expected shortfall below R_k^* by a δ fraction of the verified progress $R_k^* - \underline{R}$, with no loss assigned to returns above R_k^*. Smaller δ requires stronger consolidation; in particular, $\delta = 0$ implies $R(\tau_k) \ge R_k^*$ with probability one. The following proposition connects this criterion to the expected return.

Proposition 1 (Return decomposition)**.** *For every update k,*

$$J(\pi_k) = R_k^* - \mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R_k^* - R(\tau_k))_+] + \mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R(\tau_k) - R_k^*)_+]. \tag{3}$$

The proof is in Appendix B.

In (3), the two expectations measure the shortfall below R_k^* and the gain above it, respectively. Expected return therefore depends on both preserving discovered progress and achieving further improvement. Definition 1 controls the former by bounding the shortfall by $\delta(R_k^* - \underline{R})$, without requiring a positive gain. Since the gain term is nonnegative, this control alone yields the following lower bound on expected return.

Proposition 2 (Return guarantee)**.** *Any policy produced by a δ-consolidating adaptation rule satisfies*

$$J(\pi_k) \ge \underline{R} + (1 - \delta)(R_k^* - \underline{R}). \tag{4}$$

The proof is in Appendix C. Consolidation is therefore a sufficient condition for a return floor, while leaving room for exploration above the verified level.

3.3 Consolidation through Hippocampal Replay

Building on this consolidation criterion, we design a mechanism to preserve verified progress as a starting point for further learning. The mechanism forms a three-stage cycle: *retention* stores the best discovered behavior and useful task observations; *recovery* uses this record to prepare the next rollout's execution context; and *exploration* continues from that context, producing new experience for retention. We maintain a *consolidation record* $\mathcal{C}_k = (\tau_k^\star, R_k^*, I_k)$, containing the highest-return behavior discovered so far, its recorded return R_k^*, and compacted task information I_k. Initially, $\mathcal{C}_0 = (\varnothing, \underline{R}, \varnothing)$, as no behavior or task information has yet been collected. At each update k, the agent begins a new rollout under the task's initial conditions, retaining experience $\mathcal{H}_k$ and the consolidation record $\mathcal{C}_k$ from previous rollouts.

Retention. As discussed in Section 2, discovered return gains need not be sustained as $A(h, a)$ changes. Retention therefore stores successful behavior separately from these changing estimates. For a completed trajectory τ, let $R' = \max_{1 \le j \le T} \sum_{i=0}^{j-1} \gamma^i r_i$ be its highest cumulative return and ℓ the first step in τ at which this value is attained. The candidate *prefix* $\tau' = (h_\ell, (r_i)_{i=0}^{\ell-1})$ contains these first ℓ interaction steps. We maintain compacted task information I_k to carry forward observations and task progress not captured by the saved action sequence. For LLM-based compaction, the same frozen model summarizes the instruction ι, previous information I_k, and the latest interaction trace into task-relevant facts and remaining steps, which are combined with directly recorded data and feedback to obtain I_{k+1}. These updates define the retention function:

$$\mathcal{C}_{k+1} = \mathrm{Retain}(\mathcal{C}_k, \tau) = \begin{cases} (\tau', R', I_{k+1}), & \text{if } R' > R_k^*, \\ (\tau_k^\star, R_k^*, I_{k+1}), & \text{otherwise.} \end{cases}$$

Under this rule, new observations update I_k even when a rollout does not improve return, whereas only a better trajectory replaces the saved behavior. This keeps the best discovered action sequence available independently of advantage re-estimation.

Recovery. Recovery replays retained behavior to reconstruct the context for subsequent decisions. Given $\mathcal{C}_k$, it chooses once per rollout whether to replay the saved prefix with probability $1-\varepsilon$, or to begin at the initial history h_0 with probability ε, termed the *exploration rate*. Without a saved prefix, it begins at h_0. The output is an *execution context* c, comprising the history, accumulated rewards, and compacted task information at the handover step t:

$$c = \mathrm{Recover}(\mathcal{C}_k) = \begin{cases} (h_0, \varnothing, I_k), & \text{with probability } \varepsilon, \\ \big(h_t, (r_i)_{i=0}^{t-1}, I_k\big), & \text{with probability } 1-\varepsilon. \end{cases}$$

Here t denotes the interaction-step index at the end of recovery. Both branches carry I_k so that subsequent decisions can use earlier findings and unfinished work. Inspired by the exploration–exploitation trade-off in RL (Sutton & Barto, 2018), ε balances exploiting retained progress through replay with seeking alternative routes from the initial context.

Exploration. Exploration seeks further progress using an *adapted policy*, obtained by updating action preferences from past experience while keeping the LLM frozen. We instantiate it with JitRL, $\pi_k^{\mathrm{ad}} = \mathcal{U}_{\mathrm{Jit}}(\pi_\theta, \mathcal{H}_k)$; retention and recovery do not depend on this particular adaptation rule. Given the execution context c returned by recovery, $\mathrm{Explore}(\pi_k^{\mathrm{ad}}, c)$ continues the current rollout for up to $T-t$ additional steps. At each step j, it samples $a_j \sim \pi_k^{\mathrm{ad}}(\cdot|h_j)$, executes the action in the environment, and appends the resulting observation and reward to the interaction record. Decisions use the evolving history together with the compacted task information carried by c. The function stops at termination or budget exhaustion and returns the complete trajectory:

$$\tau_k = \mathrm{Explore}(\pi_k^{\mathrm{ad}}, c) = \big(h_T, (r_i)_{i=0}^{T-1}\big).$$

Algorithm 1 HIRE: hippocampal replay.

Given: frozen policy π_θ, K updates, T steps per rollout, exploration rate ε.

$\mathcal{H}_0 \leftarrow \varnothing, \quad \mathcal{C}_0 \leftarrow (\varnothing, \underline{R}, \varnothing)$
for $k = 0, \ldots, K-1$ **do**
  $\pi_k^{\mathrm{ad}} \leftarrow \mathcal{U}_{\mathrm{Jit}}(\pi_\theta, \mathcal{H}_k)$
  $c \leftarrow \mathrm{Recover}(\mathcal{C}_k)$
  $\tau_k \leftarrow \mathrm{Explore}(\pi_k^{\mathrm{ad}}, c)$
  $\mathcal{H}_{k+1} \leftarrow \mathcal{H}_k \uplus \{\tau_k\}$
  $\mathcal{C}_{k+1} \leftarrow \mathrm{Retain}(\mathcal{C}_k, \tau_k)$
end for

Recovery and exploration together induce the *composite policy* π_k, with $\tau_k \sim p_{\pi_k}$. The completed trajectory updates both the experience $\mathcal{H}_{k+1}$ for policy adaptation and the consolidation record $\mathcal{C}_{k+1}$ through Retain.

