

ExpRL: Unified Non-Parametric and Parametric Experience Learning for LLM Agent Evolution

Zhuoran Jin^{1,2,*} Lingshuai Wang^{1,3,*} Yupu Hao^{1,2} Tianyi Men^{1,2}
Zhitao He⁴ Daojian Zeng⁵ Kang Liu^{1,2} Jun Zhao^{1,2,†}

¹The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³School of Advanced Interdisciplinary Sciences, University of Chinese Academy of Sciences

⁴Hong Kong University of Science and Technology

⁵Yuelushan Digital Intelligence Laboratory, Hunan Normal University
{zhuoran.jin,kliu,jzhao}@nlpr.ia.ac.cn, wanglingshuai2026@ia.ac.cn

Abstract

Large language model (LLM)-based agents solve complex long-horizon tasks through iterative interaction with environments, where effectively learning from experience under sparse or delayed feedback remains a key challenge. Existing approaches typically rely on either non-parametric memory-based experience reusing or parametric reinforcement learning, each with inherent limitations when used in isolation. To this end, we propose **ExpRL**, a unified experience learning framework that bridges non-parametric memory and parametric policy optimization for LLM agents. ExpRL induces abstracted success and failure experiences from on-policy interaction trajectories, stores them in an external memory, and integrates experience acquisition and utilization directly into the reinforcement learning process via an agentic experience acquisition mechanism. This design enables memory and policy to co-evolve and mutually reinforce each other, effectively unifying online policy learning with offline experience reuse. Experiments on ALFWorld and WebShop demonstrate that ExpRL achieves an average performance improvement of **13.0%** over GRPO, with improved learning efficiency and stability. We further show that the induced experience memory generalizes across different models and can be directly transferred to stronger LLMs without additional training.

1 Introduction

Large language models (LLMs) (OpenAI, 2023; Dubey et al., 2024; Yang et al., 2024; Anthropic,

2024; DeepSeek-AI et al., 2025; Yang et al., 2025) have emerged as a promising paradigm by acting as autonomous agents capable of solving complex, long-horizon tasks (Yao et al., 2023; Zhou et al., 2024; Xie et al., 2024; Jimenez et al., 2024; Xi et al., 2025a). Unlike traditional reasoning paradigms that generate solutions in a single pass, LLM agents follow a closed-loop interaction process, continuously observing states, executing actions, and receiving feedback from the environment. Through iterative interactions with the environment, agents are able to continuously refine and evolve their behavior, allowing them to learn from past **experience** (Zhao et al., 2024; Zhang et al., 2025a).

In long-horizon agentic tasks, agents operate over extended interaction sequences and frequently encounter sparse or delayed feedback from the environment (Feng et al., 2025), which poses a central challenge in effectively acquiring and leveraging experience from interactions (Liu et al., 2025a; Qin et al., 2025). Recent studies on agentic environments have highlighted experience-driven agent evolution as an important pathway for improving agent capabilities through environment interaction (Li et al., 2026). Existing experience learning approaches for LLM agents can be broadly categorized into two paradigms: *non-parametric experience learning* and *parametric experience learning*.

Non-parametric experience learning (Tang et al., 2025; Cai et al., 2025b; Zhou et al., 2025) stores past interactions in an external **memory**, such as raw trajectories or abstracted experiences, and retrieves relevant experiences at inference time to condition agent behavior, as shown in Figure 1(a). By learning from successful or failed behaviors, these methods provide explicit guidance without

*Equal contribution.

†Corresponding author.

Code: <https://github.com/Wlings/ExpRL>

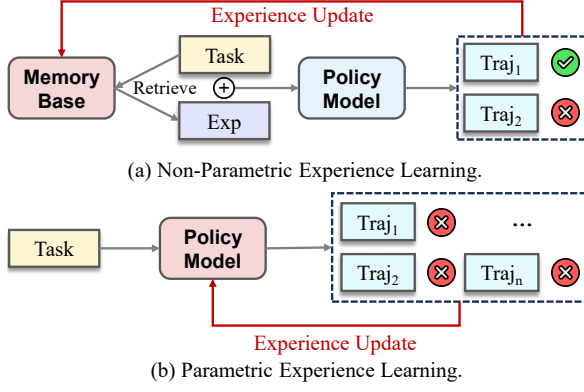


Figure 1: Comparison between Non-Parametric and Parametric Experience Learning.

modifying model parameters, making them flexible and efficient in practice. However, non-parametric experience learning suffers from inherent limitations. Since experience is not internalized into the policy itself, performance improvements heavily depend on the quality, coverage, and retrieval accuracy of the memory (Xiong et al., 2025). Moreover, heuristic retrieval mechanisms may introduce irrelevant or erroneous experiences, which may mislead the policy’s decision-making process.

Parametric experience learning aims to internalize experience directly into the policy by updating model parameters by reinforcement learning (RL) (Shao et al., 2024; Jin et al., 2025; Zhan et al., 2025; Xue et al., 2025; Liu et al., 2025b). As shown in Figure 1(b), the policy is optimized by exploring trajectories in the environment and updating model parameters according to the reward feedback returned by the environment. Despite the widespread adoption of RL, applying parametric experience learning to LLM agents remains challenging. First, RL critically depends on the quality of rollout trajectories, which is largely constrained by the underlying base model, often resulting in sparse or missing positive rewards for challenging tasks (Zhan et al., 2025; Yue et al., 2025). RL primarily reinforces behaviors already present in the base policy, making it difficult to induce fundamentally new capabilities beyond existing trajectory patterns (Yan et al., 2025). Second, parametric updates tend to entangle task-specific experience with model parameters, hindering the abstraction of transferable high-level experience and limiting generalization. Hence, these limitations raise a core question:

Is it possible to unify non-parametric and parametric experience learning in LLM agents, enabling memory and policy to co-evolve and mutually reinforce each other?

To answer this question, we introduce **ExpRL**, a unified experience learning framework that bridges non-parametric memory and parametric policy optimization for LLM agents. ExpRL is built upon three key components: (1) Instead of discarding rollout trajectories generated during RL, ExpRL explicitly exploits them by inducing abstracted success and failure experiences from interaction trajectories. These distilled experiences are then stored in an external experience memory for reuse. (2) ExpRL incorporates the distilled experience into the agent’s decision-making at each interaction step, enabling experience-guided interaction with the environment. This design substantially improves the quality of sampled trajectories, resulting in more stable and effective RL training. (3) ExpRL introduces an agentic experience acquisition mechanism, where the agent proactively formulates experience queries conditioned on the current task and state. Experience acquisition and utilization are tightly integrated into the RL optimization process, enabling experience retrieval and policy learning to co-evolve and mutually reinforce each other. By jointly leveraging on-policy interaction trajectories and off-policy distilled experiences stored in memory, ExpRL effectively unifies online policy optimization with offline experience reuse.

In this paper, we evaluate ExpRL on two representative long-horizon agent benchmarks: ALF-World (Shridhar et al., 2021), which focuses on embodied task execution in simulated household environments, and WebShop (Yao et al., 2022), which models complex, goal-oriented web navigation. We conduct experiments using Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct as the base policy models. Across both benchmarks, ExpRL achieves an average performance improvement of **13.0%** over GRPO and demonstrates better learning efficiency, highlighting the effectiveness of integrating experience into GRPO (Shao et al., 2024). We further evaluate ExpRL in a more challenging agentic RAG scenario, where it demonstrates competitive performance with strong baselines such as Search-R1 (Jin et al., 2025) and GiGPO (Feng et al., 2025). Our ablation results demonstrate that tightly coupling experience acquisition with policy optimization significantly improves both training

stability and final performance. We also validate the generality of the experience memory induced by ExpRL, showing that it can be directly plugged into stronger LLMs such as GPT-4o and GPT-5.1 without additional training. Our contributions are summarized as follows:

- We propose **ExpRL**, a unified experience learning framework that bridges non-parametric experience and parametric experience for LLM agents, enabling memory and policy to co-evolve and mutually reinforce each other in long-horizon agentic tasks.
- We introduce an *agentic experience acquisition* mechanism that tightly integrates experience induction, retrieval, and utilization into the RL optimization process by jointly leveraging on-policy interaction trajectories and off-policy distilled experiences.
- We conduct extensive experiments on two agent benchmarks, ALFWorld and WebShop. Results show that ExpRL achieves an average performance improvement of 13.0% over GRPO. Furthermore, we demonstrate that the experience memory induced by ExpRL exhibits strong cross-model generalization and can be directly plugged into stronger LLMs without additional training.

2 Preliminaries

2.1 Task Definition

We define a multi-turn agentic task as a sequential decision-making process, where an LLM agent parameterized by θ serves as the policy π_θ and interacts with an environment $\mathcal{E} = (\mathcal{S}, \mathcal{A}, r)$ over multiple steps to progressively accomplish a given task, with $\mathcal{S}$ denoting the state space, $\mathcal{A}$ the action space, and r the reward function. Each task is initialized by the initial state $s_0 \in \mathcal{S}$. At each interaction step $k = 1, 2, \dots$, the policy π_θ observes the current state $s_k \in \mathcal{S}$ and generates a textual action

$$a_k \sim \pi_\theta(\cdot | s_k),$$

where $a_k \in \mathcal{A}$ denotes a text-based action from the action space. The environment then executes the action and transitions to a new state $s_{k+1} \in \mathcal{S}$. The interaction proceeds until a terminal condition is met or a maximum horizon is reached, resulting in a trajectory of K interaction steps

$$\tau = \{(s_0, a_0), (s_1, a_1), \dots, (s_{K-1}, a_{K-1})\}.$$

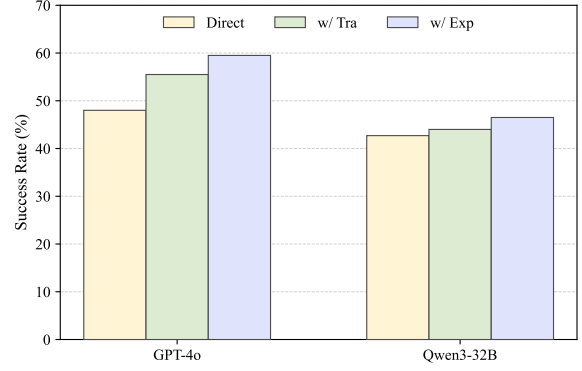


Figure 2: Evaluation results on ALFWorld. *Direct* denotes directly prompting the model to complete the task, *w/ Tra* augments the model with raw trajectories, and *w/ Exp* augments the model with abstracted experiences.

