RETHINKING EXPERIENCE UTILIZATION IN SELF-EVOLVING LANGUAGE MODEL AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-evolving agents improve by accumulating and reusing experience from past interactions. Existing work has largely focused on how experience is constructed, represented, and updated, while paying less attention to how experience should be used during runtime decision-making. As a result, most agents rely on rigid usage strategies, either injecting experience once at initialization or at every step, without considering whether it is needed for the current decision. This paper studies experience utilization as a critical design dimension of self-evolving agents. We ask whether agents benefit from interweaving experience use with decision-making, so that experience is invoked only when additional guidance is needed. To examine this question, we introduce ExpWeaver, a lightweight instantiation that leaves experience construction unchanged and modifies only runtime utilization by exposing experience as an optional resource during reasoning. Across four representative frameworks, seven LLM backbones, and three types of environments, ExpWeaver consistently achieves the best performance among different utilization strategies. Reinforcement learning experiments further show that this behavior can be amplified through training. Usage-pattern, causal ablation, and entropy-based analyses reveal that ExpWeaver enables agents to invoke experience selectively, at beneficial decision points, and under higher reasoning uncertainty. Overall, our findings call for a shift from merely studying *what* experience to store toward understanding *how* and *when* experience should enter decision-making.

1 INTRODUCTION

The emergence of self-evolving agents marks a critical milestone in the development of autonomous systems capable of continuous learning and adaptive behavior (Zhao et al., 2024b; Dou et al., 2025; Fang et al., 2025). As we move into the era of experience (Silver & Sutton, 2025), where learning is increasingly driven by agents' own interaction histories, this paradigm offers substantial potential for building systems that operate robustly in open-ended environments (Hu, 2025; Lin et al., 2026).

At a high level, self-evolving agents operate by continuously gathering, storing, and reusing experience from interactions with the environment to inform future decisions (Gao et al., 2025; Cai et al., 2025; Bell et al., 2025). This process can be naturally decomposed into two stages: constructing an experience repository and utilizing it during decision-making. Existing research has devoted substantial attention to the former, with extensive efforts devoted to designing diverse representations of experience, such as distilled insights (Zhang et al., 2025a; Ouyang et al., 2025), reusable skills (Wang et al., 2024; Xia et al., 2026), or executable workflows (Wang et al., 2025). In comparison, experience utilization has received less systematic study as an independent design dimension.

In practice, many self-evolving agent frameworks adopt fixed experience usage schemes, as conceptually illustrated in Figure 1(a) and (b). Some methods follow an *initialization-only* strategy, where experience is injected once at the beginning of task execution (Zhao et al., 2024a); others adopt an *always-on* strategy, where retrieved experience is injected at every decision step (Wang et al., 2025). Neither explicitly adapts experience usage to the current decision. This motivates our central question: *what role should experience play within the agent's runtime decision-making process?*

This motivates us to revisit the role of experience in self-evolving agents. In human cognition, experience is not used as static background knowledge, but is tightly coupled with decision-making:

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.

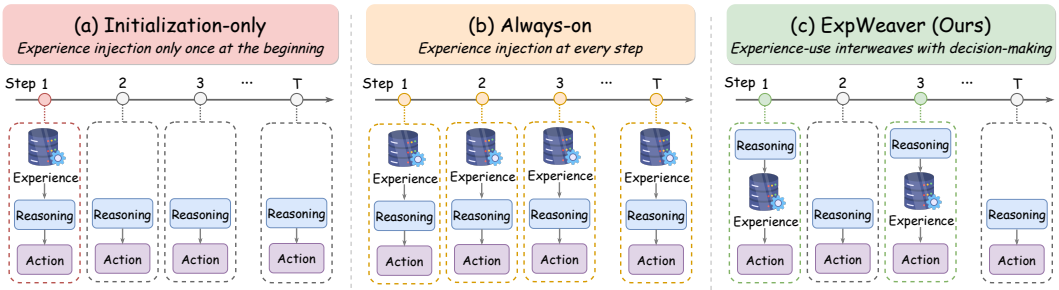


Figure 1: Illustration of different experience utilization strategies in self-evolving agents. (a) **Initialization-only**: experience is injected once at the beginning and remains static throughout the decision process. (b) **Always-on**: experience is injected at every step. (c) **ExpWeaver**: experience utilization is interwoven with the decision-making process, allowing the agent to decide during reasoning *when* to retrieve and use experience.

individuals selectively apply relevant memories when facing uncertainty, ambiguity, or increased cognitive control demands (Shiffrin & Schneider, 1977; Botvinick et al., 2001). Cognitive neuroscience further suggests that memory retrieval is actively regulated based on task demands, with the prefrontal cortex gating access to stored knowledge during goal-directed behavior (Tomita et al., 1999; Anderson et al., 2004; Bausch et al., 2026). Inspired by this mechanism, we hypothesize that: *if agents can similarly couple experience utilization with their decision-making process, enabling experience to be invoked only when needed, they may make more effective use of accumulated experience.*

To examine this hypothesis, we propose **ExpWeaver**, a lightweight paradigm that interweaves experience utilization into the agent's decision-making process. Rather than introducing on-demand retrieval itself, ExpWeaver serves as a controlled instantiation for studying experience utilization. ExpWeaver is not designed to change how experience is constructed, represented, or updated. Instead, it provides a minimal intervention on the usage side: experience is made available as an optional resource during reasoning. Concretely, ExpWeaver extends the widely adopted ReAct-style (Yao et al., 2023) decision-making paradigm in self-evolving agents, where decisions are made through iterative `reasoning` $\rightarrow$ `action` steps. As illustrated in Figure 1(c), ExpWeaver introduces a unified loop: `reasoning` $\rightarrow$ `(optional experience utilization)` $\rightarrow$ `action`. In this loop, the agent may retrieve relevant experience only when it determines during reasoning that additional guidance is needed.

This design allows us to systematically isolate the effect of experience utilization from experience construction. To this end, we integrate ExpWeaver into four representative frameworks with diverse forms of experience, spanning both single-agent (Ouyang et al., 2025; Wang et al., 2025; Xia et al., 2026) and multi-agent settings (Zhang et al., 2025a), and compare it against their vanilla experience usage strategies, which follow either initialization-only or always-on schemes. Following the original evaluation settings of these frameworks, we evaluate different experience utilization strategies across seven LLM backbones and three types of environments, including embodied interaction, web navigation, and knowledge-intensive question answering. Across these settings, ExpWeaver consistently outperforms the corresponding vanilla frameworks without modifying existing experience construction pipelines. These results show that, beyond experience construction and representation, runtime utilization also meaningfully affects self-evolving agents (§4.2).

We further investigate whether interweaving experience utilization with decision-making is merely an inference-time prompting effect or a capability that can be strengthened through learning (§4.3). To this end, we incorporate ExpWeaver into agentic reinforcement learning, where experience serves both as inference-time context and as a signal for policy optimization (Wu et al., 2025; Zhai et al., 2025; Xia et al., 2026). Experiments on Qwen3-4B and Qwen3-14B (Yang et al., 2025a), trained with GRPO (Shao et al., 2024), show that ExpWeaver consistently achieves the best performance among different experience utilization strategies. This suggests that interweaving experience utilization within decision-making is a learnable capability that can be further amplified through training.

To further understand what interweaving experience utilization with decision-making brings to agents, we conduct a series of analyses that reveal ExpWeaver's behavior and mechanism. First, ExpWeaver

induces a non-uniform pattern of experience utilization: agents invoke experience more frequently in demanding tasks and non-uniformly across a trajectory (§5.1). Second, targeted ablations show that this pattern is not incidental but causally beneficial. Removing experience at positions selected by ExpWeaver leads to consistent performance degradation, while random experience utilization fails to yield comparable gains, indicating that the effectiveness of experience depends critically on when it is invoked (§5.2). Finally, we provide mechanistic evidence that ExpWeaver tends to invoke experience at decision points associated with higher uncertainty, as reflected by increased token-level entropy during the agent's reasoning process (§5.3). Together, these findings suggest that coupling experience utilization with decision-making enables agents to adaptively regulate experience use in response to decision difficulty.

Overall, our findings suggest that effective self-evolution depends not only on accumulating better experience, but also on regulating how experience enters the decision-making process. This study highlights experience utilization as an important design dimension for self-evolving agents, motivating greater attention to not only *what* experience is stored, but also *how* and *when* it should be used.

2 Related Works

Existing works on experience-driven self-evolving agents can be broadly discussed from two perspectives: experience construction and experience utilization.

Experience construction studies how agents should represent, store, and update past interactions. A large body of work focuses on constructing external experience repositories from offline datasets or online interactions. Offline approaches pre-construct reusable experience from collected trajectories and keep the memory fixed during inference (Li & Qiu, 2023; Yang et al., 2023; Zhong et al., 2024; Zhao et al., 2024a; Fu et al., 2024; Zhou et al., 2025; Yang et al., 2025b). Online approaches further allow the experience repository to evolve during deployment, enabling agents to accumulate, retrieve, and refine experience through interaction (Chen et al., 2024; Zhang et al., 2025b; Suzgun et al., 2025; Ouyang et al., 2025). Representative systems such as ReasoningBank continuously distill reasoning patterns from recent episodes, while G-Memory extends online memory accumulation to multi-agent settings (Ouyang et al., 2025; Zhang et al., 2025a). These studies have produced diverse experience forms, including distilled insights, skills, workflows, and raw trajectories.

In contrast, *experience utilization* concerns how constructed experience should enter the agent's runtime decision-making process. This aspect remains comparatively underexplored. Recent concurrent studies have begun to explore more adaptive experience usage, such as proactive retrieval in lifelong agents (Cai et al., 2026) and self-triggered experience seeking in web agents (Zhang et al., 2026). However, since these works introduce their own experience representations and construction pipelines, the impact of experience utilization is intertwined with changes in how experience is formed, making it difficult to cleanly isolate the effect of *how* experience is used.

Our work instead takes a higher-level view by rethinking the role of experience in self-evolving agents: experience should not merely serve as static auxiliary context, but as an active component of decision-making. To isolate this question, we keep existing experience construction, representation, and update pipelines unchanged, and only modifies runtime utilization. This allows us to identify experience utilization as a critical design dimension, showing that effective self-evolution depends not only on what experience is stored, but also on how and when it enters decision-making.

3 Interweaving Experience Utilization within Decision-Making

In this section, we first formulate experience-driven self-evolution as a two-stage process consisting of experience construction and experience utilization (§3.1). We then introduce ExpWeaver, a lightweight intervention that interweaves experience utilization with the agent's reasoning process, enabling us to examine whether experience can be invoked when needed during decision-making (§3.2). Finally, we extend this formulation to agentic reinforcement learning to study whether such usage behavior can be further optimized as a learnable capability (§3.3).

3.1 PRELIMINARIES

We study self-evolving agents, where behavioral adaptation is achieved by accumulating and reusing past interactions without modifying model parameters (Gao et al., 2025). This process consists of two stages: constructing an experience repository and utilizing it during decision-making.

Experience Construction. After each interaction, the agent produces a trajectory τ and receives feedback r. From each (τ, r) pair, an experience unit E is distilled and stored in an external memory:

$$M = \{E_1, E_2, \ldots, E_n\}.$$

Different frameworks instantiate E as distilled insights (Ouyang et al., 2025), reusable skills (Xia et al., 2026), or executable workflows (Wang et al., 2025), with corresponding memory update rules. Most works focus on this stage, improving how experience is represented, organized, and maintained.