Because this mechanism preserves individual experiences and reactivates them to guide subsequent behavior, mirroring the hippocampal function in CLS, we call it **HI**ppocampal **RE**play (HIRE). Algorithm 1 summarizes this procedure using the three functions defined above.

The following theorem gives conditions under which the three-stage procedure satisfies return consolidation and inherits the expected-return guarantee of Proposition 2.

Theorem 1 (Performance guarantee for HIRE)**.** *Consider HIRE (Algorithm 1) with exploration rate $0 \le \varepsilon < 1$. At every update with a retained prefix, suppose replay reproduces its recorded rewards almost surely within the T-step budget, and the remaining execution has nonnegative cumulative discounted reward almost surely. Then HIRE satisfies ε-consolidation and, at every update k, its composite policy π_k obeys*

$$J(\pi_k) \ge \underline{R} + (1-\varepsilon)(R_k^* - \underline{R}). \tag{5}$$

The proof is in Appendix D. Under these assumptions and for fixed ε, HIRE's expected-return lower bound is non-decreasing as R_k^* improves.

4 Experiments

4.1 Settings

Benchmarks. We evaluate on two benchmarks with contrasting reward structures. **ScienceWorld** tests scientific reasoning through text-based interaction across topics including Matter, Electricity,

Table 1: **ScienceWorld results across task topics.** Mean return with Gemini-2.5-Flash.

Method	Matter	Measure	Electric	Classify	Life Stages	Lifespan	Forces	Chemistry	Avg.
Backbone	2.1	9.9	5.1	26.5	2.8	39.0	14.9	2.6	13.0
Reflexion	1.6	4.5	9.9	37.1	6.2	48.7	35.8	5.7	18.2
EvoTest	5.5	37.8	24.9	58.0	13.5	70.4	64.2	**52.6**	39.7
JitRL	12.6	30.2	18.5	35.2	8.5	73.3	61.4	8.4	29.6
HIRE	**34.5**	**62.1**	**51.0**	**71.8**	**28.4**	**92.6**	**86.6**	41.3	**57.9**

Table 2: **ScienceWorld results across backbones.** Mean return on the Matter and Electric topics; Avg. averages the five backbones.

Method	Matter						Electric					
	Gemini-2.5 Flash	GPT-4o mini	Qwen3 32B	DeepSeek V3	Kimi K2	Avg.	Gemini-2.5 Flash	GPT-4o mini	Qwen3 32B	DeepSeek V3	Kimi K2	Avg.
Backbone	2.1	0.1	1.5	2.4	5.1	2.2	5.1	3.9	4.0	6.3	5.2	4.9
Reflexion	1.6	0.5	0.4	0.9	11.5	3.0	9.9	5.4	7.4	8.0	8.0	7.7
EvoTest	5.5	0.1	4.1	4.5	**49.1**	12.7	24.9	0.2	8.7	12.7	33.2	15.9
JitRL	12.6	0.1	11.2	21.2	27.8	14.6	18.5	13.0	5.3	29.8	38.6	21.0
HIRE	**34.5**	**6.2**	**24.6**	**38.7**	46.0	**30.0**	**51.0**	**53.3**	**39.7**	**37.6**	**54.1**	**47.1**

Table 3: **WebChoreArena results across websites.** Mean success rate (%) with Gemini-3.7-Flash.

Method	Shopping Admin	Shopping	GitLab	Reddit	Avg.
Backbone	40.7	17.7	32.4	6.0	25.5
Reflexion	49.3	20.8	38.8	10.0	31.2
EvoTest	43.6	20.0	33.6	10.0	28.1
JitRL	44.1	19.2	38.2	14.0	30.0
HIRE	**63.6**	**42.3**	**70.4**	**46.0**	**56.2**

and Chemistry (Wang et al., 2022). Tasks such as testing electrical conductivity require coordinated navigation and equipment manipulation, with graded rewards for intermediate progress and a maximum return of 100. **WebChoreArena (WCA)** extends WebArena (Zhou et al., 2024a) with information-intensive workflows across Shopping Admin, Shopping, GitLab, and Reddit (Miyai et al., 2025). Tasks include collecting and tracking information across many pages, performing calculations, and using earlier findings to complete long-term objectives. WCA provides sparse binary outcome rewards: 1 for success and 0 otherwise.

Baselines. We compare against four baselines with frozen LLM weights. **Backbone** uses the LLM directly with a fixed agent configuration and no cross-rollout learning. **Reflexion** converts trajectory feedback into textual self-reflections to guide subsequent trials (Shinn et al., 2023). **EvoTest** uses an evolver agent to analyze completed trajectories and revise the acting agent's prompts and configuration between episodes (He et al., 2026). **JitRL**, our direct test-time RL baseline, adjusts the frozen model's action preferences using advantages estimated from historical outcomes in similar contexts (Li et al., 2026).

Evaluation. We measure performance by *mean return*, the empirical counterpart of the RL objective. For N tasks and M_i evaluated rollouts on task i, we compute $\text{Mean Return} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{M_i}\sum_{k=1}^{M_i} R(\tau_{i,k})$, where $\tau_{i,k}$ is the trajectory from rollout k on task i. We average within each task before averaging across tasks. For WCA, this is the mean success rate, reported as a percentage. Further parameter details are provided in Appendix E. In the tables, Avg. denotes the aggregate mean, blue shading identifies the full HIRE configuration, and bold marks the best result within each comparison.

Table 4: **Component ablations on WebChoreArena.** Recovery: progress recovery through Recover; I_k: compacted task information and associated operations; A: advantage correction in the adapted policy. Δ Avg. is relative to the full configuration (blue).