Upon completion of the trajectory, the environment provides a scalar reward or return $r(\tau) \in [0, 1]$ that reflects the overall task outcome.

2.2 Policy Optimization

To optimize the policy π_θ , we employ policy gradient-based methods (Xi et al., 2025b) (e.g., PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and REINFORCE++ (Hu, 2025)), which directly optimize the policy parameters to maximize the expected return obtained from environment feedback. Formally, given the trajectory τ , the optimization objective is to maximize the expected return over trajectories induced by the policy:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} [r(\tau)]. \quad (1)$$

The gradient of this objective with respect to the policy parameters can be expressed using the likelihood ratio formulation:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[r(\tau) \sum_{k=0}^{K-1} \nabla_\theta \log \pi_\theta(a_k | s_k) \right]. \quad (2)$$

Recent group-based RL methods have been proposed, which estimate advantages by comparing multiple rollouts generated from the same prompt.

3 Method

In this section, we present ExpRL, a unified experience learning framework for LLM agents. As illustrated in Figure 3, ExpRL comprises three core components: (1) **abstracted experience induction**, (2) **agentic experience acquisition**, and (3) **memory-policy co-evolution**, which enable effective experience reuse and stable policy optimization.

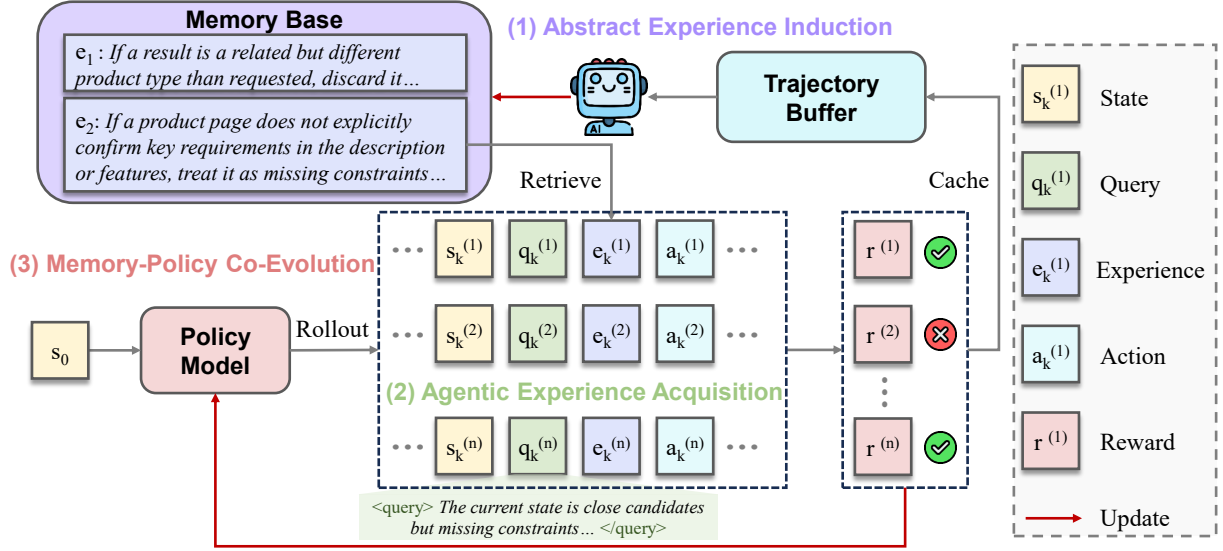


Figure 3: Overview of ExpRL, which consists of three core components: (1) **abstracted experience induction**, (2) **agentic experience acquisition**, and (3) **memory-policy co-evolution**.

3.1 Abstracted Experience Induction

We first study the effectiveness of non-parametric experience for multi-turn agentic tasks. Specifically, we investigate two forms of experience representations stored in external memory $\mathcal{M}$: raw trajectories τ and abstracted experiences e .

(1) **Raw trajectories** consist of complete interaction sequences collected from successful agent rollouts, where each trajectory records the full sequence of states, actions, and outcomes encountered during task execution. This form preserves fine-grained behavioral details but often involves long interaction sequences, which may introduce redundant or highly task-specific information.

(2) **Abstracted experiences**, in contrast, are higher-level summaries distilled from both successful and failed trajectories. We employ GPT-5.2 as the experience induction model I to induce concise and reusable experience abstractions from rollout trajectories. Each abstracted experience captures the key conditions, mistakes, or strategies underlying success or failure, producing generalized guidance that is less tied to specific surface forms.

To ensure a fair comparison between different experience types, we fix the capacity of the experience memory across all settings. At inference time, task-relevant experiences are retrieved from the memory based on the given task description and incorporated into the agent’s generation process. Experience retrieval is implemented using a dense retriever built on all-MiniLM-L6-v2, which measures semantic similarity between the current task and stored experience entries. We conduct ex-

periments on both GPT-4o and Qwen3-32B, using ALFWorld as the evaluation environment.

As illustrated in Figure 2, agents leveraging abstracted experiences consistently outperform those using raw trajectory-based experiences as well as agents without experience. Notably, GPT-4o benefits more from abstracted experiences than Qwen3-32B, indicating that stronger models better exploit experience and underscoring the need to enhance experience utilization for smaller models. Motivated by these findings, we adopt abstracted experience induction as the default memory construction strategy for ExpRL.

3.2 Agentic Experience Acquisition

To improve the quality of experience retrieval, ExpRL integrates experience acquisition directly into the agent’s decision-making process. Instead of relying on static or heuristic retrieval strategies, the agent actively acquires experience on demand, conditioned on the current task and interaction state.

Formally, given the current state s_k at interaction step k , the policy π_θ first generates an experience query $q_k \sim \pi_\theta(\cdot | s_k)$, which captures the agent’s information needs at the current decision point. Given the experience query q_k , relevant experiences are retrieved from the memory $\mathcal{M}$ via $e_k = \text{RETRIEVE}(q_k, \mathcal{M})$. Conditioned on the current state s_k , the experience query q_k , and the retrieved experiences e_k , the agent generates an action $a_k \sim \pi_\theta(\cdot | s_k, q_k, e_k)$. This design enables experience acquisition to be optimized end-to-end through task-level rewards. As the policy param-

eters θ are updated via RL, the agent simultaneously learns to generate better actions, formulate more informative experience queries, and more effectively exploit retrieved experiences to guide decision-making. At inference time, the policy adopts the same experience-guided reasoning process as during the training stage. We provide the detailed prompts used for both experience induction and agentic experience acquisition in Appendix C.

3.3 Memory-Policy Co-Evolution

In ExpRL, the memory and the policy are updated in a complementary manner. Specifically, memory is updated in an *off-policy* fashion, while policy optimization follows an *on-policy* training paradigm.

For memory updating, ExpRL maintains a trajectory buffer that caches rollout trajectories generated during RL training. These trajectories are collected on-policy but are not immediately used to update the memory. Instead, after a memory update interval T , the buffered trajectories are processed by the experience induction model, which summarizes raw interaction traces into abstracted experiences. The induced experiences are then used to update the experience memory $\mathcal{M}$, enabling the memory to accumulate reusable knowledge. For policy optimization, ExpRL follows an on-policy RL paradigm and updates the policy using GRPO. Inspired by Jin et al. (2025), we apply loss masking to the experience tokens e , computing the policy gradient objective only over the experience query q and the generated action a . We provide the detailed algorithmic procedure of ExpRL in Algorithm 1, where the newly introduced components beyond standard GRPO are highlighted for clarity.

4 Experiments

In this section, we evaluate ExpRL on three representative agentic benchmarks. We first compare ExpRL with strong baselines to assess its effectiveness in training LLM agents. We then conduct ablation studies to analyze the contribution of agentic experience acquisition in ExpRL. Finally, we assess the generalization of the experience induced by ExpRL, demonstrating its plug-and-play reuse across different models without further training.