Experience Utilization. Experience utilization determines how the constructed experience memory M enters runtime decision-making, and is largely orthogonal to experience construction. Given a task input x, existing methods retrieve relevant experience $M(x) \subseteq M$ and typically use it in fixed and rigid ways. One strategy is *initialization-only* usage:

$$y = \pi_\theta([x; M(x)]),$$

where experience is injected once. Another is *always-on* usage in multi-step decision-making:

$$a_t = \pi_\theta([s_t; M(s_t)]),$$

where experience is injected at every step t.

Both strategies predefine when experience is used, rather than allowing the agent to decide whether it is needed for the current decision. This motivates us to study experience utilization as a runtime decision process: how should experience selectively enter decision-making when useful?

3.2 PROMPTING-BASED EXPWEAVER

To study this question, we propose **ExpWeaver**, a lightweight paradigm that interweaves experience utilization with the agent's decision-making process. ExpWeaver leaves experience construction, representation, and updating unchanged, and only modifies runtime utilization by exposing experience as an optional resource during reasoning.

From Fixed Usage to Decision-Time Utilization. We build on the widely adopted ReAct-style (Yao et al., 2023) paradigm, where agents follow an iterative `reasoning` $\rightarrow$ `action` loop. At step t, the agent maintains a context c_t and generates an action $a_t \sim \pi_\theta(\cdot \mid c_t)$.

In existing agents, experience utilization is external to this loop: experience is injected either before execution or at every step. ExpWeaver instead places experience utilization inside the `reasoning` process, making the decision of whether to use experience conditioned on the agent's reasoning state.

Interweaving Experience into Reasoning. ExpWeaver implements this idea through a minimal prompting-based intervention. We augment the system prompt with an instruction that reminds the agent that it may consult past experience during reasoning if additional guidance is needed. When the agent decides to use experience, it emits a special trigger token, `[Retrieve]`, which activates retrieval. The full prompt template is provided in Appendix D.1.

Formally, at each step t, the agent first produces an intermediate reasoning trace:

$$h_t = \pi_\theta(c_t),$$

which reflects its current understanding of the state. We then check whether h_t contains the trigger token. If the trigger appears, the current context is used as a retrieval query:

$$M(c_t) = \text{Retrieve}(M, c_t),$$

and the retrieved experience is incorporated for subsequent action generation:

$$a_t = \pi_\theta([c_t; h_t; M(c_t)]).$$

If the trigger token does not appear, the agent proceeds without additional experience incorporated. Overall, this yields the following decision loop:

`reasoning` $\rightarrow$ `(optional experience utilization)` $\rightarrow$ `action`.

Role of ExpWeaver. ExpWeaver should be viewed as a minimal instantiation for studying runtime experience utilization, rather than as a modification to experience construction. It introduces three useful properties for analysis: **(1) Adaptive usage.** Experience is invoked during reasoning only when the agent determines that additional guidance may be useful. **(2) Orthogonality.** ExpWeaver is independent of how experience is represented or updated, making it applicable to diverse self-evolving frameworks. **(3) Observability.** Because retrieval is triggered explicitly within the reasoning process, we can inspect when experience is used and analyze whether such usage is meaningful.

3.3 Agentic Reinforcement Learning-based ExpWeaver

The prompting-based setup tests whether existing agents can be guided to regulate experience usage at inference time. We further ask whether this behavior can be strengthened through learning. To this end, we integrate ExpWeaver into experience-driven agentic reinforcement learning.

Policy Optimization with GRPO. We train the policy using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). For an input x, the agent samples G trajectories $\{\tau^{(1)}, \ldots, \tau^{(G)}\}$. Each trajectory receives a binary reward $R_i = r(\tau^{(i)}) \in \{0, 1\}$ indicating whether the task is successfully completed. The policy is updated with a PPO-style (Schulman et al., 2017) clipped objective:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G} \min\left(r_i A_i, \text{clip}(r_i, 1-\epsilon, 1+\epsilon)A_i\right) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\right],$$

where A_i is computed from intra-group normalized rewards, and the KL penalty keeps the updated policy close to the reference policy.

4 How Should Experience Be Utilized in Self-Evolving Agents?

4.1 Experimental Setup

Experience Utilization Strategies. We compare four experience utilization strategies. **w/o Experience** removes the experience repository and performs task execution without using any past experience. **Initialization-only (Init-only)** retrieves relevant experience once at the beginning of task execution and keeps it as static context throughout the trajectory. **Always-on** retrieves and injects experience at every decision step, regardless of whether it is needed for the current state. **ExpWeaver** interweaves experience utilization with the decision-making process, allowing the agent to invoke experience during reasoning only when additional guidance is needed.

Agent Framework. We implement the four experience utilization strategies described above across four representative self-evolving frameworks. These frameworks cover diverse forms of experience representation: **ReasoningBank** (Ouyang et al., 2025) uses distilled insights, **AWM** (Wang et al., 2025) represents experience as executable workflows, **SkillRL** (Xia et al., 2026) models experience as reusable skills, and **G-Memory** (Zhang et al., 2025a) combines raw trajectories with distilled insights. G-Memory operates in a *multi-agent* setting, while the others are *single-agent* frameworks.

The vanilla versions of ReasoningBank, SkillRL, and G-Memory follow an Init-only usage strategy. AWM does not include an explicit retrieval mechanism; therefore, its Init-only and Always-on variants are equivalent in practice. These frameworks also cover both *offline* and *online* self-evolving paradigms: SkillRL relies on a pre-constructed experience repository, whereas the other frameworks dynamically accumulate experience through interaction with the environment.

This selection spans a broad spectrum of experience representations, agent settings, and evolution paradigms, enabling a comprehensive evaluation of how different experience utilization strategies affect self-evolving agents. Further details are provided in Appendix B.

Backbone Model. We conduct experiments across a diverse set of seven LLMs from multiple families, including **GPT-5.2** (Singh et al., 2025), **DeepSeek-V4-Pro** (DeepSeek-AI, 2026), and **Kimi-K2.5** (Team, 2026), as well as open-weight Qwen series including **Qwen3.5-397B-A17B** (Qwen Team, 2026) and several **Qwen3** variants (32B, 14B and 4B) (Yang et al., 2025a), covering a wide range of model scales and architectures for comprehensive evaluation. Due to computational constraints, we conduct reinforcement learning experiments only on Qwen3-14B and Qwen3-4B.

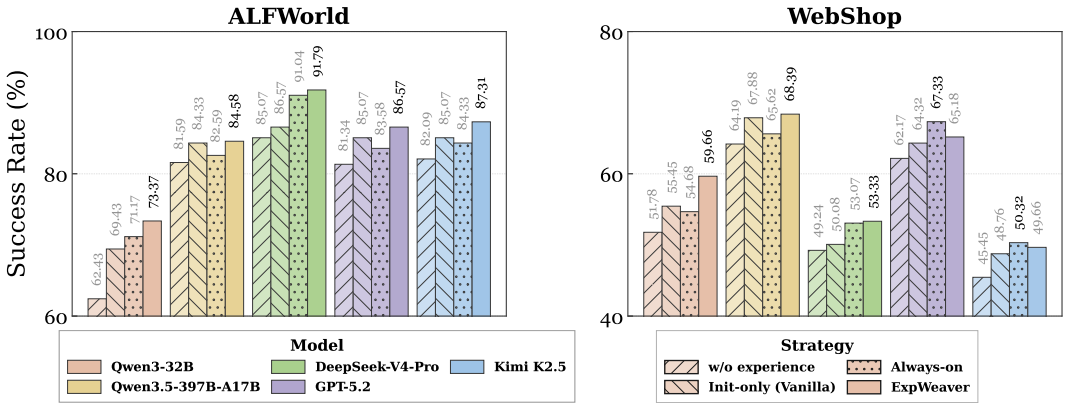


Figure 2: Prompting-based results under the ReasoningBank framework on ALFWorld and WebShop.

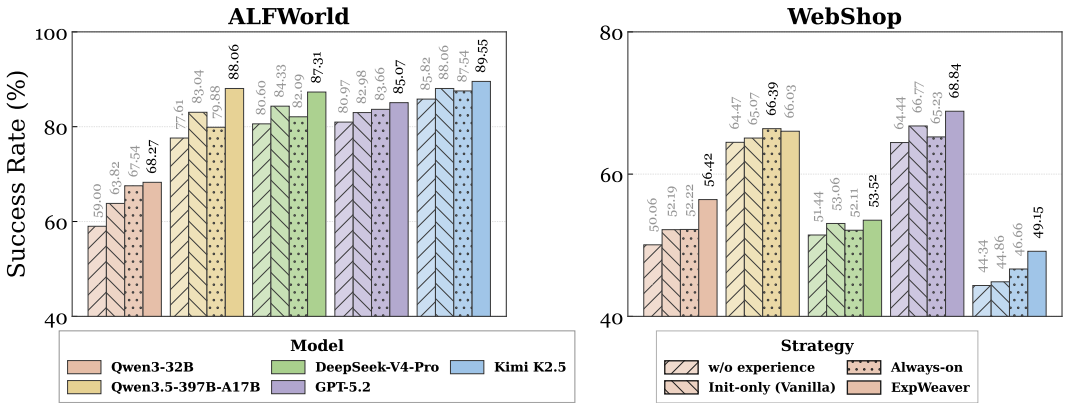


Figure 3: Prompting-based results under the SkillRL framework on ALFWorld and WebShop.

Environment & Benchmark. We evaluate across 8 benchmarks in 3 environments: (1) For *embodied interaction*, we adopt the interactive environment **ALFWorld** (Shridhar et al., 2021). (2) For *web navigation*, we evaluate on **WebShop** (Yao et al., 2022). (3) For *knowledge-intensive question answering (QA)*, we include **HotpotQA** (Yang et al., 2018), **NQ** (Kwiatkowski et al., 2019) **TriviaQA** (Joshi et al., 2017), **2Wiki** (Ho et al., 2020), **MuSiQue** (Trivedi et al., 2022) and **Bamboogle** (Press et al., 2023). These benchmarks span diverse environments and task types, providing a comprehensive evaluation of experience-driven adaptation (details in Appendix C).

Implementation Details. For prompting-based experiments, we access those models via official APIs. All agentic reinforcement learning experiments are conducted on a cluster of eight NVIDIA A100 80GB GPUs. Further implementation details are provided in Appendix D.

4.2 Results on Prompting-based ExpWeaver

Figures 2 and 3 present the prompting-based results on ALFWorld and WebShop across ReasoningBank and SkillRL frameworks. Due to space constraints, results on knowledge-intensive QA tasks and the full results for G-Memory and AWM are provided in Appendix E.1. Overall, ExpWeaver consistently achieves the best performance among different experience utilization strategies across models, environments, and experience representations. This demonstrates the efficacy of interweaving experience utilization into the decision-making process. We highlight the following key observations:

Consistent gains over rigid experience usage strategies. ExpWeaver consistently improves over both *Init-only* and *Always-on* baselines across nearly all settings. This trend holds across different backbones and environments. The results indicate that neither under-utilization nor over-utilization of experience is optimal; instead, integrating experience utilization into the decision-making process leads to more reasonable behavior and effective performance.