Components			Mean success rate (%)				Δ Avg.
Recovery	I_k	A	Calculation	Long-term	Massive Memory	Avg.	
–	✓	✓	72.0	64.0	38.0	58.0	−14.0
✓	–	✓	72.0	72.0	48.0	64.0	−8.0
✓	✓	–	78.0	68.0	48.0	64.7	−7.3
✓	✓	✓	**82.0**	**76.0**	**58.0**	**72.0**	–

4.2 Main Results

Performance across ScienceWorld topics. The evaluated HIRE configurations achieve a mean return of 57.9, compared with 39.7 for EvoTest, 29.6 for JitRL, and 18.2 for Reflexion (Table 1). HIRE improves over JitRL on all eight topics and leads all baselines on seven. On Matter and Electric, its returns reach 34.5 and 51.0, versus JitRL's 12.6 and 18.5. These tasks require prerequisite manipulations before achieving the target physical outcome, making the gains consistent with the value of recovering prior progress. HIRE also improves Lifespan from 73.3 to 92.6, extending the benefit to a topic where JitRL already performs well. EvoTest remains strongest on Chemistry (52.6 versus 41.3), so the overall advantage does not imply that replay is preferable for every task type.

Performance across backbones. Under a fixed HIRE configuration, the mean across five backbones rises from JitRL's 14.6 to 30.0 on Matter and from 21.0 to 47.1 on Electric (Table 2). HIRE improves over JitRL in all ten backbone–topic comparisons and ranks first in nine. The Electric gains include GPT-4o-mini (13.0 to 53.3) and Qwen3-32B (5.3 to 39.7), suggesting that support for recovering multi-step progress is useful across different backbones. Performance nevertheless remains task- and model-dependent: GPT-4o-mini reaches only 6.2 on Matter, and EvoTest outperforms HIRE with Kimi-K2 on that topic (49.1 versus 46.0). Recovery can support continued progress but cannot replace the ability to discover useful behavior initially.

Performance on WebChoreArena. HIRE achieves a mean success rate of 56.2%, exceeding JitRL's 30.0% by 26.2 percentage points (Table 3). Reflexion is the strongest baseline at 31.2%, followed by JitRL and EvoTest (28.1%). HIRE leads on all four websites, with gains over JitRL ranging from 19.5 points on Shopping Admin to 32.2 on GitLab. On Reddit, where all baselines score at most 14.0%, HIRE reaches 46.0%. The improvement therefore extends beyond graded scientific tasks to web workflows with sparse completion feedback. We next isolate the contributions of recovery, compacted task information, and advantage-based adaptation on a matched WCA subset.

4.3 Ablation Study

All ablations use a 30-task WCA subset, with ten tasks each from Calculation, Long-term, and Massive Memory. Starting from the full configuration with $\varepsilon = 0.4$, Table 4 removes one component at a time: (i) *recovery* of the saved action prefix, by setting $\varepsilon = 1$ so that $\mathrm{Recover}(\mathcal{C}_k)$ starts at h_0 while retaining I_k; (ii) *compacted task information*, by setting $I_k = \varnothing$ in $\mathcal{C}_k$ and disabling its associated operations; and (iii) advantage correction in the *adapted policy* π_k^{ad}, by setting $A(h, a) = 0$ in (1). These interventions correspond to the components defined in Section 3.3. We also sweep $\varepsilon \in \{0.2, 0.4, 0.6, 0.8, 1.0\}$ while keeping advantage correction and compacted task information enabled.

Recovery. Removing saved-prefix replay produces the largest observed decrease, from 72.0% to 58.0% (Table 4). The loss is greatest on Massive Memory (20.0 points), compared with 10.0 on Calculation and 12.0 on Long-term. Since I_k and advantage correction remain enabled, the results show that information reuse and policy adaptation do not fully substitute for recovering the saved action prefix.

Compacted task information. Removing I_k and its associated operations lowers mean success to 64.0%, an 8.0-point decrease. Calculation and Massive Memory each lose 10.0 points, while Long-

term loses 4.0. These losses complement the recovery ablation: retaining and replaying actions does not capture all the information needed for subsequent decisions.

Adapted policy. Setting $A(h, a) = 0$ lowers mean success to 64.7%, a 7.3-point decrease, with losses across all three task types. The largest loss is again on Massive Memory (10.0 points). Thus, advantage-based action selection contributes beyond replay and information reuse, supporting the combination of recovery with an adapted policy for further exploration.

Exploration rate. Figure 4 shows the highest performance at $\varepsilon = 0.4$ for all three task types among the tested settings. The overall mean increases from 68.0% at $\varepsilon = 0.2$ to 72.0%, then falls to 69.3%, 64.0%, and 58.0% at $\varepsilon = 0.6, 0.8, 1.0$, respectively. The $\varepsilon = 1$ endpoint matches the recovery ablation in Table 4. The intermediate optimum is consistent with balancing reuse of prior progress against exploration of alternative routes, rather than always restarting or relying more heavily on replay.

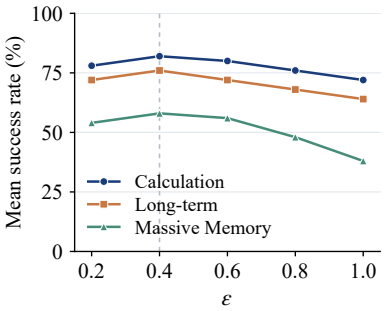


Figure 4: **Exploration rates on WCA.**

5 Related Work

Agentic RL. Training-time agentic RL optimizes model parameters using environmental feedback to improve sequential decision-making. ArCHer combines turn-level value learning with token-level policy optimization for multi-turn interactions (Zhou et al., 2024b). DigiRL develops autonomous offline-to-online RL for device-control agents (Bai et al., 2024), while WebRL trains web agents through a self-evolving task curriculum and outcome-based feedback (Qi et al., 2025). Moving beyond parameter updates, JitRL explores gradient-free test-time RL by estimating action advantages from past experience and applying closed-form logit corrections to a frozen LLM (Li et al., 2026). Our work also studies gradient-free test-time RL, focusing on consolidating discovered return gains as the policy continues to adapt.

Consolidation of Learned Behavior. Continual learning seeks to acquire new capabilities while retaining existing ones. Elastic weight consolidation protects parameters important to earlier tasks (Kirkpatrick et al., 2017), while Policy Consolidation constrains policy changes through a cascade of networks operating at different timescales (Kaplanis et al., 2019). CLEAR combines learning from new experience with off-policy replay and behavioral cloning to reduce forgetting (Rolnick et al., 2019). These methods aim to preserve acquired capabilities as model parameters are updated. We instead study consolidation during test-time adaptation with fixed model parameters, seeking to preserve previously achieved returns as action policy changes. Our consolidation criterion quantifies this preservation through expected shortfall and connects it to a lower bound on expected return.