4.1 Experiment Setup

Benchmarks. First, we evaluate ExpRL on two widely used and challenging long-horizon agent benchmarks: ALFWorld (Shridhar et al., 2021) and

Algorithm 1: ExpRL

Input: Policy model π_θ ; dataset $\mathcal{D}$;
 experience memory $\mathcal{M}$; experience
 induction model $I(\cdot)$; experience
 memory update interval T

Output: Optimized policy model $\pi_{\hat{\theta}}$

Initialize trajectory buffer $\mathcal{B}_{\text{traj}} \leftarrow \emptyset$;

for step $t = 1, 2, \dots$ **do**

 Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$;

for each state $s_0 \in \mathcal{D}_b$ **do**

for turn $k = 1, 2, \dots$ **do**

 // Experience Acquisition

 Generate experience query

$q_k \sim \pi_\theta(\cdot \mid s_k)$;

 Retrieve experiences

$e_k \leftarrow \text{RETRIEVE}(q_k, \mathcal{M})$;

 Sample action

$a_k \sim \pi_\theta(\cdot \mid s_k, q_k, e_k)$;

 Execute action a_k and observe

 next state s_{k+1} ;

 Compute reward for trajectory τ ;

 Cache $(\tau, r(\tau))$ into $\mathcal{B}_{\text{traj}}$;

 // Policy Update

 Compute advantage and update policy
 model π_θ ;

 // Memory Update

if $t \bmod T = 0$ **then**

 Summarize trajectories into

 experiences $\Delta\mathcal{M} \leftarrow I(\mathcal{B}_{\text{traj}})$;

 Update experience memory $\mathcal{M}$;

 Clear trajectory buffer $\mathcal{B}_{\text{traj}} \leftarrow \emptyset$;

return $\pi_{\hat{\theta}}$;

WebShop (Yao et al., 2022). ALFWorld is an embodied interactive environment that tests an agent’s capability for multi-step decision-making and planning. The environment comprises six categories of household tasks: Pick & Place (Pick), Examine in Light (Look), Clean & Place (Clean), Heat & Place (Heat), Cool & Place (Cool), and Pick Two & Place (Pick2). We report the success rate for each subtask, along with the overall average success rate (All). WebShop is a large-scale web-based environment designed to evaluate LLM agents in realistic online shopping scenarios. The environment contains over 1.1 million products and approximately 12k task instructions. We report the average score and the success rate (SR). Moreover, we evaluate ExpRL in an agentic RAG setting, where the

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	SR
<i>Closed-Source Models</i>									
GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
<i>Qwen2.5-1.5B-Instruct</i>									
Direct	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
PPO	64.8 $\pm$ 3.5	40.5 $\pm$ 6.9	57.1 $\pm$ 4.9	60.6 $\pm$ 6.6	46.4 $\pm$ 4.0	47.4 $\pm$ 1.9	54.4 $\pm$ 3.1	73.8 $\pm$ 3.0	51.5 $\pm$ 2.9
RLOO	88.3 $\pm$ 3.0	52.8 $\pm$ 8.6	71.0 $\pm$ 5.9	62.8 $\pm$ 8.7	66.4 $\pm$ 5.5	56.9 $\pm$ 4.7	69.7 $\pm$ 2.5	73.9 $\pm$ 5.6	52.1 $\pm$ 6.7
GiGPO _{w/ std}	94.4 $\pm$ 5.9	67.5 $\pm$ 4.6	94.8 $\pm$ 3.8	94.4 $\pm$ 7.8	79.8 $\pm$ 4.7	76.4 $\pm$ 5.4	86.7 $\pm$ 1.7	83.1 $\pm$ 1.6	65.0 $\pm$ 3.2
GiGPO _{w/o std}	96.0 $\pm$ 1.4	76.5 $\pm$ 3.9	91.8 $\pm$ 5.5	91.3 $\pm$ 6.3	71.7 $\pm$ 8.4	79.5 $\pm$ 7.7	86.1 $\pm$ 4.7	83.5 $\pm$ 1.8	67.4 $\pm$ 4.5
GRPO	85.3 $\pm$ 1.5	53.7 $\pm$ 8.0	84.5 $\pm$ 6.8	78.2 $\pm$ 7.9	59.7 $\pm$ 5.0	53.5 $\pm$ 5.6	72.8 $\pm$ 3.6	75.8 $\pm$ 3.5	56.8 $\pm$ 3.8
ExpRL	98.7 $\pm$ 2.2	70.7 $\pm$ 7.3	93.7 $\pm$ 2.8	97.8 $\pm$ 3.9	82.5 $\pm$ 8.9	77.7 $\pm$ 1.0	88.5 $\pm$ 2.4	83.7 $\pm$ 0.9	71.1 $\pm$ 0.8
<i>w/o Exp</i>	90.2 $\pm$ 6.0	63.5 $\pm$ 9.5	87.3 $\pm$ 2.2	65.1 $\pm$ 8.4	78.5 $\pm$ 5.2	71.1 $\pm$ 9.6	78.9 $\pm$ 5.6	77.2 $\pm$ 4.4	61.7 $\pm$ 5.1
<i>w/o Aea</i>	81.3 $\pm$ 8.5	49.5 $\pm$ 6.4	82.6 $\pm$ 5.8	70.4 $\pm$ 2.5	41.9 $\pm$ 5.6	31.9 $\pm$ 6.9	60.9 $\pm$ 2.0	73.7 $\pm$ 1.3	49.7 $\pm$ 0.9
<i>Qwen2.5-7B-Instruct</i>									
Direct	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
Mem0*	54.0	55.0	26.9	36.4	20.8	7.69	33.6	23.9	2.0
ExpeL*	21.0	67.0	55.0	52.0	71.0	6.0	46.3	30.9	11.2
MemP*	54.3	38.5	48.1	56.2	32.0	16.7	41.4	25.3	6.4
SimpleMem*	64.5	33.3	20.0	12.5	33.3	3.84	29.7	33.2	8.59
PPO	92.3 $\pm$ 4.0	64.0 $\pm$ 8.4	92.5 $\pm$ 2.4	89.5 $\pm$ 7.0	80.3 $\pm$ 2.0	68.8 $\pm$ 8.3	80.4 $\pm$ 2.7	81.4 $\pm$ 3.1	68.7 $\pm$ 5.1
RLOO	87.6 $\pm$ 4.3	78.2 $\pm$ 8.3	87.3 $\pm$ 5.8	81.3 $\pm$ 7.6	71.9 $\pm$ 5.2	48.9 $\pm$ 8.4	75.5 $\pm$ 4.6	80.3 $\pm$ 3.2	65.7 $\pm$ 4.0
GiGPO _{w/ std}	97.7 $\pm$ 1.6	82.7 $\pm$ 7.9	98.8 $\pm$ 1.6	83.7 $\pm$ 7.2	89.3 $\pm$ 8.2	79.2 $\pm$ 6.6	90.8 $\pm$ 1.3	84.4 $\pm$ 2.9	72.8 $\pm$ 3.2
GiGPO _{w/o std}	91.8 $\pm$ 5.4	88.6 $\pm$ 6.3	95.9 $\pm$ 3.2	90.2 $\pm$ 2.6	86.5 $\pm$ 5.5	85.2 $\pm$ 7.5	90.2 $\pm$ 2.3	86.2 $\pm$ 2.6	75.2 $\pm$ 3.8
GRPO	90.8 $\pm$ 5.1	66.1 $\pm$ 6.7	89.3 $\pm$ 5.4	74.7 $\pm$ 6.9	72.5 $\pm$ 5.4	64.7 $\pm$ 7.3	77.6 $\pm$ 5.2	84.8 $\pm$ 2.8	68.0 $\pm$ 5.9
ExpRL	93.9 $\pm$ 2.6	83.4 $\pm$ 6.7	90.3 $\pm$ 1.8	100.0 $\pm$ 0.0	82.4 $\pm$ 6.8	81.3 $\pm$ 8.2	89.8 $\pm$ 2.3	88.3 $\pm$ 3.9	77.3 $\pm$ 2.1
<i>w/o Exp</i>	88.4 $\pm$ 3.5	84.8 $\pm$ 7.4	87.3 $\pm$ 0.3	71.4 $\pm$ 6.8	65.7 $\pm$ 9.7	56.9 $\pm$ 8.3	77.1 $\pm$ 3.5	79.3 $\pm$ 2.8	66.1 $\pm$ 3.7
<i>w/o Aea</i>	96.9 $\pm$ 5.4	93.5 $\pm$ 5.8	95.3 $\pm$ 4.3	82.9 $\pm$ 8.8	73.1 $\pm$ 11.1	70.6 $\pm$ 8.9	84.6 $\pm$ 1.2	81.9 $\pm$ 2.2	66.7 $\pm$ 2.8

Table 1: Evaluation results on ALFWorld and WebShop. All results are averaged over three random seeds. Results highlighted in blue are obtained from our implementation, results marked with * are from Xia et al. (2026), and the remaining results are reported from Feng et al. (2025).

agent iteratively interacts with search engines to collect evidence and produce final answers. Following prior work (Jin et al., 2025), we consider both single-hop QA datasets (NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)) and multi-hop QA datasets (HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023)). We adopt the exact match (EM) score as the evaluation metric.

Baselines. For the ALFWorld and WebShop benchmarks, we compare ExpRL against two categories of baselines: (1) *Prompt-based methods*, which rely on in-context prompting to guide multi-step behavior without parameter updates, including Direct Prompt, ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023); (2) *Memory-based methods*, which leverage external memory to store

and retrieve historical interactions or distilled experiences for decision making, including Mem0 (Chhikara et al., 2025), Expe L (Zhao et al., 2024), MemP (Fang et al., 2025), and SimpleMem (Hu, 2025); and (3) *RL-based methods*, which optimize agent policies through RL, including PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), and GiGPO (Feng et al., 2025). GiGPO is a state-of-the-art RL method that achieves fine-grained credit assignment for LLM agents. For the agentic RAG, we compare ExpRL with R1-Instruct, Search-R1 (Jin et al., 2025), ZeroSearch (Sun et al., 2025), StepSearch (Wang et al., 2025c), and GiGPO.

Implement Details. We conduct experiments using Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2024) as the base policy models. We follow the same hyperparameter and train-

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2Wiki [*]	MuSiQue [*]	Bamboogle [*]	
R1-Instruct	21.0	44.9	17.1	20.8	27.5	6.0	19.2	22.4
Search-R1	39.3	61.0	39.7	37.0	40.1	14.6	36.8	38.5
ZeroSearch	43.6	61.8	51.5	34.6	35.2	18.4	27.8	39.1
StepSearch	-	-	-	38.6	36.6	22.6	40.0	-
GiGPO	46.4	64.7	46.1	<u>41.6</u>	43.6	<u>18.9</u>	<u>68.9</u>	47.2
ExpRL	<u>45.7</u>	<u>63.6</u>	47.2	42.4	<u>38.6</u>	19.2	71.4	<u>47.0</u>

Table 2: Evaluation results of Qwen2.5-7B-Instruct on agentic RAG tasks. The best and second-best results are highlighted in **bold** and underline, respectively. [†] and ^{*} indicate in-domain and out-of-domain datasets, respectively.