Robust effectiveness across diverse experience representations. ExpWeaver achieves consistent improvements across frameworks with fundamentally different experience forms, including distilled insights (ReasoningBank), workflows (AWM), skills (SkillRL), and hybrid memory (G-Memory). Importantly, within each framework, ExpWeaver leaves the original experience representation and construction pipeline unchanged, modifying only how the existing experience is utilized at runtime. The consistent gains across these heterogeneous frameworks suggest that the effectiveness of adaptive utilization is not tied to a particular form of experience. This provides further evidence that, beyond improving how experience is represented and constructed, how it is utilized during decision-making constitutes an important and broadly applicable design dimension.

Strong generalization across agent settings and self-evolving paradigms. ExpWeaver generalizes across both single-agent and multi-agent settings (i.e., G-Memory), as well as offline and online self-evolving paradigms. This is notable given their distinct experience construction and update mechanisms. In particular, SkillRL provides a controlled offline setting where ExpWeaver uses the same pre-constructed experience repository as the vanilla baseline and only changes runtime utilization. The gains in this setting indicate that the improvement comes from regulating experience usage itself, supporting ExpWeaver as a general mechanism across heterogeneous agent designs.

4.3 Results on RL-based ExpWeaver

We further investigate whether interweaving experience utilization with the decision-making process can be strengthened through agentic reinforcement learning. Specifically, we adopt the ReasoningBank framework and train Qwen3-4B on the ALFWorld environment with GRPO under four experience utilization strategies. Detailed training settings are provided in Appendix D.2.

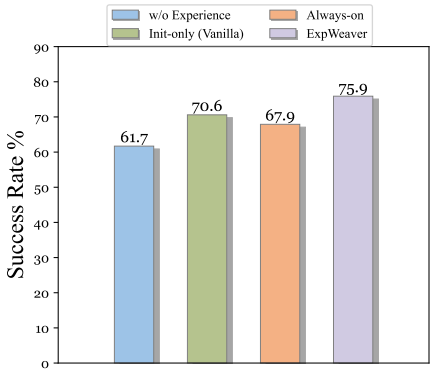


Figure 4: RL results of Qwen3-4B on ALFWorld after GRPO training.

Figure 4 shows that the way experience is used substantially affects RL performance. All experience-based variants outperform w/o Experience, confirming the value of experience under GRPO training. Yet Always-on underperforms Init-only, suggesting that excessive experience exposure can introduce interference. ExpWeaver achieves the highest success rate, improving over Init-only by 5.3 points and Always-on by 8.0 points. This indicates that RL benefits most when experience utilization is interwoven with decision-making rather than fixed as static context or injected at every step.

We also observe consistent trends on knowledge-intensive QA with Qwen3-14B. As shown in Appendix E.2, ExpWeaver generally achieves stronger learning dynamics and higher final success rates than the vanilla baseline across QA benchmarks such as NQ, TriviaQA, Bamboogle, and 2Wiki. These results suggest that the learnability of experience utilization is not limited to embodied interaction, but also generalizes to reasoning-intensive QA scenarios.

5 Analysis and Discussions

5.1 Usage Pattern Analysis of ExpWeaver

To understand how ExpWeaver utilizes experience in practice, we analyze its usage patterns from two perspectives: (1) a *horizontal comparison* across tasks, measured by the average number of experience retrievals per sample; and (2) a *temporal analysis* within trajectories, measured by the fraction of trajectories that invoke experience at each step.

Horizontal comparison across tasks. Table 1 reports the average number of experience retrievals per sample under two representative frameworks, ReasoningBank and G-Memory. Additional results on AWM and SkillRL are provided in Appendix F.1. Overall, ExpWeaver exhibits a clear task-aware, model-aware, and framework-agnostic usage pattern.

Table 1: Average number of experience retrievals per sample under ExpWeaver. We report results on ReasoningBank and G-Memory.

Model	ReasoningBank			G-Memory		
	ALFWorld	WebShop	HotpotQA	ALFWorld	WebShop	HotpotQA
`Qwen3-32B`	2.17	0.13	0.42	1.87	0.04	0.99
`Qwen3.5-397B-A17B`	0.43	0.00	0.00	0.75	0.00	0.01
`DeepSeek-V4-Pro`	0.82	0.95	0.17	0.94	0.16	0.18
`GPT-5.2`	0.00	0.00	0.00	0.00	0.15	0.01
`Kimi-K2.5`	1.14	0.02	0.03	2.31	0.00	0.18

First, experience usage is highly task-dependent. Across most frameworks and backbones, ALFWorld induces substantially more retrievals than WebShop and HotpotQA, suggesting that experience is more frequently needed in embodied interaction, where agents must perform long-horizon planning, track partial observations, and recover from uncertain states. Second, retrieval frequency tends to decrease as model capability increases. Stronger backbone models such as GPT-5.2 and Qwen3.5-397B-A17B rarely invoke experience, while smaller models such as Qwen3-32B rely on it more often. This indicates that ExpWeaver does not mechanically retrieve experience, but treats it as a complementary resource when the model requires additional guidance. Third, similar patterns appear across different self-evolving frameworks, despite their distinct experience representations. This suggests that the observed behavior is not tied to a specific memory format, but is induced by interweaving experience utilization into the decision-making process.

Temporal analysis within decision trajectories. Figure 5 reports the fraction of trajectories that invoke experience at each interaction step in ALFWorld under the ReasoningBank framework. Experience utilization shows a clearly non-uniform temporal pattern: retrieval is concentrated at the first step, drops sharply afterward, and occasionally resurges at later stages. This suggests that agents mainly consult experience when task understanding and high-level planning are most critical, rather than using it uniformly throughout the trajectory.

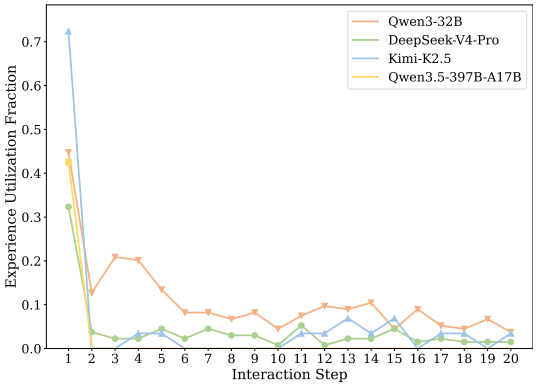


Figure 5: Temporal pattern of experience utilization under ExpWeaver on ALFWorld with the ReasoningBank framework.

Importantly, later retrievals are not random. In ALFWorld, they often occur during object-search phases, where the agent must locate target items under partial observability. When search attempts fail repeatedly, experience is re-invoked to provide corrective guidance. This suggests that experience utilization may be associated with moments of increased decision uncertainty, a hypothesis we further examine through entropy-based analysis in §5.3. Additional temporal analyses across more frameworks and tasks are provided in Appendix F.2, and specific qualitative examples of ExpWeaver on different tasks are also provided in Appendix F.3.

5.2 Causal Effect of ExpWeaver-Invoked Experience Utilization

We further examine whether the observed ExpWeaver's non-uniform usage pattern is causally beneficial. We compare ExpWeaver with two ablations: **Empty**, which removes retrieved experience at positions invoked by ExpWeaver, and **Random Utilization**, which invokes experience at random positions. For Random Utilization, we first compute the average number of experience invocations and the average trajectory length of ExpWeaver on each benchmark, and then invoke experience at each step with probability given by their ratio.

As shown in Table 2, both ablations consistently underperform ExpWeaver. We focus on Qwen3-32B and DeepSeek-V4-Pro because these two backbones invoke experience more frequently under ExpWeaver, providing enough usage events for reliable ablation analysis. For Qwen3-32B, removing

Table 2: Causal ablation of ExpWeaver-invoked experience utilization. Empty removes retrieved experience at positions where ExpWeaver chooses to invoke it, while Random Utilization invokes experience at randomly selected positions.

Method	ReasoningBank			G-Memory		
	ALFWorld	WebShop	HotpotQA	ALFWorld	WebShop	HotpotQA
Qwen3-32B						
ExpWeaver	**73.37**	**59.66**	**45.33**	**83.33**	**45.56**	**47.00**
Random Utilization	68.46	49.94	44.67	71.64	37.72	45.33
Empty	71.64	54.56	43.00	75.37	39.53	45.33
DeepSeek-V4-Pro						
ExpWeaver	**91.79**	**53.33**	**55.00**	**88.81**	**66.02**	**55.00**
Random Utilization	84.33	48.29	52.00	81.34	62.11	52.67
Empty	86.57	50.92	53.00	85.82	64.09	53.67

experience leads to clear drops across ReasoningBank and G-Memory, while random utilization also fails to match ExpWeaver. DeepSeek-V4-Pro exhibits the same trend, with ExpWeaver consistently outperforming both ablation variants across all evaluated settings.

These results provide causal evidence that interweaving experience utilization with decision-making is effective because the agent learns *when* experience should enter the reasoning process.

5.3 Mechanistic Analysis: Experience Usage under Uncertainty

We further analyze what happens inside the agent when ExpWeaver decides to invoke experience. Our hypothesis is that experience usage is triggered when the agent encounters higher decision uncertainty. To examine this, we measure the token-level entropy of the agent's output distribution during each reasoning step, and compare the entropy at positions with and without experience usage.

Figure 6 presents the entropy analysis results on ALFWorld. The blue curve denotes the mean token entropy at each step, while red points indicate steps where ExpWeaver invokes experience. We observe that experience usage tends to occur at entropy peaks. In contrast, steps without experience usage generally correspond to lower-entropy regions. This suggests that ExpWeaver is more likely to retrieve experience when the agent is uncertain about its next decision.

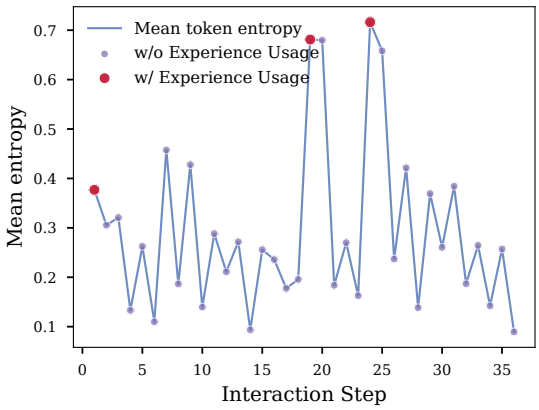


Figure 6: Mechanistic analysis of experience usage in ExpWeaver. The blue curve shows the mean token entropy at each reasoning step, while red points denote steps where experience is invoked.

6 Conclusion

This work studies how experience should be utilized in self-evolving agents. We show that existing rigid usage schemes overlook an important design dimension: when and how experience should enter decision-making. To examine this question, we introduce ExpWeaver, a lightweight instantiation that keeps experience construction unchanged while modifying runtime utilization. Across diverse frameworks, models, and environments, ExpWeaver consistently outperforms rigid usage strategies. Further analyses show that agents invoke experience selectively, at causally beneficial positions, and under higher reasoning uncertainty. Overall, our findings suggest that effective self-evolution depends not only on what experience is stored, but also on how and when it is used.