Test-Time Inference Enhancement. Training-free methods improve frozen agents by modifying the information and procedures used at inference time. Reflexion converts feedback into textual reflections for subsequent trials (Shinn et al., 2023), and Voyager builds a library of executable skills (Wang et al., 2024a). EvoTest evolves prompts, stored experience, and tool-use routines between episodes (He et al., 2026). HIRE likewise aims to enhance agent capabilities at test time without updating model weights. It combines experience-based policy adaptation with replay and compacted task information, allowing the agent to recover prior progress and build on it through exploration.

6 Conclusion

We identify a discovery–consolidation gap in gradient-free test-time RL, where advantage re-estimation can reverse successful action preferences. Inspired by CLS theory, HIRE complements statistical adaptation with hippocampal replay, enabling exploration to build on recovered progress. Under explicit recovery and execution conditions, HIRE satisfies our return consolidation criterion and guarantees an expected-return lower bound that is non-decreasing as the best verified return improves, for a fixed exploration rate. Experiments on ScienceWorld and WebChoreArena show the best overall performance among the evaluated methods.

AI USE STATEMENT

In this work, we used generative AI tools only for editing the text (grammar, spelling, and word choice). We did not use generative AI tools for research ideation, for writing or verifying code, for generating or analyzing experimental results, or for producing figures, and the remaining required-disclosure tasks are not applicable to this work. We have reviewed all AI-assisted edits and take responsibility for the final content of this work.

REFERENCES

Anthropic. The Claude 3 model family: Opus, Sonnet, Haiku. Model card, 2024. URL https://www.anthropic.com/claude-3-model-card.

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. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, pp. 12461–12495, 2024.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 10088–10115, 2023. URL https://arxiv.org/abs/2305.14314.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169, 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

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. doi: 10.1038/s41586-025-09422-z.

Yufei He, Juncheng Liu, Yue Liu, Yibo Li, Tri Cao, Zhiyuan Hu, Xinxing Xu, and Bryan Hooi. EvoTest: Evolutionary test-time learning for self-improving agentic systems. In *International Conference on Learning Representations (ICLR)*, 2026.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Policy consolidation for continual reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3242–3251, 2019. URL https://proceedings.mlr.press/v97/kaplanis19a.html.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20(7):512–534, 2016. doi: 10.1016/j.tics.2016.05.004.

Yibo Li, Zijie Lin, Ailin Deng, Xuan Zhang, Yufei He, Shuo Ji, Tri Cao, and Bryan Hooi. Just-in-time reinforcement learning: Continual learning in LLM agents without gradient updates. In *International Conference on Machine Learning (ICML)*, 2026.

Zechun Liu, Changsheng Zhao, Forrest Iandola, Chen Lai, Yuandong Tian, Igor Fedorov, Yunyang Xiong, Ernie Chang, Yangyang Shi, Raghuraman Krishnamoorthi, Liangzhen Lai, and Vikas Chandra. MobileLLM: Optimizing sub-billion parameter language models for on-device use cases. In *International Conference on Machine Learning (ICML)*, pp. 32431–32454, 2024.

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):419–457, 1995. doi: 10.1037/0033-295X.102.3.419.

Atsuyuki Miyai, Zaiying Zhao, Kazuki Egashira, Atsuki Sato, Tatsumi Sunada, Shota Onohara, Hiromasa Yamanishi, Mashiro Toyooka, Kunato Nishina, Ryoma Maeda, Kiyoharu Aizawa, and Toshihiko Yamasaki. WebChoreArena: Evaluating web browsing agents on realistic tedious web tasks. *arXiv preprint arXiv:2506.01952*, 2025. URL https://arxiv.org/abs/2506.01952.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. doi: 10.1038/nature14236.

OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, Jie Tang, and Yuxiao Dong. WebRL: Training LLM web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations (ICLR)*, pp. 79791–79821, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c66e1fcc9691aae706250638f36f681b-Abstract-Conference.html.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.neurips.cc/paper_files/paper/2019/hash/fa7cdfad1a5aaf8370ebeda47a1ff1c3-Abstract.html.

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, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 8634–8652, 2023.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

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. *Transactions on Machine Learning Research*, 2024a.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b. doi: 10.1007/s11704-024-40231-1.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 11279–11298, 2022. doi: 10.18653/v1/2022.emnlp-main.775.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhong-Zhi Li, Xiangyuan Xue, Yijiang Li, Yifan Zhou, Yang Chen, Chen Zhang, Yutao Fan, Zihu Wang, Songtao Huang, Francisco Piedrahita Velez, Yue Liao, Hongru Wang, Mengyue Yang, Heng Ji, Jun Wang, Shuicheng Yan, Philip Torr, and Lei Bai. The landscape of agentic reinforcement learning for LLMs: A survey. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=RY19y2RI1O. Survey Certification.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. WebArena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning (ICML)*, pp. 62178–62209, 2024b.

A Proof of Proposition 3

Proposition 3 (Return change under fixed-reference updates)**.** *In a finite-horizon POMDP with bounded rewards, let π_{k+1} be obtained from (1) using the true advantage A^{π_k} of the current policy π_k, with $\beta > 0$. Assume common finite action supports with positive probabilities and finite expected KL terms. Then*

$$J(\pi_{k+1}) - J(\pi_k) = \frac{1}{\beta}\mathbb{E}_{\tau\sim p_{\pi_{k+1}}} \sum_{t=0}^{T-1} \gamma^t \Big[D_{\mathrm{KL}}(\pi_{k+1}\|\pi_\theta) - D_{\mathrm{KL}}(\pi_k\|\pi_\theta) + D_{\mathrm{KL}}(\pi_k\|\pi_{k+1}) \Big]_{h_t}, \tag{6}$$

where all KL divergences compare action distributions at the same history h_t.

Proof. Fix π_k and the task's initial-state distribution. Define

$$V^{\pi_k}(h_t) = \mathbb{E}_{\pi_k}\left[\sum_{u=t}^{T-1} \gamma^{u-t} r_u \,|\, h_t\right], \qquad V^{\pi_k}(h_T) = 0.$$

The true advantage is

$$A^{\pi_k}(h_t, a) = \mathbb{E}[r_t + \gamma V^{\pi_k}(h_{t+1}) \,|\, h_t, a_t = a] - V^{\pi_k}(h_t).$$

Hence $\mathbb{E}_{a\sim\pi_k(\cdot|h_t)} A^{\pi_k}(h_t, a) = 0$.