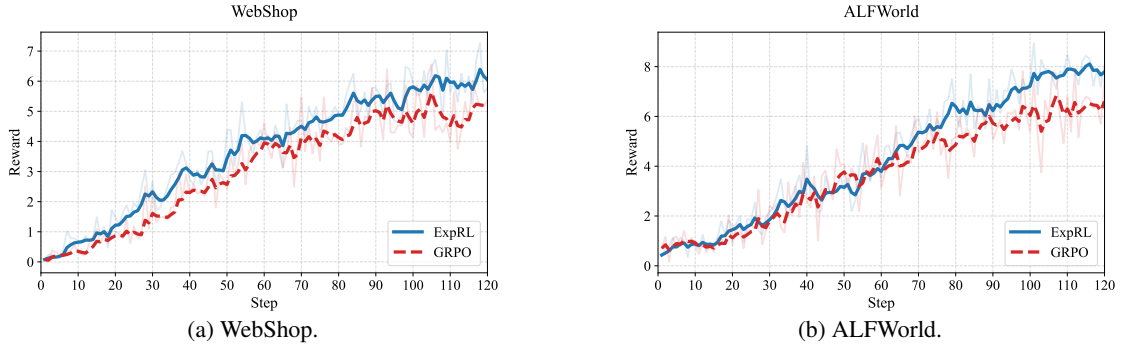


Figure 4: Training reward over iterations on WebShop and ALFWorld with Qwen2.5-1.5B-Instruct.

ing configuration as GiGPO for a fair comparison. Due to computational resource constraints, all experiments are conducted using four NVIDIA A100 GPUs. We set the group size n to 8 for ALFWorld, and 4 for both WebShop and the agentic RAG tasks. We employ GPT-5.2 as the experience induction model and use all-MiniLM-L6-v2 as the experience retriever. For RAG, we use E5 as the retriever.

4.2 Performance on ALFWorld and WebShop

The main results comparing ExpRL with other baseline methods across ALFWorld and WebShop are presented in Table 1. From the results, we make the following key observations: **(1) ExpRL consistently improves the performance upper bound of GRPO.** Across both benchmarks, ExpRL achieves an average performance improvement of **13.0%** over the original GRPO. This substantial gain indicates that injecting abstracted experience into the RL training process effectively enhances the policy’s capability beyond what can be achieved by pure on-policy optimization. Moreover, ExpRL attains performance that is comparable to, and in some cases surpasses, the state-of-the-art method GiGPO, demonstrating the competitiveness of the proposed framework. **(2) ExpRL exhibits higher training efficiency and more stable optimization.** As illustrated in Figure 4, ExpRL converges faster than GRPO and achieves stronger performance with fewer training steps. This suggests that experience-guided interaction improves the quality

of sampled trajectories, providing more informative learning signals and leading to more stable and efficient policy optimization. **(3) ExpRL brings larger gains for smaller base models.** Compared to the 7B model, ExpRL yields more pronounced improvements on the 1.5B model. This observation suggests that smaller models benefit more from explicit experience guidance, as injected experience can effectively compensate for limited intrinsic reasoning and exploration capacity.

4.3 Performance on Agentic RAG

As shown in Table 2, we further evaluate the effectiveness of ExpRL in an agentic RAG setting. Despite the high diversity and complexity of RAG tasks, ExpRL achieves performance that is comparable to GiGPO across all evaluated datasets. This result is particularly noteworthy, as agentic RAG involves heterogeneous queries, dynamic evidence gathering, and multi-step reasoning, posing challenges that differ substantially from embodied or web-based environments. The strong performance of ExpRL in this setting suggests that the experiences induced by our framework capture **general and high-level procedural knowledge**, rather than heuristics that overfit to particular task settings.

4.4 Analysis

Ablation Studies. We conduct ablation experiments to assess the contribution of each component in ExpRL. Specifically, we consider two variants:

Method	ALFWorld						
	Pick	Look	Clean	Heat	Cool	Pick2	All
GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0
w/ Exp	84.9	70.6	37.5	70.0	50.0	42.1	59.5
GPT-5.1	71.7	58.8	28.1	55.0	27.5	34.2	46.0
w/ Exp	84.9	70.6	21.9	80.0	65.0	47.4	62.0

Table 3: Performance on ALFWorld using experience memory constructed by ExpRL.

(1) *w/o Exp*, where the external experience memory is removed entirely; and (2) *w/o Aea*, where gradients are blocked for experience query generation, preventing the policy from learning how to acquire experience. As shown in Table 1, both ablations lead to noticeable performance degradation, demonstrating the effectiveness of incorporating experience memory and jointly optimizing experience acquisition within the RL framework.

Generalization of Experience Memory. We further examine the generality of the experience memory induced by ExpRL. Specifically, we reuse the experience memory constructed during training with smaller policy models and directly apply it to stronger LLMs, including GPT-4o and GPT-5.1, without any additional fine-tuning or adaptation. Table 3 shows that the transferred memory consistently improves performance across different models, indicating that the induced experiences capture high-level, model-agnostic knowledge rather than model-specific behaviors. This finding suggests that ExpRL learns reusable and abstract experience representations that can be effectively leveraged by more capable agents in a plug-and-play manner. To further assess the quality of the induced experiences, we conduct a human evaluation on 100 randomly sampled experiences generated by ExpRL. The results show that 94% of the experiences are judged to be effective and helpful for task execution, demonstrating the high quality and practical usefulness of the induced memory. We provide detailed experience examples in Table 10.

Effect of Experience Induction Models. To study the impact of the experience induction model, we further evaluate WebShop with Qwen2.5-1.5B-Instruct as the base policy and replace GPT-5.2 with Qwen3-Next-80B-A3B-Instruct. As shown in Table 4, experiences summarized by smaller models are often relatively superficial and fail to provide sufficiently effective guidance, which leads to a decline in ExpRL’s performance. Nevertheless, compared with GRPO without any experience, Ex-

Method	Score	SR
GRPO	75.8	56.8
ExpRL (Qwen3-Next-80B-A3B-Instruct)	76.9	60.9
ExpRL (GPT-5.2)	83.7	71.1

Table 4: Comparison of experience induction models on WebShop.

pRL still achieves a 4.1% improvement in success rate on the WebShop dataset.

5 Related Work

Agentic RL. Research has increasingly shifted from static single-turn tasks to agentic settings that require long-horizon interaction with dynamic environments. Agentic RL (Wang et al., 2025b,a; Dong et al., 2025) has emerged as a key paradigm in this shift, enabling models to explore solution spaces through interaction and optimize behavior using environment-driven rewards. By learning through trial and error rather than relying solely on supervision or prompting, agentic RL is well suited for tasks involving planning (Wang et al., 2025b) and tool use (Lu et al., 2025b). Recently, much of the progress (Feng et al., 2025; Zhang et al., 2025b) in agentic RL has focused on addressing reward-related challenges, particularly sparse rewards and fine-grained credit assignment over long horizons.

Experience Replay. Experience replay (Lin, 1992) is a widely adopted technique in RL that stores past transitions in a replay buffer and samples them to improve data efficiency and training stability. Recent studies (Zhan et al., 2025; Lu et al., 2025a; Qin et al., 2025) show that replaying successful trajectories can accelerate convergence and further enhance LLM training stability. Specifically, we treat experience as an explicit input to the agent, rather than replaying raw trajectories.

Experience Learning. Experience (Zhao et al., 2024) can be viewed as a form of memory that captures execution traces, feedback, and other information arising from interaction. For LLM agents, experience has been widely used to guide task execution (Fang et al., 2025; Chen et al., 2025; Chhikara et al., 2025; Wang et al., 2025d; Cai et al., 2025a), especially in complex and tool-driven environments. However, most existing approaches treat experience as a static, external aid and decouple it from the learning process. To address this limitation, we propose ExpRL, which integrates experience knowledge directly into agentic RL, enabling

agents to jointly explore, leverage, and internalize experience during policy optimization.

6 Conclusion

We introduce **ExpRL**, a unified experience learning framework that bridges non-parametric memory and parametric RL for LLM agents. ExpRL enables memory and policy to co-evolve by inducing abstracted experiences from interaction trajectories, actively acquiring and utilizing experience during decision-making, and jointly optimizing experience reuse and policy learning. Through experiments on long-horizon agent benchmarks and agentic RAG tasks, we demonstrate that ExpRL consistently improves performance and learning efficiency over RL baselines. Moreover, we show that the experience memory induced by ExpRL captures high-level, reusable knowledge that generalizes across models and can be directly transferred to stronger LLMs in a plug-and-play manner.

Limitations

While ExpRL demonstrates consistent performance gains by unifying static experience reuse with dynamic policy optimization, it also introduces several limitations. First, agentic experience acquisition requires the policy to actively generate experience queries and incorporate retrieved experiences at each interaction step, which inevitably increases token consumption during both training and inference. Moreover, the effectiveness of ExpRL depends on the quality of the induced experience memory and the retrieval mechanism. Although we observe strong empirical gains, suboptimal experience induction or noisy experience utilization may introduce irrelevant guidance and reduce efficiency. We leave the investigation of more token-efficient experience utilization strategies and adaptive experience acquisition policies as important directions for future work.