AI USE STATEMENT

In this work, we used generative AI tools to assist with implementing the proposed methods, translating and polishing technical text, and interpreting experimental results. We did not use generative AI tools to develop theoretical models or conceptual frameworks, propose or refine hypotheses, design or provide feedback on the research methodology or experiments, generate or clean datasets, or conduct qualitative or thematic data analysis. We did not use generative AI tools to generate synthetic datasets, formulate mathematical claims, provide essential ingredients for mathematical proofs, or assist in writing proofs; these tasks were not applicable to this work.

Additionally, we used generative AI tools to suggest experimental parameters, create and edit software code, summarize and analyze related literature, identify research topics and gaps, brainstorm ideas, search for relevant information, format references, suggest the structure of the paper, improve the title and keywords, and edit the manuscript for readability. AI assistance was also used to draft and refine portions of the manuscript, including the abstract, conclusion, figure captions, and qualitative case-study descriptions. All AI-assisted code, analyses, interpretations, and written material were reviewed and verified by the authors. Experimental results were independently checked against the implemented methods and reported measurements, and all claims and case-study descriptions were revised where necessary. The authors take full responsibility for the final content of this work, including all text, claims, code, analyses, figures, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work involves no human subjects, no private data, and no new dataset or model release; all experiments use public benchmarks and LLMs under their intended licenses. The main risk is that experience memory may contain biased, unsafe, or outdated content, and adaptive retrieval could amplify it; deployment should therefore include filtering, monitoring, and human oversight. Our evaluation is limited to English benchmarks and selected backbones, so fairness and cross-domain robustness are not guaranteed. Generative AI use is disclosed in the AI Use Statement, and all outputs were verified by the authors. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

Detailed information on environment and benchmark usage is provided in Appendix C. Appendix B introduces all baseline methods used for comparison, covering all self-evolving agents discussed in this paper. For reproducibility, Appendix D describes the experimental setup and implementation details, including the codebases used, software versions, hardware configuration, and hyperparameter settings.

REFERENCES

Michael C Anderson, Kevin N Ochsner, Brice Kuhl, Jeffrey Cooper, Elaine Robertson, Susan W Gabrieli, Gary H Glover, and John DE Gabrieli. Neural systems underlying the suppression of unwanted memories. *Science*, 303(5655):232–235, 2004.

Marcel Bausch, Johannes Niediek, Thomas P Reber, Sina Mackay, Jan Boström, Christian E Elger, and Florian Mormann. Distinct neuronal populations in the human brain combine content and context. *Nature*, pp. 1–11, 2026.

Jack Bell, Luigi Quarantiello, Eric Nuertey Coleman, Lanpei Li, Malio Li, Mauro Madeddu, Elia Piccoli, and Vincenzo Lomonaco. The future of continual learning in the era of foundation models: Three key directions. *arXiv preprint arXiv:2506.03320*, 2025.

Matthew M Botvinick, Todd S Braver, Deanna M Barch, Cameron S Carter, and Jonathan D Cohen. Conflict monitoring and cognitive control. *Psychological review*, 108(3):624, 2001.

Yuxuan Cai, Yipeng Hao, Jie Zhou, Hang Yan, Zhikai Lei, Rui Zhen, Zhenhua Han, Yutao Yang, Junsong Li, Qianjun Pan, et al. Building self-evolving agents via experience-driven lifelong learning: A framework and benchmark. *arXiv preprint arXiv:2508.19005*, 2025.

Yuxuan Cai, Jie Zhou, Qin Chen, and Liang He. Ask only when needed: Proactive retrieval from memory and skills for experience-driven lifelong agents. *arXiv preprint arXiv:2604.20572*, 2026.

Minghao Chen, Yihang Li, Yanting Yang, Shiyu Yu, Binbin Lin, and Xiaofei He. Automanual: Constructing instruction manuals by llm agents via interactive environmental learning. *Advances in Neural Information Processing Systems*, 37:589–631, 2024.

DeepSeek-AI. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026.

Shihan Dou, Ming Zhang, Chenhao Huang, Jiayi Chen, Feng Chen, Shichun Liu, Yan Liu, Chenxiao Liu, CHENG ZHONG, Zongzhang Zhang, et al. Evalearn: Quantifying the learning capability and efficiency of llms via sequential problem solving. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, et al. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems. *arXiv preprint arXiv:2508.07407*, 2025.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Yao Fu, Dong-Ki Kim, Jaekyeom Kim, Sungryull Sohn, Lajanugen Logeswaran, Kyunghoon Bae, and Honglak Lee. Autoguide: Automated generation and selection of context-aware guidelines for large language model agents. *Advances in Neural Information Processing Systems*, 37:119919–119948, 2024.

Huan-ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Yiran Wu, et al. A survey of self-evolving agents: On path to artificial super intelligence. *arXiv preprint arXiv:2507.21046*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop qa dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

Botao'Amber' Hu. On improvisation and open-endedness: Insights for experiential ai. *arXiv preprint arXiv:2511.00529*, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan O Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *Second Conference on Language Modeling*, 2025.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Xiaonan Li and Xipeng Qiu. Mot: Memory-of-thought enables chatgpt to self-improve. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6354–6374, 2023.

Minhua Lin, Hanqing Lu, Zhan Shi, Bing He, Rui Mao, Zhiwei Zhang, Zongyu Wu, Xianfeng Tang, Hui Liu, Zhenwei Dai, et al. Position: Agentic evolution is the path to evolving llms. *arXiv preprint arXiv:2602.00359*, 2026.

Siru Ouyang, Jun Yan, I Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long T Le, Samira Daruki, Xiangru Tang, et al. Reasoningbank: Scaling agent self-evolving with reasoning memory. *arXiv preprint arXiv:2509.25140*, 2025.

Wenbo Pan, Shujie Liu, Xiangyang Zhou, Shiwei Zhang, Wanlu Shi, Mirror Xu, and Xiaohua Jia. M-star: Every task deserves its own memory harness. *arXiv preprint arXiv:2604.11811*, 2026.

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

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

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

Shuai Shao, Qihan Ren, Chen Qian, Boyi Wei, Dadi Guo, Yang JingYi, Xinhao Song, Linfeng Zhang, Weinan Zhang, Dongrui Liu, et al. Your agent may misevolve: Emergent risks in self-evolving llm agents. In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*, 2025.

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

Richard M Shiffrin and Walter Schneider. Controlled and automatic human information processing: Ii. perceptual learning, automatic attending and a general theory. *Psychological review*, 84(2):127, 1977.

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10740–10749, 2020.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Mirac Suzgun, Mert Yuksekgonul, Federico Bianchi, Dan Jurafsky, and James Zou. Dynamic cheatsheet: Test-time learning with adaptive memory. *arXiv preprint arXiv:2504.07952*, 2025.

Kimi Team. Kimi k2.5: Visual agentic intelligence, 2026.

Hyoe Tomita, Machiko Ohbayashi, Kiyoshi Nakahara, Isao Hasegawa, and Yasushi Miyashita. Top-down signal from prefrontal cortex in executive control of memory retrieval. *Nature*, 401(6754): 699–703, 1999.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

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*, 2024.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In *International Conference on Machine Learning*, pp. 63897–63911. PMLR, 2025.

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

Peng Xia, Jianwen Chen, Hanyang Wang, Jiaqi Liu, Kaide Zeng, Yu Wang, Siwei Han, Yiyang Zhou, Xujiang Zhao, Haifeng Chen, et al. Skillrl: Evolving agents via recursive skill-augmented reinforcement learning. In *ICLR 2026 Workshop on Memory for LLM-Based Agentic Systems*, 2026.

Yiming Xiong, Shengran Hu, and Jeff Clune. Learning to continually learn via meta-learning agentic memory designs. *arXiv preprint arXiv:2602.07755*, 2026.

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

Wei Yang, Jinwei Xiao, Hongming Zhang, Qingyang Zhang, Yanna Wang, and Bo Xu. Coarse-to-fine grounded memory for llm agent planning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 13040–13067, 2025b.

Zeyuan Yang, Peng Li, and Yang Liu. Failures pave the way: Enhancing large language models through tuning-free rule accumulation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1751–1777, 2023.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pp. 2369–2380, 2018.

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

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.

Yunpeng Zhai, Shuchang Tao, Cheng Chen, Anni Zou, Ziqian Chen, Qingxu Fu, Shinji Mai, Li Yu, Jiaji Deng, Zouying Cao, et al. Agentevolver: Towards efficient self-evolving agent system. *arXiv preprint arXiv:2511.10395*, 2025.

Guibin Zhang, Muxin Fu, Kun Wang, Guancheng Wan, Miao Yu, and Shuicheng YAN. G-memory: Tracing hierarchical memory for multi-agent systems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025a.

Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, et al. Agentic context engineering: Evolving contexts for self-improving language models. *arXiv preprint arXiv:2510.04618*, 2025b.

Wenyuan Zhang, Xinghua Zhang, Haiyang Yu, Shuaiyi Nie, Bingli Wu, Juwei Yue, Tingwen Liu, and Yongbin Li. Expseek: Self-triggered experience seeking for web agents. *arXiv preprint arXiv:2601.08605*, 2026.

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

Weixiang Zhao, Shilong Wang, Yulin Hu, Yanyan Zhao, Bing Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. Sapt: A shared attention framework for parameter-efficient continual learning of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11641–11661, 2024b.

Weixiang Zhao, Yingshuo Wang, Yichen Zhang, Yang Deng, Yanyan Zhao, Wanxiang Che, Bing Qin, and Ting Liu. Large language model agents are not always faithful self-evolvers. *arXiv preprint arXiv:2601.22436*, 2026a.

Weixiang Zhao, Yichen Zhang, Yingshuo Wang, Yang Deng, Yanyan Zhao, Xuda Zhi, Yongbo Huang, Wanxiang Che, Bing Qin, Ting Liu, et al. On safety risks in experience-driven self-evolving agents. *arXiv preprint arXiv:2604.16968*, 2026b.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19724–19731, 2024.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, et al. Memento: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153*, 2025.

A Limitations and Impact Statement

Limitations. This work studies how experience should be utilized in self-evolving agents and shows that interweaving experience utilization with decision-making can improve performance across diverse frameworks and environments. Nevertheless, several limitations remain.

First, ExpWeaver is implemented through prompting and trigger-based retrieval, which makes it lightweight and broadly applicable, but also means that its behavior depends on the instruction-following ability of the underlying LLM. We observe consistent gains across multiple backbones, but weaker small-scale models may fail to reliably decide when experience is needed through prompting.

Second, our analysis focuses on experience utilization under existing experience construction pipelines. While this design helps isolate the effect of runtime utilization, it does not jointly optimize experience construction and utilization. A promising direction is the recent trend of fully automated experience memory design, where agents can automatically discover, refine, and reorganize experience representations over time (Xiong et al., 2026; Pan et al., 2026). Combining such automated memory construction with adaptive utilization may provide a more unified path toward self-evolving agents that can improve not only what they store, but also how and when they use it.

Finally, our current study primarily evaluates task performance. Other dimensions, such as safety, robustness, calibration, and long-term stability under continuous deployment, remain important directions for future investigation (Shao et al., 2025; Zhao et al., 2026b;a).