At a fixed history h, write

$$Z(h) = \sum_a \pi_\theta(a|h) \exp(\beta A^{\pi_k}(h, a)).$$

The update satisfies $\log(\pi_{k+1}/\pi_\theta) = \beta A^{\pi_k} - \log Z(h)$. Taking expectations under π_k and π_{k+1}, respectively, gives

$$\begin{aligned} D_{\mathrm{KL}}(\pi_k\|\pi_{k+1}) &= D_{\mathrm{KL}}(\pi_k\|\pi_\theta) + \log Z(h), \\ D_{\mathrm{KL}}(\pi_{k+1}\|\pi_\theta) &= \beta\mathbb{E}_{a\sim\pi_{k+1}} A^{\pi_k}(h, a) - \log Z(h). \end{aligned}$$

Eliminating $\log Z(h)$ yields

$$\beta\mathbb{E}_{a\sim\pi_{k+1}} A^{\pi_k}(h, a) = D_{\mathrm{KL}}(\pi_{k+1}\|\pi_\theta) - D_{\mathrm{KL}}(\pi_k\|\pi_\theta) + D_{\mathrm{KL}}(\pi_k\|\pi_{k+1}). \tag{7}$$

Taking expectations along trajectories generated by π_{k+1} and telescoping the value terms gives

$$\begin{aligned} \mathbb{E}_{\tau\sim p_{\pi_{k+1}}} \sum_{t=0}^{T-1} \gamma^t A^{\pi_k}(h_t, a_t) &= \mathbb{E}_{\tau\sim p_{\pi_{k+1}}} \sum_{t=0}^{T-1} \gamma^t \big[r_t + \gamma V^{\pi_k}(h_{t+1}) - V^{\pi_k}(h_t)\big] \\ &= J(\pi_{k+1}) - \mathbb{E}[V^{\pi_k}(h_0)] \\ &= J(\pi_{k+1}) - J(\pi_k). \end{aligned}$$

The initial-history distribution is shared by both policies. Substituting (7) proves (6). □

Strict regression with exact advantages. Consider a two-stage task with $\gamma = 1$ in which action a^* at the initial history terminates with reward $M > 0$, while action b leads to a second decision with zero immediate reward. At that decision there are $n \geq 2$ terminal actions with rewards $0 \leq x_i < M$ that are not all equal. Let the fixed reference choose a^* with any probability $\alpha \in (0, 1)$ and assign any positive probabilities q_i to the terminal actions, with $\sum_i q_i = 1$.

Initialize $\pi_0 = \pi_\theta$ and use exact advantages at every update. Because the terminal-action values do not depend on the policy, their distribution after the first and all subsequent updates is

$$q_i^+ = \frac{q_i e^{\beta x_i}}{\sum_j q_j e^{\beta x_j}}.$$

Write $v_0 = \sum_i q_i x_i$ and $v_1 = \sum_i q_i^+ x_i$ for the corresponding values of branch b. For $\psi(\lambda) = \log \sum_i q_i e^{\lambda x_i}$, $\psi'(0) = v_0$, $\psi'(\beta) = v_1$, and $\psi''(\lambda) > 0$ because the rewards are not all equal. Thus, for every $\beta > 0$, $v_0 < v_1 < M$.

The probabilities of choosing a^* after the first two updates are

$$p_1 = \frac{\alpha e^{\beta M}}{\alpha e^{\beta M} + (1-\alpha)e^{\beta v_0}}, \qquad p_2 = \frac{\alpha e^{\beta M}}{\alpha e^{\beta M} + (1-\alpha)e^{\beta v_1}}.$$

Since $v_1 > v_0$, we have $p_2 < p_1$. The continuation under branch b is identical for π_1 and π_2, so

$$J(\pi_2) - J(\pi_1) = (p_2 - p_1)(M - v_1) < 0.$$

Thus every task and full-support reference in this family exhibits regression for every finite $\beta > 0$, despite exact advantage evaluation.

Discussion. The proof combines the performance-difference identity (Kakade & Langford, 2002) with the KL structure of regularized policy updates (Geist et al., 2019). With the current policy π_k as the reference, the bracket in (6) becomes $D_{\mathrm{KL}}(\pi_{k+1}\|\pi_k) + D_{\mathrm{KL}}(\pi_k\|\pi_{k+1}) \geq 0$ under exact evaluation. In the family above, the shortfall below the verified level M equals $M - J(\pi)$ and therefore increases after the second update.

B Proof of Proposition 1

Proposition 1 (Return decomposition). For every update k, with the accumulated experience held fixed,

$$J(\pi_k) = R_k^* - \mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R_k^* - R(\tau_k))_+] + \mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R(\tau_k) - R_k^*)_+].$$

Proof. Fix $\mathcal{H}_k$, so that π_k and R_k^* are fixed. The identity $x = (x)_+ - (-x)_+$ gives

$$R(\tau_k) = R_k^* - (R_k^* - R(\tau_k))_+ + (R(\tau_k) - R_k^*)_+.$$

Taking expectations over $\tau_k \sim p_{\pi_k}$ proves the result. All terms are integrable because returns are bounded. □

C Proof of Proposition 2

Proposition 2 (Return guarantee). For $0 \leq \delta < 1$, every policy induced by a δ-consolidating adaptation rule satisfies

$$J(\pi_k) \geq \underline{R} + (1-\delta)(R_k^* - \underline{R}).$$

Proof. Fix $\mathcal{H}_k$. By Proposition 1 and Definition 1,

$$\begin{aligned} J(\pi_k) &\geq R_k^* - \mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R_k^* - R(\tau_k))_+] \\ &\geq R_k^* - \delta(R_k^* - \underline{R}) = \underline{R} + (1-\delta)(R_k^* - \underline{R}). \end{aligned}$$

□

D Proof of Theorem 1

Theorem 1 (Performance guarantee for HIRE). Consider HIRE with exploration rate $0 \leq \varepsilon < 1$. At every update with a retained prefix, suppose replay reproduces its recorded rewards almost surely within the T-step budget, and the remaining execution has nonnegative cumulative discounted reward almost surely. Then HIRE satisfies ε-consolidation and, at every update k,

$$J(\pi_k) \geq \underline{R} + (1-\varepsilon)(R_k^* - \underline{R}).$$

Proof. Fix $\mathcal{H}_k$ and $\mathcal{C}_k$. If no prefix is retained, $R_k^* = \underline{R}$, so the consolidation inequality holds with zero shortfall. Otherwise, let ℓ be the retained prefix length. By Retain and the two execution assumptions, the recovered branch satisfies

$$R(\tau_k) = \sum_{i=0}^{\ell-1} \gamma^i r_i^* + \sum_{i=\ell}^{T-1} \gamma^i r_i \geq R_k^* \qquad \text{almost surely.}$$