Ethical Consideration

This work relies exclusively on publicly available datasets. No private, sensitive, or proprietary data are used, and the study does not raise additional ethical concerns.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. U24A20335 and 62276095) and the Independent Research Project

of the Key Laboratory of Cognition and Decision Intelligence for Complex Systems.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. [Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 12248–12267. Association for Computational Linguistics.
- Anthropic. 2024. [Introducing the next generation of claude](#). Accessed: 2025-04-10.
- Yuxuan Cai, Yipeng Hao, Jie Zhou, Hang Yan, Zhikai Lei, Rui Zhen, Zhenhua Han, Yutao Yang, Junsong Li, Qianjun Pan, Tianyu Huai, Qin Chen, Xin Li, Kai Chen, Bo Zhang, Xipeng Qiu, and Liang He. 2025a. [Building self-evolving agents via experience-driven lifelong learning: A framework and benchmark](#). *CoRR*, abs/2508.19005.
- Yuzheng Cai, Siqi Cai, Yuchen Shi, Zihan Xu, Lichao Chen, Yulei Qin, Xiaoyu Tan, Gang Li, Zongyi Li, Haojia Lin, Yong Mao, Ke Li, and Xing Sun. 2025b. [Training-free group relative policy optimization](#). *CoRR*, abs/2510.08191.
- Silin Chen, Shaoxin Lin, Xiaodong Gu, Yuling Shi, Heng Lian, Longfei Yun, Dong Chen, Weiguo Sun, Lin Cao, and Qianxiang Wang. 2025. [Swe-exp: Experience-driven software issue resolution](#). *CoRR*, abs/2507.23361.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. [Mem0: Building production-ready AI agents with scalable long-term memory](#). *CoRR*, abs/2504.19413.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shiron Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *CoRR*, abs/2501.12948.
- Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2025. [Agentic entropy-balanced policy optimization](#). *CoRR*, abs/2510.14545.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang,

- Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, and 82 others. 2024. [The llama 3 herd of models](#). *CoRR*, abs/2407.21783.
- Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. 2025. [Memp: Exploring agent procedural memory](#). *CoRR*, abs/2508.06433.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. [Group-in-group policy optimization for LLM agent training](#). *CoRR*, abs/2505.10978.
- Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. 2020. [Constructing A multi-hop QA dataset for comprehensive evaluation of reasoning steps](#). In *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pages 6609–6625. International Committee on Computational Linguistics.
- Jian Hu. 2025. [REINFORCE++: A simple and efficient approach for aligning large language models](#). *CoRR*, abs/2501.03262.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. 2024. [Swe-bench: Can language models resolve real-world github issues?](#) In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. [Search-r1: Training llms to reason and leverage search engines with reinforcement learning](#). *CoRR*, abs/2503.09516.
- Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. 2017. [Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1601–1611. Association for Computational Linguistics.
- Tom Kwiattkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur P. Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. [Natural questions: a benchmark for question answering research](#). *Trans. Assoc. Comput. Linguistics*, 7:452–466.
- Jiachun Li, Zhuoran Jin, Tianyi Men, Yupu Hao, Kejian Zhu, Lingshuai Wang, Dongqi Huang, Longxiang Wang, Shengjia Hua, Lu Wang, and 1 others. 2026. [Agentic environment engineering for large language models: A survey of environment modeling, synthesis, evaluation, and application](#). *arXiv preprint arXiv:2606.12191*.
- Long-Ji Lin. 1992. [Self-improving reactive agents based on reinforcement learning, planning and teaching](#). *Machine learning*, 8(3):293–321.
- Zichen Liu, Anya Sims, Keyu Duan, Changyu Chen, Simon Yu, Xiangxin Zhou, Haotian Xu, Shaopan Xiong, Bo Liu, Chenmian Tan, Chuen Yang Beh, Weixun Wang, Hao Zhu, Weiyan Shi, Diyi Yang, Michael Shieh, Yee Whye Teh, Wee Sun Lee, and Min Lin. 2025a. [GEM: A gym for agentic llms](#). *CoRR*, abs/2510.01051.
- Ziru Liu, Cheng Gong, Xinyu Fu, Yaofang Liu, Ran Chen, Shoubo Hu, Suiyun Zhang, Rui Liu, Qingfu Zhang, and Dandan Tu. 2025b. [GHPO: adaptive guidance for stable and efficient LLM reinforcement learning](#). *CoRR*, abs/2507.10628.
- Fanbin Lu, Zhisheng Zhong, Shu Liu, Chi-Wing Fu, and Jiaya Jia. 2025a. [Arpo: end-to-end policy optimization for GUI agents with experience replay](#). *CoRR*, abs/2505.16282.
- Siyuan Lu, Zechuan Wang, Hongxuan Zhang, Qintong Wu, Leilei Gan, Chenyi Zhuang, Jinjie Gu, and Tao Lin. 2025b. [Don't just fine-tune the agent, tune the environment](#). *CoRR*, abs/2510.10197.
- Alex Mullen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. [When not to trust language models: Investigating effectiveness of parametric and non-parametric memories](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A. Smith, and Mike Lewis. 2023. [Measuring and narrowing the compositionality gap in language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, pages 5687–5711. Association for Computational Linguistics.
- Yulei Qin, Xiaoyu Tan, Zhengbao He, Gang Li, Haojia Lin, Zongyi Li, Zihan Xu, Yuchen Shi, Siqi Cai, Renting Rui, Shaoifei Cai, Yuzheng Cai, Xuan Zhang, Sheng Ye, Ke Li, and Xing Sun. 2025. [Learn the ropes, then trust the wins: Self-imitation with progressive exploration for agentic reinforcement learning](#). *CoRR*, abs/2509.22601.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *Preprint*, arXiv:1707.06347.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *CoRR*, abs/2402.03300.

- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. [Reflexion: language agents with verbal reinforcement learning](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. 2021. [Alfworld: Aligning text and embodied environments for interactive learning](#). In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.
- Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. 2025. [Zerosearch: Incentivize the search capability of llms without searching](#). *CoRR*, abs/2505.04588.
- Xiangru Tang, Tianrui Qin, Tianhao Peng, Ziyang Zhou, Daniel Shao, Tingting Du, Xinming Wei, Peng Xia, Fang Wu, He Zhu, Ge Zhang, Jiaheng Liu, Xingyao Wang, Sirui Hong, Chenglin Wu, Hao Cheng, Chi Wang, and Wangchunshu Zhou. 2025. [Agent KB: leveraging cross-domain experience for agentic problem solving](#). *CoRR*, abs/2507.06229.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2022. [9835 musique: Multi-hop questions via single-hop question composition](#). *Trans. Assoc. Comput. Linguistics*, 10:539–554.
- Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Chi Wan, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. 2025a. [VAGEN: reinforcing world model reasoning for multi-turn VLM agents](#). *CoRR*, abs/2510.16907.
- Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025b. [RAGEN: understanding self-evolution in LLM agents via multi-turn reinforcement learning](#). *CoRR*, abs/2504.20073.
- Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. 2025c. [Stepsearch: Igniting llms search ability via step-wise proximal policy optimization](#). *CoRR*, abs/2505.15107.
- Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2025d. [Agent workflow memory](#). In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.
- Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, and 9 others. 2025a. [The rise and potential of large language model based agents: a survey](#). *Sci. China Inf. Sci.*, 68(2).
- Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, and 4 others. 2025b. [Agentgym-rl: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning](#). *CoRR*, abs/2509.08755.
- Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, and 1 others. 2026. [Skillrl: Evolving agents via recursive skill-augmented reinforcement learning](#). *arXiv preprint arXiv:2602.08234*.
- Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. [Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. 2025. [How memory management impacts LLM agents: An empirical study of experience-following behavior](#). *CoRR*, abs/2505.16067.
- Zhenghai Xue, Longtao Zheng, Qian Liu, Yingru Li, Xiaosen Zheng, Zejun Ma, and Bo An. 2025. [Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning](#). *CoRR*, abs/2509.02479.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. [Learning to reason under off-policy guidance](#). *CoRR*, abs/2504.14945.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and

- 22 others. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [Hotpotqa: A dataset for diverse, explainable multi-hop question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 2369–2380. Association for Computational Linguistics.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. [Webshop: Towards scalable real-world web interaction with grounded language agents](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model?](#) *CoRR*, abs/2504.13837.
- Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. 2025. [Exgrpo: Learning to reason from experience](#). *CoRR*, abs/2510.02245.
- Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2025a. [Agent learning via early experience](#). *CoRR*, abs/2510.08558.
- Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2025b. [RLVMR: reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents](#). *CoRR*, abs/2507.22844.
- Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. [Expel: Llm agents are experiential learners](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.
- Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. 2025. [Memento: Fine-tuning LLM agents without fine-tuning llms](#). *CoRR*, abs/2508.16153.
- 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. 2024. [Webarena: A realistic web environment for building autonomous agents](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

A Computational Cost Analysis

A.1 Token Consumption

To quantify the token consumption of ExpRL, we select Qwen2.5-7B-Instruct as the base model. We perform rollouts on the 100 samples from the WebShop test set and conduct inference for both ExpRL and GRPO. This process involves only forward generation and environment interaction, excluding any training, parameter updates, or experience induction. For each episode, we record the numbers of input and output tokens and average them across all episodes. The comparison of token consumption between ExpRL and GRPO during inference is summarized in Table 5, which reports the results of these trials.

Method	Input Tokens	Output Tokens	Total Tokens
GRPO	15891	7359	23250
ExpRL	22533	13229	35762

Table 5: Token consumption of GRPO and ExpRL.

A.2 Training Time

ExpRL introduces additional engineering complexity on top of the verl framework. Under the same experimental setting, ExpRL requires approximately $1.5\times$ the training time of GRPO. The additional overhead mainly comes from the rollout process for experience retrieval queries and the corresponding gradient updates. By contrast, the overhead introduced by memory maintenance is negligible in practice, since the experience memory contains only 40–60 entries, making both memory updates and vector-based retrieval efficient. Overall, the increased training cost of ExpRL mainly stems from experience acquisition and optimization during training, rather than from maintaining the external memory itself.