Impact Statement. This work highlights experience utilization as a critical design dimension for self-evolving agents. By showing that agents benefit from selectively invoking experience during decision-making, our findings may help build more adaptive, efficient, and interpretable agent systems. In particular, ExpWeaver provides a lightweight way to improve existing self-evolving frameworks without modifying their experience construction pipelines, which may lower the cost of deploying experience-driven agents in diverse environments.

At the same time, more effective experience utilization may also amplify both desirable and undesirable behaviors contained in accumulated experience. If the experience repository contains biased, unsafe, outdated, or task-inappropriate content, selectively invoking such experience may still mislead the agent. Therefore, practical deployment should be accompanied by careful experience filtering, monitoring, and safety evaluation. We hope this work encourages future research to study not only how agents accumulate experience, but also how experience should be governed, audited, and safely integrated into decision-making.

B Self-Evolving Agents

We present detailed description of the four experience-driven self-evolving agents used in our experiments: ReasoningBank (Ouyang et al., 2025), AWM(Wang et al., 2025), SkillRL(Xia et al., 2026) and G-Memory (Zhang et al., 2025a). Across all frameworks, agents adapt their behavior by accumulating, retrieving, and reusing external experiences stored in explicit memory structures, rather than by updating the parameters of the underlying language model. In the following, we summarize the core design principles and memory mechanisms of these agents in detail:

- **ReasoningBank** (Ouyang et al., 2025) embodies an instance of an online self-evolving agent. It keeps an ever-expanding memory that records condensed reasoning patterns distilled from the agent's past interactions, covering both successful cases and failures. Following each task completion, the agent assesses its own performance and selectively incorporates newly acquired experiences into this memory repository. During inference, pertinent reasoning strategies are retrieved from the memory and incorporated into the agent's context to guide future interactions. The system forms a closed feedback loop where experiences are continuously accumulated, accessed, and reused throughout deployment, enabling the agent's behavior to progressively adapt over time.
- **AWM** (Wang et al., 2025) embodies another instance of an online self-evolving agent. It maintains a growing memory of reusable workflows—common sub-routines distilled from successful agent trajectories, with example-specific values abstracted into variables. After each task, an LLM-based evaluator judges whether the trajectory succeeded; if so, the agent induces one or more workflows from the trajectory and adds them to the memory. During inference, relevant workflows are retrieved and incorporated into the agent's context to guide future actions. AWM operates both offline (inducing workflows from annotated training examples) and online (learning purely from test queries in a streaming fashion). This closed-loop process enables the agent to accumulate increasingly complex workflows over time, rapidly improving performance from only a handful of examples and generalising across tasks, websites, and domains.
- **SkillRL** (Xia et al., 2026) is an online self-evolving agent that bridges raw experience and policy improvement via automatic skill discovery and recursive evolution. It maintains a hierarchical skill library—SKILLBANK—comprising general strategies and task-specific heuristics, both distilled from successful and failed trajectories by a teacher model. During inference, relevant skills are retrieved to augment the agent's context, achieving 10–20× token compression over raw trajectories. A cold-start supervised fine-tuning stage teaches the base model how to use skills, followed by reinforcement learning (GRPO) that jointly optimises the policy and recursively evolves the skill library: failed validation trajectories trigger generation of new skills, enabling the library and policy to co-adapt. This closed-loop process yields faster convergence, higher asymptotic performance, and strong generalisation across embodied and web tasks.
- **G-Memory** (Zhang et al., 2025a) is an online self-evolving memory mechanism designed for multi-agent systems. It maintains a shared, persistent memory that records past multi-agent collaboration experiences across tasks, capturing both abstract insights and condensed interaction histories. When a new task arrives, relevant memory entries are retrieved and selectively injected into the contexts of different agents to support coordination and reasoning. After task completion, newly generated interactions and distilled insights are incorporated into the memory, updating its contents during deployment. This continual retrieval-and-update process enables agent teams to accumulate and reuse collaborative experience over time, allowing collective behavior to adapt without modifying the underlying language models.

C Environment and Benchmark

C.1 Embodied Interaction

- **ALFWorld** (Shridhar et al., 2021) is an embodied benchmark that aligns abstract, text-based environments with interactive visual-based scenes to execute household tasks. It builds on the ALFRED (Shridhar et al., 2020) benchmark by providing paired representations of the same underlying tasks, where agents can operate through high-level textual commands in a simulated environment. The tasks span multiple categories such as pick-and-place, cleaning, heating, and cooling, and require multi-step interaction with objects and receptacles distributed across diverse room layouts. The benchmark is constructed to maintain a shared underlying world state across modalities, enabling consistent correspondence between language-level actions and embodied executions. We utilized the 134 solvable tasks.

C.2 Web Navigation

- **WebShop** (Yao et al., 2022) is a large-scale web interaction benchmark that simulates realistic online shopping scenarios through a self-contained e-commerce environment. It includes over one million real-world products and 12,087 crowdsourced natural language instructions, each specifying a product requirement to be fulfilled through a sequence of web-based actions. Given an

instruction, an agent must navigate search results, inspect product pages, select appropriate options, and complete a purchase to satisfy the specified constraints. We follow the instructions of ExpeL to set the implementation details of the WebShop Environment.

C.3 Knowledge-Intensive Question Answering

- **HotpotQA** (Yang et al., 2018) is a large-scale question answering benchmark designed to support multi-hop reasoning over natural language text. It contains question–answer pairs constructed from Wikipedia articles, where answering each question requires reasoning across multiple supporting documents. The questions are diverse in form, including standard factoid queries as well as comparison and yes/no questions, and are not constrained by predefined knowledge base schemas.
- **TriviaQA** (Joshi et al., 2017) is a challenging reading comprehension dataset containing over 650K question–answer–evidence triples. It consists of 95K question-answer pairs authored by trivia enthusiasts, with evidence documents provided for distant supervision. The questions have relatively complex, compositional syntax and often require cross-sentence reasoning to answer.
- **2WikiMultiHopQA** (Ho et al., 2020) is constructed using a hybrid of templated generation and logical rules based on Wikidata. Each question requires multi-step reasoning across two Wikipedia articles, and the dataset includes explicit evidence information that reveals the full reasoning path from the question to the answer.
- **Natural Questions** (NQ) (Kwiatkowski et al., 2019) is derived from real, anonymized Google search queries. The dataset contains over 300,000 training examples, each annotated by human raters with both long and short answers from the corresponding Wikipedia page, providing a realistic benchmark for open-domain QA.
- **Bamboogle** (Press et al., 2023) is a small yet challenging hand-crafted multi-hop QA dataset designed to measure the compositional reasoning gap in language models. Each question explicitly requires combining information from two separate facts, and it is commonly used as a difficult out-of-domain evaluation set.

For each of the above datasets following the official settings of evaluated self-evolving frameworks, we randomly selected 100 questions from its development split for our experiments.

D Implementation Details

D.1 Prompting-based Setup

ExpWeaver preserves the original design of each self-evolving agent framework with respect to experience construction, representation, and updating. Specifically, we leave the mechanisms for extracting, storing, and maintaining experience unchanged, and instead introduce only lightweight prompting. This is achieved by only augmenting the system prompt with instructions that interweave experience utilization with the reasoning process. We present the prompt designs for ALFWorld, QA tasks, and WebShop in Figure 7, Figure 8 and Figure 9, respectively. The detailed hyperparameter settings in our experiments for each self-evolving framework are described below. All reported results in our experiments are averaged over 3 independent runs.

- **ReasoningBank** Our experimental setup largely follows the configuration of ReasoningBank (Ouyang et al., 2025). For the agent, we set the decoding temperature to 0.7 and adopt greedy decoding as the decoding strategy. On the WebArena benchmark, each task is allowed a maximum of 30 interaction steps. We use *text-embedding-ada-002* as the embedding model to encode queries and memory items, and employ cosine similarity for retrieval. For each new query, the agent retrieves the top-3 most relevant memory items, which are then injected into the agent's prompt.
- **Agent Workflow Memory (AWM)** We adopt AWM (Wang et al., 2025) as a baseline. Following the original setup, we use temperature 0.0. Workflows are induced from successful trajectories, and during online inference the agent uses relevant workflows, which means no additional training data is used; the agent learns entirely from test queries in a streaming fashion.
- **G-Memory** We follow the experimental setup of G-Memory (Zhang et al., 2025a). For coarse-grained retrieval over the query graph, queries are encoded using a MiniLM sentence embedding

Table 3: Hyperparameters for ExpWeaver training.

Hyperparameter	Value
RL Training	
Learning rate	1e-6
Batch size	8
Group size G	8
Gradient accumulation steps	64
KL penalty coefficient β	0.01
Max prompt length (tokens)	32768
Max response length (tokens)	512
Total training epochs	150

model and matched with cosine similarity, retrieving the top-*k* most similar historical queries. For fine-grained retrieval, we further select the top-*M* relevant queries using an LLM-based relevance scorer and sparsify their interaction graphs with an LLM-based graph compression module. The values of *k* and *M* are treated as tunable hyperparameters. Query nodes are labeled with execution status from Failed, Resolved. All retrieved insights and interaction subgraphs are injected into agent prompts before task execution, and the memory graphs are updated after each task without any gradient-based training.

- **SkillRL** We follow SkillRL (Xia et al., 2026) for inference. The agent uses a hierarchical SkillBank with general and task-specific skills. For each query, it retrieves top-6 relevant task-specific skills (similarity threshold 0.4) using an embedding model (`text-embedding-ada-002`) and includes all general skills. Retrieved skills are added to the prompt, and the agent decodes with temperature 0.7. The skill library is fixed during testing.

D.2 Reinforcement Learning-based Setup

We conduct RL training on two types of interactive environments: ALFWorld and a suite of search-augmented question answering (QA) tasks. For ALFWorld, we use the standard training splits with 3553 training tasks. All reported results are averaged over 3 independent runs. For QA training, we follow prior work and merge the training sets of NQ and HotpotQA to form a unified dataset for ExpWeaver. Out-of-domain generalization is assessed on TriviaQA, 2Wiki, and Bamboogle, where we sample 100 test instances per dataset. Hyperparameters used for ExpWeaver RL training are summarized in Table 3. Our implementation is built on the verl-agent framework (Feng et al., 2025). Training runs for a fixed number of epochs or until validation performance converges.

E Additional Results with ExpWeaver

E.1 Additional Prompting-based Results

Results on QA tasks. Figures 10 and 11 present the prompting-based results on three knowledge-intensive QA tasks under the ReasoningBank and SkillRL frameworks, respectively. Overall, ExpWeaver consistently achieves the best or competitive performance across TriviaQA, HotpotQA, and 2Wiki, further confirming the effectiveness of interweaving experience utilization into the decision-making process.

Under ReasoningBank, ExpWeaver improves over the vanilla initialization-only strategy in most settings. The gains are particularly clear on TriviaQA and HotpotQA, where ExpWeaver achieves the best performance across nearly all backbone models. On 2Wiki, ExpWeaver also generally improves over rigid usage strategies, though the margin is smaller for stronger models whose baseline performance is already high. These results suggest that allowing agents to decide when to consult distilled reasoning experiences is beneficial for knowledge-intensive reasoning.

Under SkillRL, we observe similar trends despite the different form of experience. Since SkillRL uses a pre-constructed skill repository, all variants share the same underlying experience memory and differ only in how skills are utilized at runtime. ExpWeaver consistently improves over vanilla

ALFWorld

You are now in a household environment called Alfworld, and your tasks include locating objects, heating or cooling items, and other similar activities.