Let τ_k^{ad} denote a fresh rollout using the adapted policy and the same information I_k. Conditioning on the recovery branch gives

$$\begin{aligned}\mathbb{E}_{\tau_k \sim p_{\pi_k}}[(R_k^* - R(\tau_k))_+] &= \varepsilon\, \mathbb{E}[(R_k^* - R(\tau_k^{\mathrm{ad}}))_+] \\ &\leq \varepsilon(R_k^* - \underline{R}).\end{aligned}$$

This is Definition 1 with $\delta = \varepsilon$. Applying Proposition 2 completes the proof. □

Sufficient execution conditions. Fixed resets and deterministic transitions, observations, and rewards make faithful action replay reproduce the prefix by induction on the interaction steps. Non-negative continuation rewards or cost-free termination then preserve its return. For terminal-score evaluation with initial score zero, use $\gamma = 1$ and $r_i = S_{i+1} - S_i$, where S_i is the score after i actions.

Implications for expected return. For fixed ε, the guaranteed floor is non-decreasing because Retain never decreases R_k^*. More specifically, branch conditioning gives

$$\begin{aligned}J(\pi_k) &\geq (1-\varepsilon)R_k^* + \varepsilon J(\pi_k^{\mathrm{ad}}), \\ J(\pi_k) - J(\pi_k^{\mathrm{ad}}) &\geq (1-\varepsilon)\big(R_k^* - J(\pi_k^{\mathrm{ad}})\big).\end{aligned}$$

Here π_k^{ad} uses the same experience and information I_k as the fresh branch. Thus $R_k^* > J(\pi_k^{\mathrm{ad}})$ guarantees a strict improvement over this matched adapted policy.

E Experimental Settings

Tables 5 and 6 report one reference configuration for each benchmark. Each table lists the HIRE parameters in Algorithm 1 and the JitRL settings used for policy adaptation. The return discount γ applies to benchmark evaluation; the separate advantage discount applies to historical returns used by JitRL. Correction weights multiply normalized advantages in the implemented action scores.

ScienceWorld. We select 25 tasks across eight topics. Each task receives a budget of 50 rollouts with at most 60 actions per rollout ($K = 50$, $T = 60$).

WebChoreArena. The initial evaluation set contains 100 tasks across four websites. Each task receives a budget of 5 rollouts with at most 50 actions per rollout ($K = 5$, $T = 50$).

Table 5: **ScienceWorld parameters.** One reference configuration; k counts completed rollouts and t counts interaction steps in the current rollout.

Parameter	Value
Evaluation	
Number of tasks	25
Main backbone	Gemini-2.5-Flash
Return discount γ	1
HIRE	
Updates per task K	50
Action budget per rollout T	60
Exploration rate ε	0.25
JitRL	
Advantage correction weight β_k	$\min\{1 + 0.5k/50,\ 1.5\}$
Advantage-return discount	0.5
Number of matched experience entries	10
Experience similarity threshold	$0.95 - 0.1\min\{t/20, 1\}$
Initial action candidates	3
Action-generation temperature	0.8
Recent-history buffer capacity	30
Action-confidence exponent	0.65

Table 6: **WebChoreArena parameters.** One reference configuration; $\varepsilon = 0.4$ is the reference setting for the exploration-rate sweep in Section 4.3.

Parameter	Value
Evaluation	
Number of selected tasks	100
Backbone	Gemini-3.7-Flash
Return discount γ	1
HIRE	
Updates per task K	5
Action budget per rollout T	50
Exploration rate ε	0.4
JitRL	
Advantage correction weight β	1
Advantage-return discount	0.1
Number of matched experience entries	3
Experience similarity threshold	$0.8 - 0.1\min\{t/20, 1\}$
Task similarity threshold	0.27
Initial action candidates	3
Action-generation temperature	0.8
Recent-history buffer capacity	30

F CASE STUDY

We present four state–action cases covering two uses of recovery: first resuming from partial progress to complete an unfinished task, then reproducing an already successful trajectory. Each use is illustrated on both ScienceWorld and WebChoreArena, with an unsuccessful or lower-return JitRL trajectory on the same task and backbone for comparison.

Extending partial progress in ScienceWorld. Figure 5 shows a living-thing delivery task. An earlier HIRE rollout remains unfinished at 75 after 60 actions. A later rollout replays five actions to reach 67, then completes the task through 14 new actions, attaining 100. JitRL reaches 83 but remains unfinished after 60 actions, repeatedly inspecting objects and revisiting rooms.