B Supplementary Experiments

B.1 Experience Internalization

To further analyze whether ExpRL can internalize part of the acquired experience into the model parameters during training, we conduct an additional experiment on WebShop. Specifically, we

use Qwen2.5-1.5B-Instruct as the base model and first train it under the ExpRL framework up to the 90th epoch, where the success rate reaches 52.3%. We then remove the experience retrieval module from both the training and the test phases and continue training until convergence. The results are shown in Table 6.

The results show that ExpRL (internalized) still significantly outperforms GRPO, improving the success rate from 56.8% to 67.7%. This suggests that part of the acquired experience can be internalized into the model parameters during ExpRL training. However, ExpRL (internalized) remains slightly behind the full ExpRL model, indicating that explicit experience retrieval continues to provide additional performance gains.

Method	Score	SR
GRPO	75.8	56.8
ExpRL (internalized)	83.3	67.7
ExpRL	83.7	71.1

Table 6: Effect of experience internalization on WebShop.

B.2 Effect of Agentic Experience Acquisition

We conducted a comparative experiment on WebShop using Qwen2.5-1.5B-Instruct, where the task description itself is used for experience retrieval rather than allowing the model to actively generate experience queries. The results are shown in Table 7. Directly using the task description for retrieval cannot adapt to the evolving interaction state, making it difficult to retrieve experiences that precisely match the agent’s current decision needs. This leads to lower success rates compared to ExpRL, further demonstrating the effectiveness of agentic experience acquisition.

Method	Score	SR
GRPO	75.8	56.8
ExpRL (Task)	78.9	60.9
ExpRL	83.7	71.1

Table 7: Comparison of task-based retrieval and agentic experience acquisition.

C Implement Details

We provide prompt templates for experience summarization and query generation in both WebShop and ALFWorld in Tables 8 and 9. Our implemen-

tation is developed based on the verl-agent¹. Additionally, to mitigate the drift of the main policy induced by frequent query optimization, we decouple the query stage and apply actor updates only at periodic epochs instead of performing gradient computation on query outputs each epoch.

To further analyze the effect of experience memory size on ExpRL, we experimented with memory sizes of 25, 50, 100, and 150 during training on WebShop and ALFWorld. The results show that ExpRL is not sensitive to the memory size. We attribute this mainly to the relatively limited task types in WebShop and ALFWorld, and to the fact that ExpRL stores abstract and reusable experiences rather than complete trajectories. As a result, a small experience memory is already sufficient to cover most common decision-making patterns.

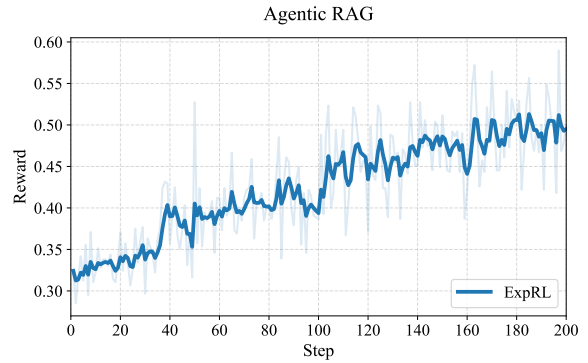


Figure 5: Training reward over iterations on RAG with Qwen2.5-7B-Instruct.

D Experience Examples

We provide abstracted experience examples of ExpRL in Table 10. For each task instance, we show the original instruction, a brief summary, and the corresponding generalizable experience distilled from the trajectory, which captures reusable task-solving principles rather than instance-specific examples.

E Case Study

We present case studies to illustrate how abstracted experiences guide agent behavior in practice. As shown in Figure 6, experience enables the agent to make better decisions by avoiding meaningless actions and unproductive loops.

¹<https://github.com/langfengQ/verl-agent/>

Prompt for Experience Summarization

SYSTEM PROMPT: You are an expert autonomous agent operating in the WebShop e-commerce environment. Given one or more shopping trajectories, your goal is to extract 1~5 short but clear and generalizable lessons. Do not mention any specific product information.

USER PROMPT: You are an expert autonomous agent operating in the WebShop e-commerce environment. Given one or more shopping trajectories, your goal is to extract 1~5 short but clear and generalizable experiences. Please carefully examine the given trajectory. If you believe it contains valuable insights, summarize 1~5 concise and general experiences from it. If you believe it contains no valuable insights, return an empty experiences list in the required JSON format.

Follow these steps:

1. First, provide a brief summary of the trajectory, no more than 15 words.
2. Derive 1~5 reusable, general experiences from the trajectory to guide the model in completing similar tasks more effectively. The experiences should be concise.
 - Do NOT mention any brands, product names, IDs/ASINs, prices, colors, sizes, or exact attribute values.
 - Experiences should be decision-making heuristics (general rules), not a narration of actions.
3. The experiences should only involve observable trajectory behaviors and outcomes; do not summarize or use any information related to `action_valid = false`.

Do NOT output your reasoning. Only follow the steps internally. Then output your final result **only in JSON format** as shown below:

```
"""json
{{
  "summary": "...",
  "experiences": [
    "experience_1",
    "experience_2",
    "experience_3"
  ]
}}
```

Input: {trajectories}

Prompt for Query Generation

USER PROMPT: You are an expert autonomous agent operating in the WebShop e-commerce environment.

Your task is to: {task_description}

Your current observation is: {current_observation}

You need to generate a retrieval query based on the task to recall experiential knowledge to help you complete the task more effectively. Follow the steps exactly:

1. Inferring the decision-level state from the current observation in ONE short sentence (≤ 15 words).
 - The decision-level state should describe logical progress (e.g., initial search or broad results or close candidates but missing constraints).
 - Do NOT mention UI elements, page controls, product names, or IDs.
2. The retrieval query consists ONLY of the given task description followed by the decision-level state phrase.
3. Wrap the retrieval query with `<query_experience>` tags.

Output Structure:

```
<query_experience> The task is to [task_description]. The current state is [decision-level state].
</query_experience>
```

Table 8: Prompt templates for experience summarization and query generation in WebShop environment.

Prompt for Experience Summarization

SYSTEM PROMPT: You are an expert agent operating in the ALFRED Embodied Environment. Given task trajectories, your goal is to extract 1~5 short but clear and generalizable lessons about effective planning, reasoning, and interaction strategies. Do not mention any specific object names or room layouts.

USER PROMPT: An Agent system interacted with the ALFWorld environment, producing several trajectories that include both successful and failed cases. Please carefully examine the given trajectory. If you believe it contains valuable insights, summarize 1~5 concise and general experiences from it. If you believe it contains no valuable insights, return an empty experiences list in the required JSON format. A trajectory is considered to contain valuable insights only if it reveals a reusable decision pattern that affects task success or failure.

Follow these steps:

1. First, provide a brief summary of the trajectory, no more than 10 words.
2. Derive 1~5 reusable, general experiences from the trajectory to guide the model in completing similar tasks more effectively. The experiences should be concise.
 - Do NOT mention specific object IDs or room names.
 - Experiences should be decision-making heuristics (general rules), not a narration of actions.
 - Each experience should be directly applicable to guiding a future decision in a similar ALFWorld task.
3. The experiences should only involve observable trajectory behaviors and outcomes; do not summarize or use any information related to `action_valid = false`.

Do NOT output your reasoning. Only follow the steps internally. Then output your final result **only in JSON format** as shown below:

```
"""json
{{
  "summary": "...",
  "experiences": [
    "experience_1",
    "experience_2",
    "experience_3"
  ]
}}
```

Input: {trajectories}

Prompt for Query Generation

USER PROMPT: You are an expert agent operating in the ALFRED Embodied Environment.

Your task is to: {task_description}

Your current observation is: {current_observation}

You need to generate a retrieval query based on the task to recall experiential knowledge to help you complete the task more effectively. Follow the steps exactly:

1. Inferring the decision-level progress from the current observation in ONE short phrase (≤ 15 words).
 - Describe which subgoal stage you are in (e.g., locating target, seeking tool, returning to target, ready to interact).
 - Do NOT mention specific object IDs or room names.
2. The retrieval query consists ONLY of the given task description followed by the decision-level progress .
3. Wrap the retrieval query with <query_experience> tags.

Output Structure:

<query_experience> The task is to [task_description]. The current state is [decision-level progress].</query_experience>

Table 9: Prompt templates for experience summarization and query generation in ALFWorld environment.

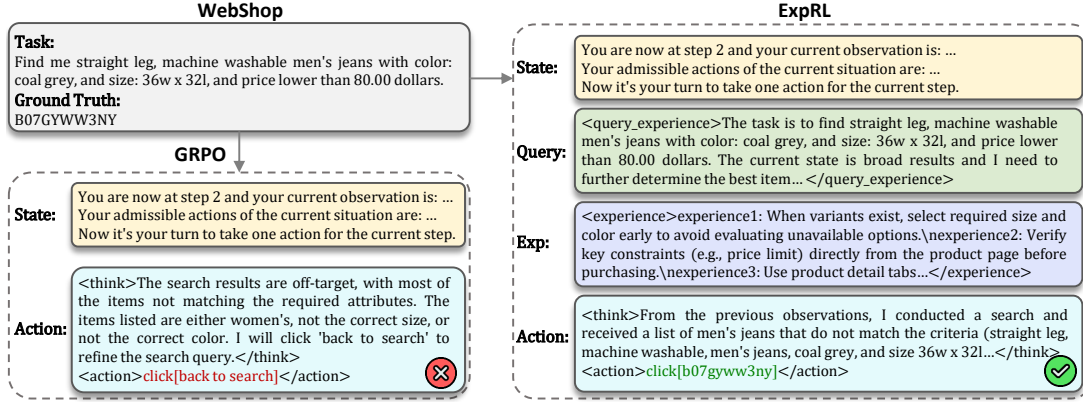


Figure 6: Case study of GRPO and ExpRL on the WebShop.