AVAILABLE COMMANDS (you can ONLY output ONE command at a time):
- Environment-interactive actions (where 'a' and 'b' are variables):
> take a from b
> go to a
> open a
> move a to b
> clean a with b
> heat a with b
> cool a with b
> use a
- Internal reasoning (does not affect environment):
> think: xxx

MEMORY ASSISTANCE:
- You have been provided with memory items from past experiences at the beginning. Use them as reference when needed.

CRITICAL RULES - STRICT ALTERNATING PATTERN:
1. You MUST output ONLY ONE command per turn
(either one "think" OR one action, never both at once).
2. Follow this simple rule to decide what to output:
- If the LAST message is an observation (from environment):
output "> think: [reasoning]"
- If the LAST message is your own think: output an environment action
(e.g., "> go to cabinet 1")
3. DO NOT output multiple commands in one response
(e.g., do NOT write "> think: xxx > go to cabinet 1").
4. DO NOT output consecutive "think" steps without an action in between.
5. You must check carefully that your output is exactly ONE of the allowed commands above!!!

You may take maximum of [MAX_STEPS] steps.

Here are some examples:
[FEWSHOTS]
(END OF EXAMPLES)

Now it's your turn!
[TASK GOAL]

Figure 7: System prompt on ALFWorld.

usage on TriviaQA and HotpotQA, and remains competitive on 2Wiki. This provides controlled evidence that the gains arise from better runtime experience utilization rather than changes in the experience repository itself. Together, these QA results complement the main results on ALFWorld and WebShop, showing that ExpWeaver generalizes beyond interactive environments to knowledge-intensive reasoning tasks.

Results on G-Memory. Figures 12 and 13 present the prompting-based results under the G-Memory framework. In the G-Memory framework, we did not design a variant that retrieves experience at every

QA

You are a question answering assistant. Your goal is to answer multi-hop questions by searching Wikipedia.

AVAILABLE ACTIONS (output ONE per turn):
- Search[entity]: searches Wikipedia for the entity and returns the first paragraph; if not found, returns similar entity suggestions
- Lookup[keyword]: returns the next sentence containing the keyword from the last successfully found passage
- Finish[answer]: returns your final answer and ends the task

REASONING FORMAT:
- Output "Thought N: ..." to reason about the current situation
- Then output "Action N: ..." to take an action
- Always strictly alternate: Thought → Action → Observation → Thought → Action → ...

OPTIONAL: Add [RETRIEVE] at the end of your Thought:
Thought N: your reasoning here [RETRIEVE]

When you add [RETRIEVE], the system will automatically provide relevant memory from past QA trials.
If [RETRIEVE] returns "[No relevant memory found]", do NOT use [RETRIEVE] again in the same task.

CRITICAL RULES:
1. Output ONLY ONE action per turn.
2. NEVER output two Thoughts in a row — a Thought must be immediately followed by an Action.
3. Use Lookup[keyword] to extract specific details from a page you searched.
4. Call Finish[answer] as soon as you have enough information to answer.

You may take maximum of [MAX_STEPS] steps.

Here are some examples:
[FEWSHOTS]
(END OF EXAMPLES)

Now it's your turn!
[TASK GOAL]

Figure 8: System prompt on QA tasks.

step (Always-on), as doing so would violate its design philosophy and core architecture—namely, building a hierarchical graph memory store with complete task trajectories as the central nodes. In our experiments, we only adjusted the timing of experience presentation during execution, without changing G-Memory's own experience mechanism. This allows us to independently examine the effect of experience presentation timing while preserving the unique design advantages of the G-Memory framework. Unlike the single-agent frameworks, G-Memory operates in a multi-agent setting and represents experience using both raw trajectories and distilled insights. Across ALFWorld, WebShop, and QA tasks, ExpWeaver generally improves over the vanilla initialization-only strategy,

Webshop

You are an online shopping assistant. Your goal is to find and purchase the item described in the instruction from an online store.

AVAILABLE COMMANDS (you can ONLY output ONE command at a time):
- Environment-interactive actions:
Action: search[query] - search for products using keywords
Action: click[target] - click a button, product ASIN link, or option value
- Internal reasoning:
Action: think[your reasoning here]

OPTIONAL: Add [RETRIEVE] tag inside your think to recall past shopping experience:
Action: think[your reasoning here [RETRIEVE]]

When you add [RETRIEVE] inside a think, the system intercepts it (no OK. returned)
and automatically provides relevant memory from past shopping trials. If [RETRIEVE] returns "
[No relevant memory found]", do NOT use [RETRIEVE] again in the same task.

CRITICAL RULES:
1. You MUST output ONLY ONE command per turn.
2. After each think (or think[RETRIEVE] + retrieved memory), output a search or click action.
3. Start with a targeted search using key product attributes from the instruction.
4. On the search results page, you must first click Back to Search before executing a new search[...]; otherwise, it will return an Invalid action error.
5. You won't get any score if you don't buy any item. So please MIND your max_steps limitation. if you purchase any item, you will get a score of range from 0.0 to 1.0 according to the similarity between the item and the instruction.

You may take maximum of [MAX_STEPS] steps.

Here are some examples:
[FEWSHOTS]
(END OF EXAMPLES)

Now it's your turn!
[TASK GOAL]

Figure 9: System prompt of WebShop.

showing that interweaving experience utilization into decision-making remains effective in multi-agent self-evolving systems.

On ALFWorld, ExpWeaver consistently improves performance across all models. The gains are especially clear for Qwen3.5-397B-A17B and DeepSeek-V4-Pro, suggesting that even strong models benefit when experience is introduced at appropriate decision points rather than only at the beginning. On WebShop, ExpWeaver also improves over the vanilla strategy in most cases, indicating that adaptive experience usage helps agents resolve interactive decision uncertainty in web navigation.

For QA tasks, ExpWeaver achieves consistent or competitive improvements across TriviaQA, HotpotQA, and 2Wiki. The gains are generally smaller than those on ALFWorld, which is consistent with our usage-pattern analysis showing that QA tasks require less frequent experience invocation. Overall, the G-Memory results demonstrate that ExpWeaver generalizes beyond single-agent settings and remains effective when experience is shared and utilized in a multi-agent memory framework.

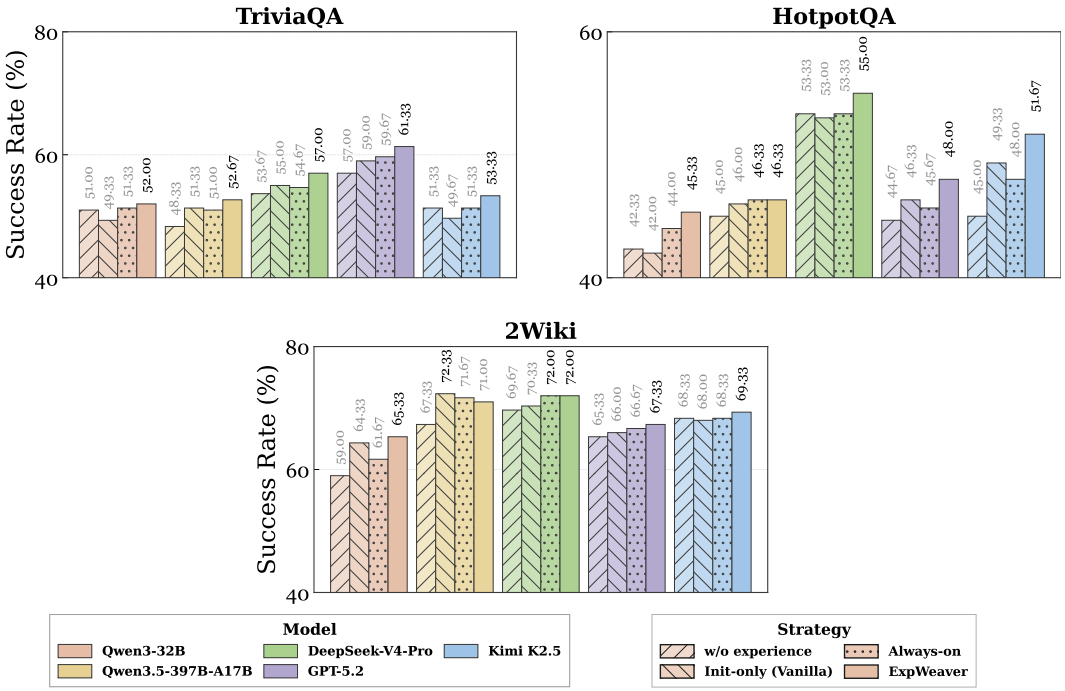


Figure 10: Prompting-based results under the ReasoningBank framework on three QA tasks.

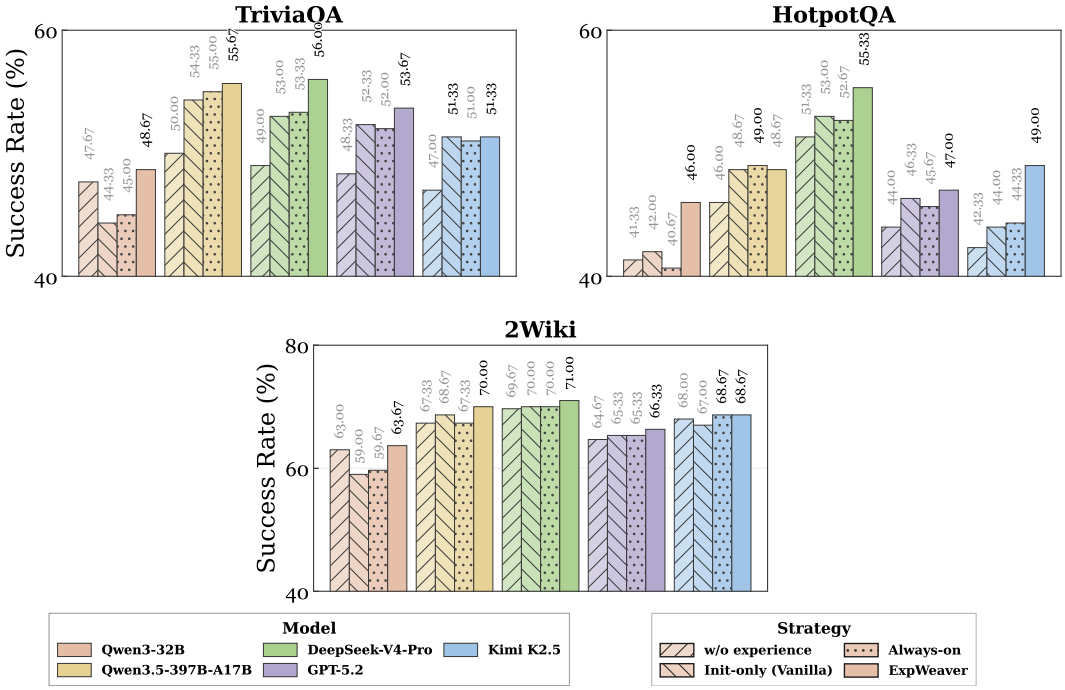


Figure 11: Prompting-based results under the SkillRL framework on three QA tasks.