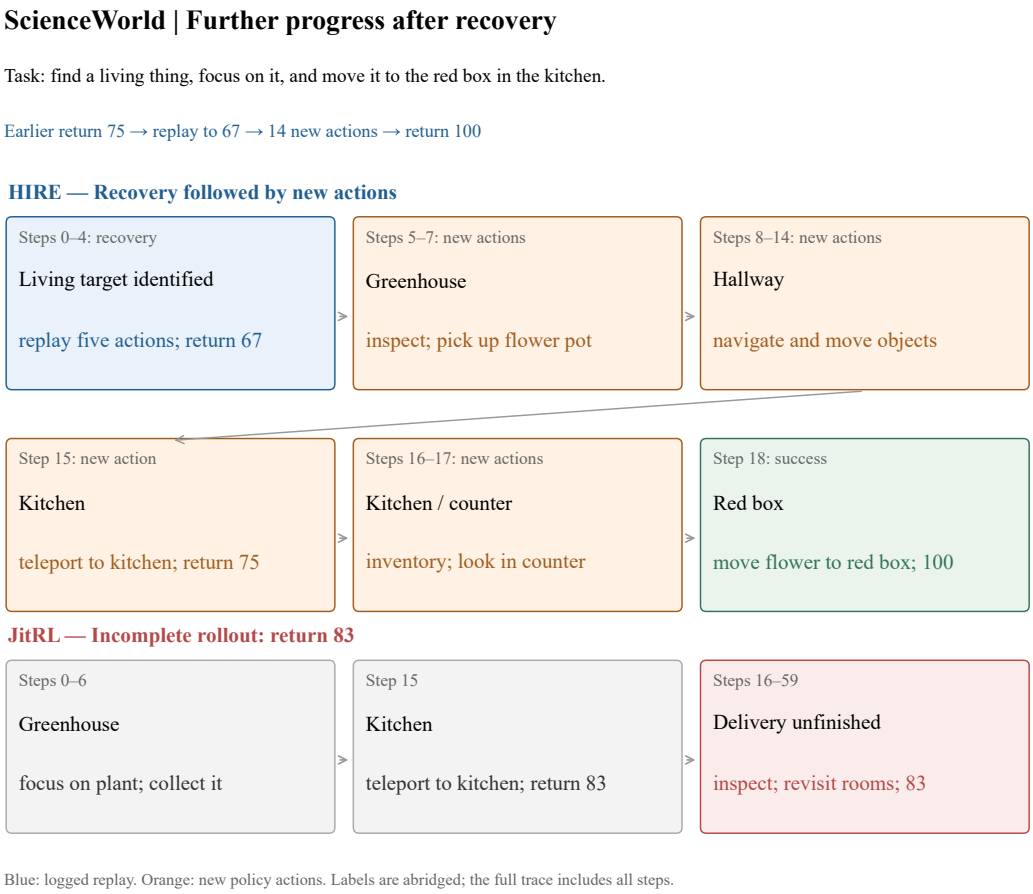


Figure 5: **Recovery followed by further exploration in ScienceWorld.** Five replayed actions recover partial progress, after which new actions complete the task in a 19-action rollout; the JitRL comparison ends at 83 after 60 actions. Blue marks replay, orange marks new policy actions, and green marks completion.

Completing a partially explored web task. Figure 6 concerns identifying the highest-rated product and its most recent completed order. The first two rollouts exhaust their 50-action budgets without an accepted answer. The next rollout uses seven replayed actions and two interface-repair actions to recover report navigation, followed by 17 further actions that complete the query. It submits an accepted answer within 26 actions. Unlike replay of a completed workflow, the retained trajectory contains no previously successful answer to reproduce. The JitRL comparison repeatedly revisits review and order reports, exhausting its 50-action budget without submitting an answer.

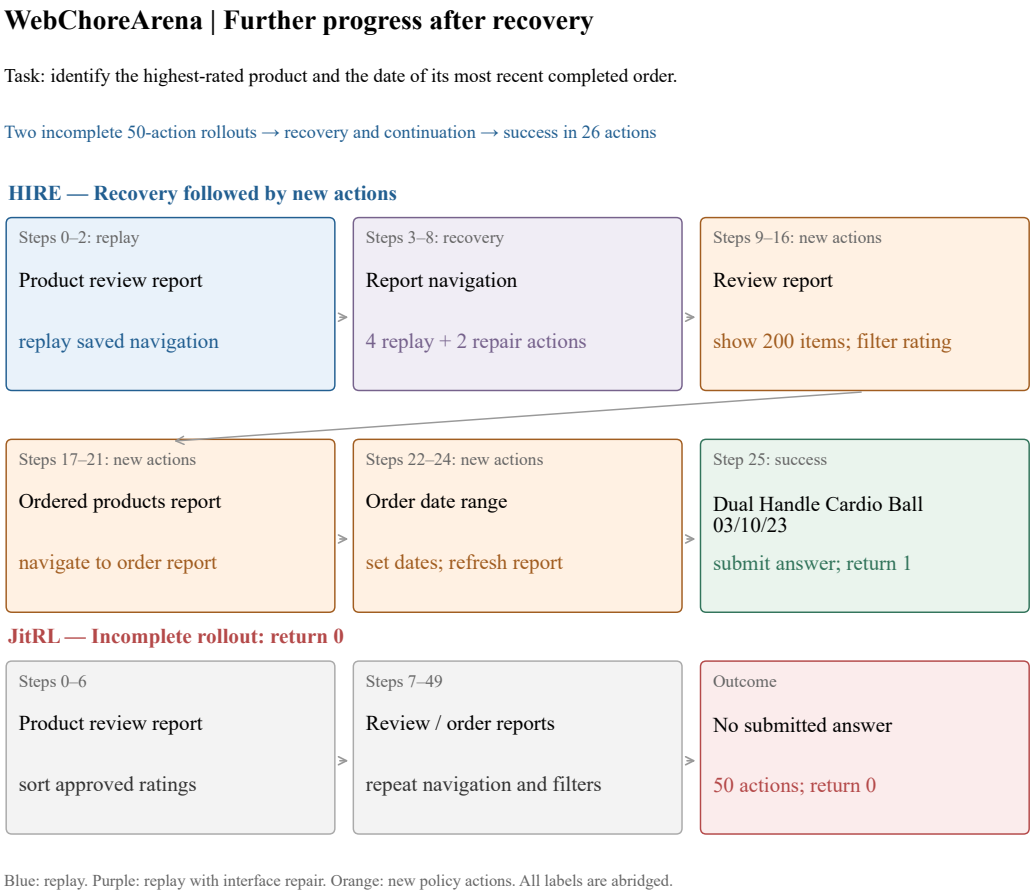


Figure 6: **Recovery followed by further exploration in WebChoreArena.** HIRE completes the recovered query, while JitRL remains unfinished after 50 actions. Purple marks a recovery segment containing both replay and interface repair; orange marks subsequent new policy actions.

Recovering a successful scientific procedure. Figure 7 illustrates how replay reinstates a previously successful freezing-water procedure. HIRE repeats the recorded actions for collecting water, identifying the target substance, and transferring it to the freezer, attaining a return of 100. In the contrasted JitRL trajectory, the agent instead focuses on the fridge and terminates with return 0. The replayed sequence makes the successful procedure available without requiring the action-selection policy to reconstruct it.