Dataset	Instruction	Generalizable Experience
ALFWorld	Task: put a clean kettle in diningtable. Summary: found kettle on a surface, cleaned it, then placed it.	(1) Before searching containers, quickly check obvious surfaces for the target item. (2) If the goal specifies an item state (e.g., clean), explicitly perform the state-changing action before placing it. (3) After completing a required transformation, immediately go to the destination and place the item to finish the task.
ALFWorld	Task: heat some egg and put it in fridge. Summary: heated an egg in a microwave, then stored it in the fridge.	(1) Check nearby open surfaces first to locate target items quickly. (2) For heating tasks, insert the item into the appliance and satisfy required preconditions (e.g., closing it if needed) before heating. (3) If the appliance contains other items, avoid removing them unless they block the required operation. (4) After completing the transformation (heating), immediately proceed to the final placement goal (open the target container, then move the item in). (5) Use the environment's available action list to confirm preconditions before executing key actions.
WebShop	Task: find non-slip, easy-install ottomans made of faux leather, size 80×40×40 cm, price < \$170. Summary: searched, opened an item, selected the required size variant, then purchased.	(1) Use targeted search queries including key material and hard constraints to narrow results quickly. (2) Open promising results to verify constraints on the product page rather than relying on listing titles. (3) When a required attribute is a variant option, explicitly select the correct variant before purchasing. (4) Confirm the displayed price on the product page meets the budget constraint before checkout.
WebShop	Task: find men's loafers & slip-ons with rubber outsole/sole, color yellow-a, size 9.5, price < \$40. Summary: searched, opened a listing, selected required variants, and purchased.	(1) Start with a search query that includes key constraints to reduce irrelevant results. (2) From results pages, open promising items and use on-page variant selectors to match required attributes. (3) After selecting required variants, confirm the displayed price range still satisfies the budget constraint before buying. (4) Proceed to purchase only once all mandatory constraints are visibly selected on the product page.
RAG	verification strategy for comparison of two entities shared attribute evidence.	For each entity, retrieve a reliable source that explicitly states the target attribute, then compare; if either side lacks explicit evidence, report insufficient support rather than guessing.
RAG	temporal verification strategy for current attribute of an entity across sources.	Prefer the most recent explicitly dated source for the target attribute and cross-check multiple sources for recency; if no dated confirmation exists, ask for a timeframe or report uncertainty rather than guessing.

Table 10: Abstracted experience examples of ExpRL.

F Reasoning Behavior

We next present the complete trajectory of multi-turn interaction between agent and environment, and demonstrate the reasoning behavior of LLM agents (Qwen2.5-7B-Instruct) trained by ExpRL.

F.1 ALFWorld

Environment (Step 1)

-- Welcome to TextWorld, ALFRED! --
Your task is to: heat some egg and put it in fridge. You are in the middle of a room. Looking quickly around you, you see a cabinet 12, a cabinet 11, a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 1, a drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.

Agent (Step 1)

```
<think>  
To accomplish the task of heating an egg and putting it in the fridge, I need to find an egg first. Since the most likely places to find an egg are the cabinets, countertops, and fridge, I will start by checking the cabinets and countertops.  
</think>  
<action>go to countertop 1</action>
```

Environment (Step 2)

You arrive at countertop 1. On the countertop 1, you see a egg 1, and a winebottle 1.

Agent (Step 2)

```
<think>  
I am currently at countertop 1 and see an egg 1 and a winebottle 1. Since my task is to heat an egg and put it in the fridge, I need to take the egg first. The egg is already on the countertop, so I can take it directly.  
</think>  
<action>take egg 1 from countertop 1</action>
```

Environment (Step 3)

You pick up the egg 1 from the countertop 1.

Agent (Step 3)

```
<think>  
The task is to heat some egg and put it in fridge. So far, I have the egg and need to heat it. The microwave seems the most suitable option in the current setup for heating an egg. After heating, I should then put it in the fridge.  
</think>  
<action>go to microwave 1</action>
```

Environment (Step 4)

You arrive at microwave 1. The microwave 1 is closed.

Agent (Step 4)

```
<think>  
I need to heat the egg and then put it in the fridge. The microwave is closed, so I should first open it to check if it's functional and to heat the egg properly. After that, I can proceed to put the heated egg in the fridge.  
</think>  
<action>open microwave 1</action>
```

Environment (Step 5)

You open the microwave 1. The microwave 1 is open. In it, you see a apple 2.

Agent (Step 5)

<think>

1. The current observation shows that the microwave is open and contains an apple 2.
2. My goal is to heat the egg and put it in the fridge.
3. The action 'heat egg 1 with microwave 1' is in the list of admissible actions, but it requires the egg to be inside the microwave.
4. The most logical action is to move the egg 1 into the microwave 1 since it is already open.

</think>

<action>move egg 1 to microwave 1</action>

Environment (Step 6)

You move the egg 1 to the microwave 1.

Agent (Step 6)

<think>

I have moved the egg to the microwave, and now I need to heat it. The next logical step is to close the microwave after placing the egg inside. This will ensure that it is properly sealed to prevent any spills or issues during heating. Once the egg is heated, I will need to cool it down before placing it in the fridge.

</think>

<action>close microwave 1</action>

Environment (Step 7)

You close the microwave 1.

Agent (Step 7)

<think>

1. I have an egg in the microwave 1, and I have just closed it. The next step is to heat the egg.
2. Since the microwave is already closed, I should check if it is suitable to heat the egg now.
3. The admissible actions include examining the microwave and opening it again, but I need to confirm if it's safe to heat the egg first.
4. Considering that I need to heat the egg and put it in the fridge later, I need to ensure the microwave is functioning correctly.
5. I should examine the microwave to check its current state and condition.

</think>

<action>examine microwave 1</action>

Environment (Step 8)

The microwave 1 is closed.

Agent (Step 8)

<think>

I need to heat the egg and then put it in the fridge. The microwave is closed and I need to open it to heat the egg. The action 'open microwave 1' is available, so I should perform this action to open the microwave and heat the egg.

</think>

<action>open microwave 1</action>

Environment (Step 9)

You open the microwave 1. The microwave 1 is open. In it, you see a apple 2, and a egg 1.

Agent (Step 9)

`<think>`

I need to heat the egg and then put it in the fridge. The egg is in the microwave, and the microwave is currently open. I need to take the egg out, heat it, and then put it back in the fridge.

The first step is to take the egg out of the microwave. The admissible actions include "take egg 1 from microwave 1", which is a valid option.

`</think>`

`<action>take egg 1 from microwave 1</action>`

Environment (Step 10)

You pick up the egg 1 from the microwave 1.

Agent (Step 10)

`<think>`

Currently, I have the egg 1 in my hand. The next step is to heat the egg and then put it in the fridge. Given the list of admissible actions, the most appropriate actions to proceed are:

1. Heat egg 1 with microwave 1 (since I already have the egg and microwave 1 is open)
2. Go to fridge 1 (to put the heated egg in the fridge after it is heated).

I should start by heating the egg in the microwave.

`</think>`

`<action>heat egg 1 with microwave 1</action>`

Environment (Step 11)

You heat the egg 1 using the microwave 1.

Agent (Step 11)

`<think>`

The task is to heat the egg and put it in the fridge. After heating the egg in the microwave, I need to put it in the fridge. The fridge should be the most appropriate place to store a heated egg.

Looking at the available actions, the option "go to fridge 1" is available, which seems to be the most relevant action to complete the task.

`</think>`

`<action>go to fridge 1</action>`

Environment (Step 12)

You arrive at fridge 1. The fridge 1 is closed.

Agent (Step 12)

`<think>`

The task is to heat an egg and put it in the fridge. I have already heated the egg using the microwave, but now I need to put it in the fridge. The current observation shows that fridge 1 is closed. The most appropriate action would be to open the fridge 1 so that I can place the heated egg inside.

`</think>`

`<action>open fridge 1</action>`

Environment (Step 13)

You open the fridge 1. The fridge 1 is open. In it, you see a apple 1, a mug 1, a tomato 2, and a tomato 1.

Agent (Step 13)

```
<think>
I need to heat the egg and then put it in the fridge. The egg is currently the "egg 1" that I had
previously heated in the microwave. Since the fridge is already open, I can directly place the
heated egg inside it. The action "cool egg 1 with fridge 1" is not suitable because that would
involve refrigerating the egg further, which is not necessary at this point. The correct action
would be to move the egg 1 to the fridge 1.
</think>
<action>move egg 1 to fridge 1</action>
```

F.2 Webshop

Environment (Step 1)

Instruction: Find me straight leg, machine washable men's jeans with color: coal grey, and size: 36w x 32l, and price lower than 80.00 dollars. Your current observation is: 'Search'.