Results on AWM. Figures 14 and 15 present the prompting-based results under the AWM framework. AWM represents experience as executable workflows and does not include an explicit retrieval mechanism during task execution. Therefore, the initialization-only and always-on variants are equivalent in implementation: the same workflow experience is provided as fixed guidance rather than dynamically retrieved at different steps.

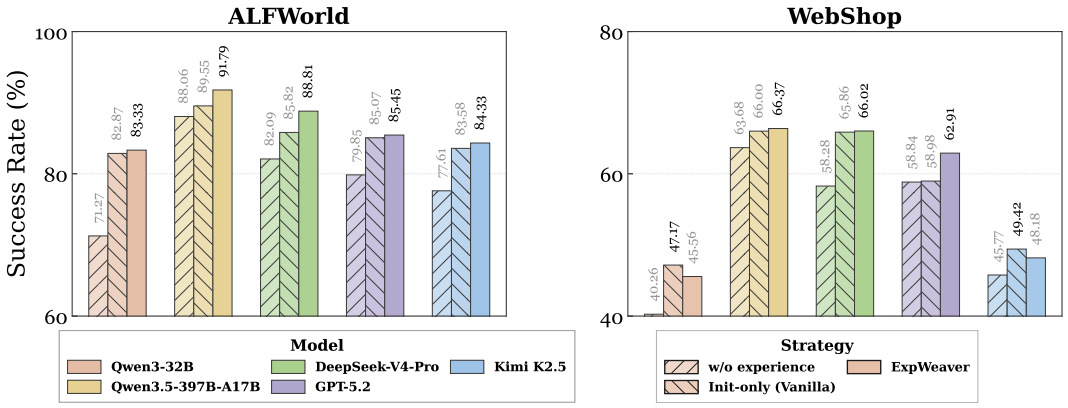


Figure 12: Prompting-based results under the G-Memory framework on ALFWorld and WebShop.

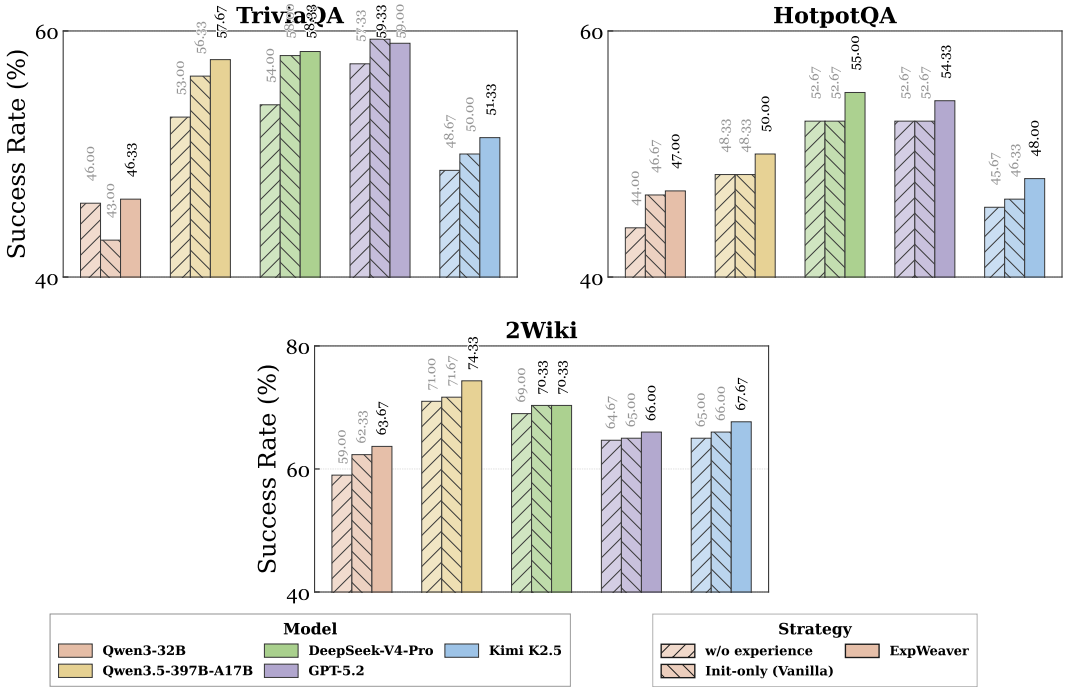


Figure 13: Prompting-based results under the G-Memory framework on three QA tasks.

Across ALFWorld, WebShop, and QA tasks, ExpWeaver generally improves over the vanilla AWM setting. The gains are especially clear on ALFWorld, where ExpWeaver consistently achieves higher success rates across different models, suggesting that even when experience is represented as workflows, allowing the agent to decide when to rely on experience remains beneficial. On WebShop and QA tasks, ExpWeaver also achieves competitive or improved performance in most settings.

These results further support our main conclusion: the benefit of ExpWeaver is not tied to retrieval-based memory alone. Even in a workflow-based framework without explicit retrieval, interweaving experience utilization into decision-making can improve how agents make use of existing experience.

E.2 Additional RL Results on QA Tasks

We further evaluate RL-based ExpWeaver on knowledge-intensive QA tasks using Qwen3-14B under the ReasoningBank framework. Specifically, we train the model with GRPO on a merged training set consisting of NQ and HotpotQA (Jin et al., 2025). During training, we periodically

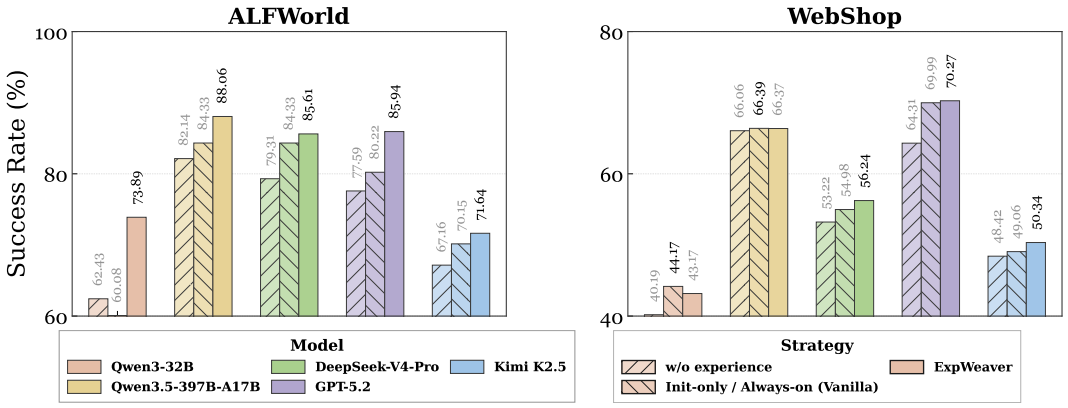


Figure 14: Prompting-based results under the AWM framework on ALFWorld and WebShop.

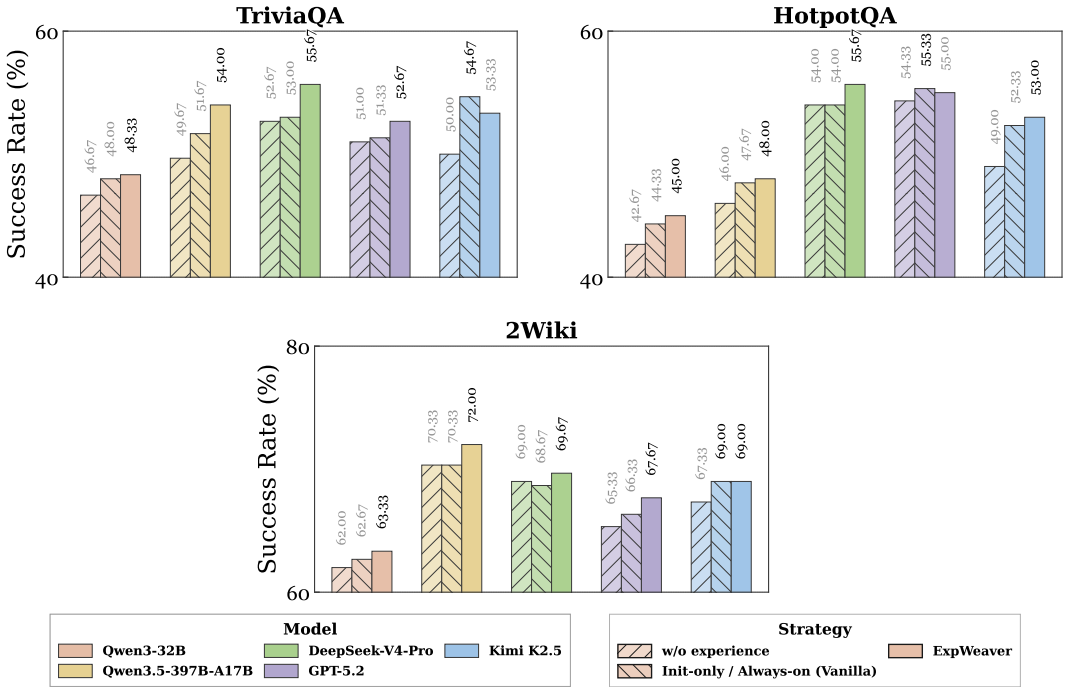
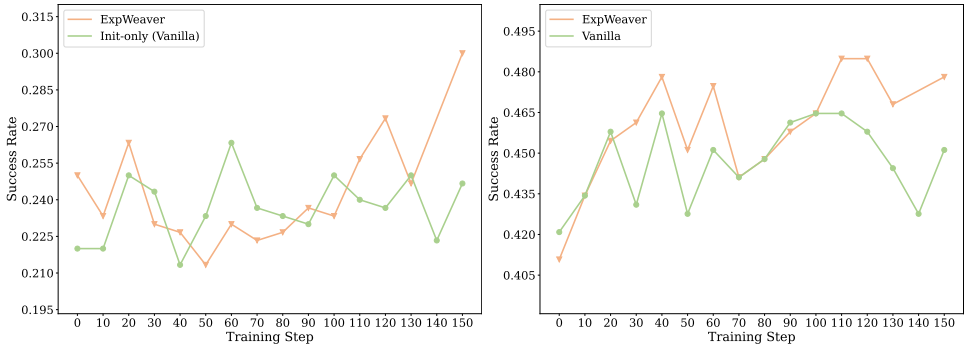


Figure 15: Prompting-based results under the AWM framework on three QA tasks.

evaluate intermediate checkpoints on five test sets: NQ, TriviaQA, HotpotQA, Bamboogle, and 2Wiki. Figure 16 reports the success rates of ExpWeaver and the vanilla baseline across training steps, where each point corresponds to the performance of a checkpoint on the corresponding test set.

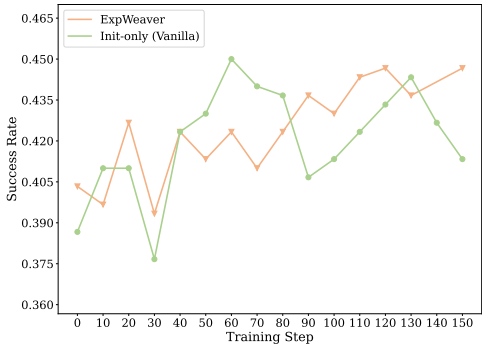
Overall, ExpWeaver achieves stronger improvements than the vanilla baseline (Init-only) across QA tasks. On NQ, ExpWeaver shows a clear upward trend and reaches a substantially higher success rate at later training steps, indicating that the model gradually learns to leverage experience more effectively during reasoning. Similar trends are observed on TriviaQA, Bamboogle, and 2Wiki, where ExpWeaver tends to achieve higher final performance.