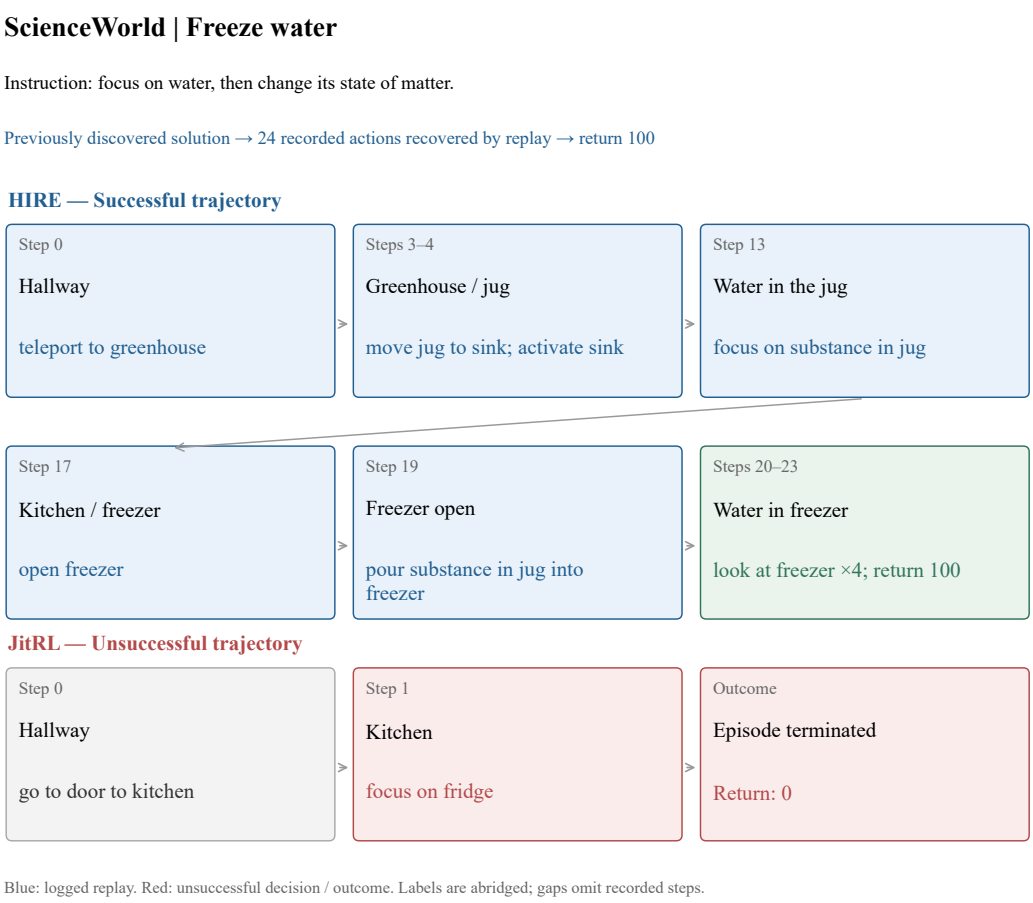


Figure 7: **State–action illustration on ScienceWorld.** HIRE replays a successful freezing-water procedure, whereas the contrasted JitRL rollout terminates after selecting the wrong target. Blue indicates replayed actions; step labels identify the portions shown in this abridged visualization.

Recovering a completed web workflow. Figure 8 illustrates replay on a GitLab branch-comparison task. Both trajectories reach the repository, but JitRL repeatedly swaps identical comparison endpoints and exhausts its action budget without submitting an answer. The replay trajectory reproduces a previously accepted workflow, including its final answer.

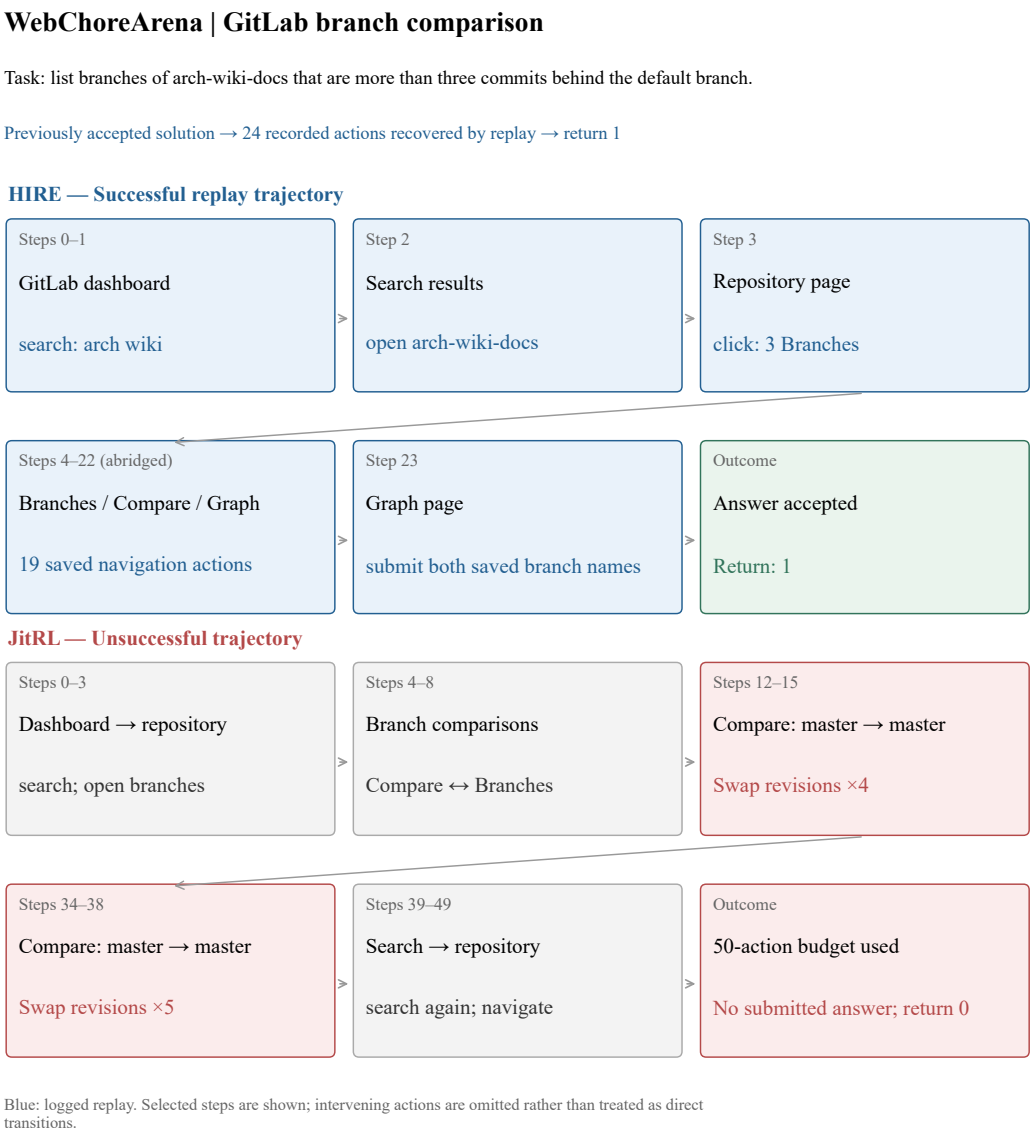


Figure 8: **State–action illustration on WebChoreArena.** The replayed workflow reaches an accepted answer, while the contrasted JitRL trajectory ends without an answer after repeated navigation.