Agent (Step 1)

```
<think>
To find straight leg, machine washable men's jeans with color: coal grey, and size: 36w x 32l,
and price lower than 80.00 dollars, I need to start my query with the most critical attributes to
narrow down the search effectively. The query should specifically look for these attributes and
filter the results based on the price limit.
The query should be something like: "straight leg machine washable men's jeans coal grey 36w x
32l price<80.00" to find the correct product and filter the results based on the price limit.
</think>
<action>search[straight leg machine washable men's jeans coal grey 36w x 32l price<80.00]</action>
```

Environment (Step 2)

'Back to Search'
'Page 1 (Total results: 50)'
'Next >'
'B07GYWW3NY': 'J. Crew - Men's - Sutton Straight-Fit Flex Chino (Multiple Size/Color Options)', '\$59.5'
'B01MSP2B0I': 'Wrangler Men's Retro Slim Fit Straight Leg Jean, Black, 42W x 32L', '\$63.05'
'B08LKKSL8F': 'BOCOMAL FR Pants for Men Cargo Flame Resistant Pants(2112&CAT2) 100% C 7.5oz Utility Fire Resistant Pants', '\$45.99 to \$49.99'
'B09QVCYVY': 'Burband Womens High Waist Bike Shorts Tummy Control Workout Yoga Pants Running Exercise Running Shorts with Side Pockets', '\$6.82'
'B099231V35': 'Mens Casual Cargo Pants Hi Vis Viz Reflective Overalls High Visibility Safe Work Pants Outdoor Hiking Trousers Big and Tall', '\$16.79'
'B09PBPZ24Z': 'myhehthw Women's High Waisted Jeans for Women Distressed Ripped Jeans Slim Fit Butt Lifting Skinny Stretch Jeans Trousers', '\$22.99 to \$25.99'
'B09C1YWG8W': 'Soft Plush Electric Heated Blanket Throw with Foot Pocket | Navy Blue 50 x 62 | 3 Heat Settings with 2 Hour Auto Shut Off, UL Certified | Machine Washable', '\$69.99'
'B09M63B87V': 'YALFJV Women Long Sleeve Crew Neck Side Button T Shirts Tunic Dress Loose Asymmetric Hem Tunic Pullover to Wear with Leggings', '\$10.71 to \$18.34'
'B07BB1FJV8': 'Edna Wig Color FS4/30 - Foxy Silver Wigs Short Pixie Wispy Fringe Synthetic Straight Layers African American Women's Machine Wefted Lightweight Average Cap Bundle with MaxWigs Hairloss Booklet', '\$36.69'
'B09NNMV9LN': 'JSPOYOU Mens Short Sleeve Crewneck 3D Graphic Tunic Shirts Big & Tall Tie Dye Summer Top Basic Designed Classic Cotton Shirt', '\$3.99 to \$10.99'

Agent (Step 2)

```
<think>
From the previous observations, I conducted a search and received a list of men's jeans that do
not match the criteria (straight leg, machine washable, men's jeans, coal grey, and size 36w x
32l with a price lower than $80.00). The current list includes various types of pants and jeans
but does not filter the results based on the specific criteria. To find the correct product, I
need to click through the list and select the appropriate item based on the criteria.
</think>
<action>click[b07gyww3ny]</action>
```

Environment (Step 3)

'Back to Search'
'< Prev'
'size': '32w x 30l', '32w x 32l', '34w x 30l', '34w x 32l', '36w x 30l', '36w x 32l'
'color': 'charcoal dust', 'coal grey', 'navy', 'overcast blue', 'khaki'
'J. Crew - Men's - Sutton Straight-Fit Flex Chino (Multiple Size/Color Options)'
'Price: \$59.5' [SEP] 'Rating: N.A.', 'Description', 'Features', 'Reviews'
'Buy Now'.

Agent (Step 3)

<think>
From the previous observations, I have already performed a search and received a list of men's jeans that do not match the criteria (straight leg, machine washable, men's jeans, coal grey, and size 36w x 32l, with a price lower than \$80.00). The current list of filters includes various sizes and colors, but not the specific item that I need. To find the correct item, I need to click the "coal grey" filter to narrow down the results and find the appropriate product.
</think>
<action>click[coal grey]</action>

Environment (Step 4)

'Back to Search'
'< Prev'
'size': '32w x 30l', '32w x 32l', '34w x 30l', '34w x 32l', '36w x 30l', '36w x 32l'
'color': 'charcoal dust', 'coal grey', 'navy', 'overcast blue', 'khaki'
'J. Crew - Men's - Sutton Straight-Fit Flex Chino (Multiple Size/Color Options)'
'Price: \$59.5' [SEP] 'Rating: N.A.', 'Description', 'Features', 'Reviews'
'Buy Now'.

Agent (Step 4)

<think>
From the previous observations, I have already selected the color coal grey for the men's jeans. The next step is to find the correct size, which should be "36w x 32l" based on the requirements. To do this, I need to click the corresponding size option on the page.
</think>
<action>click[36w x 32l]</action>

Environment (Step 5)

'Back to Search'
'< Prev'
'size': '32w x 30l', '32w x 32l', '34w x 30l', '34w x 32l', '36w x 30l', '36w x 32l'
'color': 'charcoal dust', 'coal grey', 'navy', 'overcast blue', 'khaki'
'J. Crew - Men's - Sutton Straight-Fit Flex Chino (Multiple Size/Color Options)'
'Price: \$59.5' [SEP] 'Rating: N.A.', 'Description', 'Features', 'Reviews'
'Buy Now'.

Agent (Step 5)

<think>
From the previous observations, I have already selected the correct color (coal grey) and size (36w x 32l) for the men's jeans. The next step is to verify the price to ensure it is lower than \$80.00. However, the price is currently displayed as \$59.5, which is well below the threshold. Therefore, the next action is to click the "Buy Now" button to proceed with the purchase.
</think>
<action>click[buy now]</action>

E3 RAG

Environment (Step 1)

Are Fir and Asystasia both coniferous trees?

Agent (Step 1)

<think>

To answer this question, I need to determine if Fir and Asystasia are both coniferous trees. Fir is indeed a coniferous tree. However, I need to check the nature of Asystasia to be sure.

</think>

<search>what type of plant is Asystasia</search>

Environment (Step 2)

<information> {"result": "Doc 1: "Asystasia gangetica" Asystasia gangetica Asystasia gangetica is a species of plant in the Acanthaceae family. It is commonly known as the Chinese violet, coromandel or creeping foxglove. In South Africa this plant may simply be called asystasia. This plant is a spreading herb or groundcover, reaching 600 mm in height or up to 1 m if supported. The stems root easily at the nodes. The leaves are simple and opposite. The fruit is an explosive capsule which starts out green in colour, but dries to brown after opening. Widespread throughout the Old World Tropics, and introduced into tropical Americas and Hawaii, where Doc 2: "Asystasia alba" Asystasia alba Asystasia alba is a species of tropical herb in the Acanthaceae family. It is endemic to Christmas Island, an Australian territory in the north-eastern Indian Ocean. Its specific epithet comes from the Latin "alba" (white), referring to the colour of its flowers. "Asystasia alba" is an erect woody herb, growing to 0.5–0.75 m in height. Its 30–140 mm long leaves are ovate, acuminate or acute, pale green in colour and usually bristly when young. The inflorescence is 60–80 mm long, the flowers single or occasionally paired, the bracts and bracteoles about 2 mm long and the pedicels 1.5–3 Doc 3: "Asystasia alba" mm long. The corolla is white or violet, and the tube 14–18 mm long. The capsule is usually 2-seeded and about 26 mm long. The seeds are 4 mm long. "Asystasia alba" is a variable species; it forms part of a species complex that includes "A. australasica" and "A. oppositiflora", but differs from both of them in having single flowers at each axil. Found only on Christmas Island, the plant occurs in cleared areas on the island's terraces up to about 200 m above sea level. Asystasia alba Asystasia alba is a species of tropical herb in the Acanthaceae family."}</information>

Agent (Step 2)

<think>

From the information provided, Asystasia is not a coniferous tree. It belongs to the Acanthaceae family and is described as a herb or groundcover. Now, I need to find information about Fir trees to compare.

</think>

<search>what type of tree is Fir</search>

Environment (Step 3)

<information> {"result": "Doc 1: Fir. Fir Firs (Abies) are a genus of 48-56 species of evergreen coniferous trees in the family Pinaceae. They are found through much of North and Central America, Europe, Asia, and North Africa, occurring in mountains over most of the range. Firs are most closely related to the genus "Cedrus" (cedar). Douglas firs are not true firs, being of the genus "Pseudotsuga". They are large trees, reaching heights of tall with trunk diameters of when mature. Firs can be distinguished from other members of the pine family by the way in which their needle-like leaves are attached singly to the branches Doc 2: Fir species, including "Chionodes abella" (recorded on white fir), autumnal moth, conifer swift (a pest of balsam fir), the engrailed, grey pug, mottled umber, pine beauty and the tortrix moths "Cydia illutana" (whose caterpillars are recorded to feed on European silver fir cone scales) and "C. duplicana" (on European silver fir bark around injuries or canker). "Abies spectabilis" or Talispatra is used in Ayurveda as an antitussive (cough suppressant) drug. Philips, Roger. "Trees of North America and Europe", Random House, Inc., New York , 1979. Fir Firs (Abies) are a genus of 48-56 species of evergreen coniferous trees in the family Doc 3: "Fir wave" Fir wave A fir wave is a set of alternating bands of fir trees in sequential stages of development, observed in forests on exposed mountain slopes in several areas, including northeastern North America and Japan. Fir waves develop by wave-regeneration following wind disturbance, and is one of various types of patterned vegetation. Fir waves form by the ecological process of wave-regeneration. When a tree falls, a gap in the canopy is formed. This exposes trees at the leeward edge of the gap to greater wind. These trees are thus more likely to die from damage and desiccation than windward trees."}</information>

Agent (Step 3)

<think>

From the information provided in the previous steps, we know:- Asystasia is a genus of plants in the family Acanthaceae. It is not a coniferous tree, but a herb or groundcover. Fir (Abies) is a genus of evergreen coniferous trees in the family Pinaceae. It is a coniferous tree. Therefore, Fir is a coniferous tree, but Asystasia is not.

</think>

<answer>No</answer>