These results suggest that the benefits of ExpWeaver are not limited to embodied interaction. Even in knowledge-intensive QA, where reasoning and evidence integration are central, allowing the agent to regulate when experience enters the reasoning process can improve RL optimization. This further supports our claim that interweaving experience utilization with decision-making is a learnable capability that generalizes beyond inference-time prompting and across task domains.

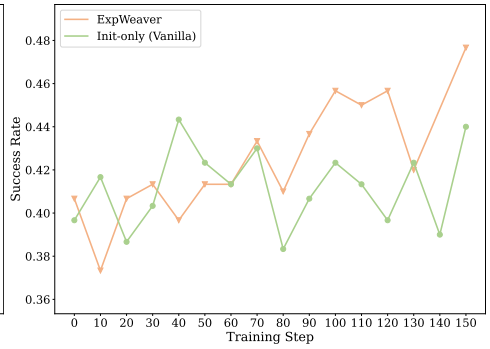


(a) Checkpoint evaluation on NQ.

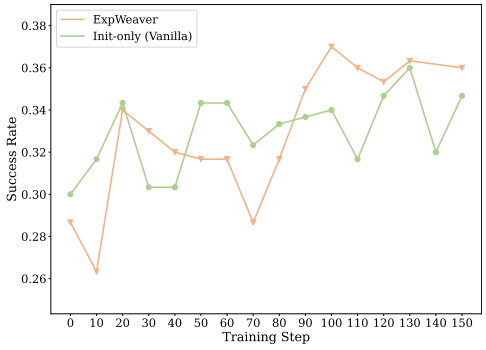
(b) Checkpoint evaluation on TriviaQA.



(c) Checkpoint evaluation on HotpotQA.



(d) Checkpoint evaluation on 2Wiki.



(e) Checkpoint evaluation on Bamboogle.

Figure 16: Reinforcement learning results of Qwen3-14B on knowledge-intensive QA tasks.

Table 4: Additional usage pattern results on AWM and SkillRL.

Model	AWM			SkillRL		
	ALFWorld	WebShop	HotpotQA	ALFWorld	WebShop	HotpotQA
Qwen3-32B	2.62	0.16	0.53	2.50	0.14	0.39
Qwen3.5-397B-A17B	0.47	0.00	0.00	0.28	0.00	0.02
DeepSeek-V4-Pro	0.50	0.58	0.33	0.60	0.42	0.11
GPT-5.2	0.00	0.00	1.14	0.00	0.00	0.00
Kimi-K2.5	1.20	0.00	0.00	0.88	0.03	0.08

F Usage Pattern Analysis of ExpWeaver

F.1 Additional Horizontal Comparison Results

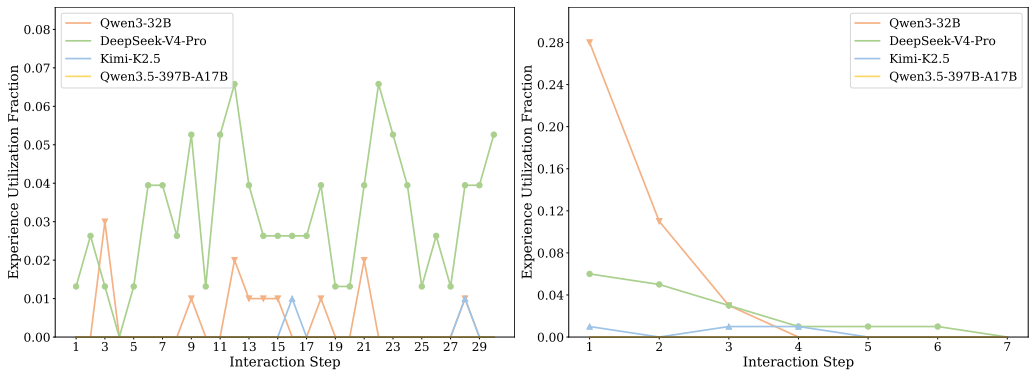
Table 4 reports the average number of experience retrievals per sample on AWM and SkillRL.

First, ALFWorld again induces substantially more experience retrievals than WebShop and HotpotQA. For example, under AWM, Qwen3-32B invokes experience 2.62 times per sample on ALFWorld, compared with 0.16 on WebShop and 0.53 on HotpotQA. Similarly, under SkillRL, Qwen3-32B retrieves experience 2.50 times per sample on ALFWorld, while retrieval frequencies remain below 0.40 on WebShop and HotpotQA. This reinforces the observation that embodied interaction, with long-horizon planning and partial observability, creates stronger demand for experience.

Second, stronger models tend to invoke experience less frequently. To be more specific, Qwen3.5-397B-A17B and GPT-5.2 exhibit near-zero retrieval on WebShop and HotpotQA under both AWM and SkillRL, suggesting that stronger backbones can often solve these tasks without additional experience. In contrast, smaller or less capable models rely on experience more often, especially in ALFWorld. This further supports the view that ExpWeaver adapts experience usage according to model capability rather than following a fixed retrieval schedule.

Third, the consistency across AWM and SkillRL indicates that the usage pattern is not specific to a particular form of experience. AWM represents experience as workflows, while SkillRL represents experience as reusable skills, yet both exhibit similar task- and model-dependent retrieval behavior. This suggests that interweaving experience utilization into decision-making induces a general regulatory mechanism over experience use, regardless of how experience is constructed or represented.

Together, these results reveal an important property of ExpWeaver: it does not simply reduce or increase experience usage uniformly. Instead, it produces a calibrated usage pattern conditioned on task difficulty, model capability, and decision context. This provides direct evidence that experience utilization should not be treated as an always-on operation; its value depends critically on when the agent chooses to invoke it.



(a) On WebShop. (b) On HotpotQA.

Figure 17: Temporal pattern of experience utilization under ExpWeaver with ReasoningBank.

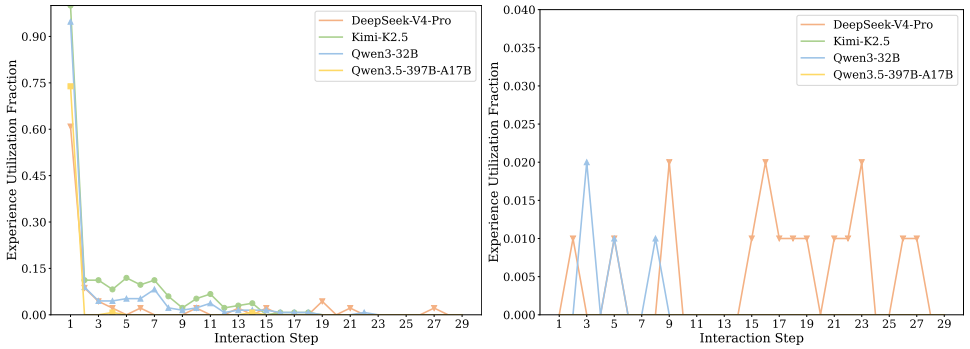
Table 5: The example of ExpWeaver on ALFWorld.

Task Goal: heat some apple and put it in garbagecan.

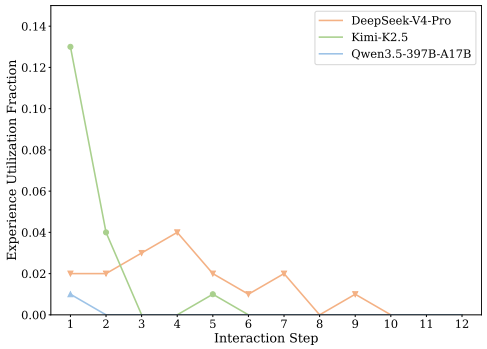
> think: I need to find an apple first. Apples are often found on diningtables, countertops, or in cabinets. Let me start by checking the diningtable.
> go to cabinet 1
You arrive at cabinet 1. On the cabinet 1, you see a glassbottle 1, and a soapbottle 1.
> think: No apple in cabinet 1. Let me check cabinet 2.
> go to cabinet 2
You arrive at cabinet 2. The cabinet 2 is closed.
> think: The cabinet 2 is closed. I need to open it to check for an apple.
> open cabinet 2
You open the cabinet 2. The cabinet 2 is open. In it, you see a plate 1.
> think: No apple found in cabinet 2 either. I have checked all visible locations without finding an apple. **I might need to recall where apples are typically found or consider other storage areas like the fridge. [RETRIEVE]**
<Retrieved Memory Items>
In past experiences, apples are often in the fridge or on the counter.
<End of Retrieved Memory>
> think: I have checked all visible locations without finding an apple. I might need to check the fridge next, as it is a common place to store apples. Let me go to the fridge.
> go to fridge 1
You arrive at fridge 1. The fridge 1 is closed.
> think: The fridge is closed. I need to open it to check for apples. Let me open the fridge.
> open fridge 1
You open the fridge 1. The fridge 1 is open. In it, you see a bowl 1, a bread 1, a cup 1, a lettuce 2, and a potato 1.
> think: No apple in the fridge either. I have exhausted all potential storage locations without finding an apple. **Perhaps I need to consider other possibilities or review past experiences to identify alternative search locations. [RETRIEVE]**
<Retrieved Memory Items>
The garbage can is in the kitchen. They should be checked.
<End of Retrieved Memory>
> go to garbagecan 1
You arrive at garbagecan 1. On the garbagecan 1, you see a apple 2, and a apple 1. > think: I found apples in the garbagecan. I need to take one, heat it, and then return it to the garbagecan. Let me first take one of the apples.
> take apple 1 from garbagecan 1
You pick up the apple 1 from the garbagecan 1.
> think: Now that I have the apple, I need to heat it. The most reliable way to heat items is using the microwave. Let me go to the microwave.
> go to microwave 1
You arrive at microwave 1. The microwave 1 is closed.
> think: The microwave is closed. I need to open it before I can heat the apple. Let me open it.
> open microwave 1
You open the microwave 1. The microwave 1 is open.
> think: I'm at the microwave and have the apple. Let me heat the apple using the microwave.
> heat apple 1 with microwave 1
You heat the apple 1 using the microwave 1.
> think: The apple is now hot. The original plan might have been to place it on a plate after heating. I see a plate 1 inside cabinet 2. Let me go there and put the hot apple on the plate.
> go to cabinet 2
> You arrive at cabinet 2. The cabinet 2 is open. In it, you see a plate 1.
> put apple 1 on plate 1
> Task is SOLVED.

Table 6: The example of ExpWeaver on WebShop.

Task Goal: I am looking for a long sleeve men t-shirt, color black, and price lower than 25.00 dollars.
> Think[I need a black long sleeve men's t-shirt under $25. I will search with the most specific attributes.] > Search[long sleeve men t-shirt black] > Observation: [Back to Search] [B09QQP3356] HAUKLIE Men's Sports Waffle Ribbed Long Sleeve Henley T-Shirt $10.99 > Think[B09QQP3356 is a long sleeve men's shirt at $10.99, well under $25. Let me check its color options.] > Click[B09QQP3356] > Observation: color [black][gray][light blue] size [small][medium][large][x-large][xx-large] HAUKLIE Men's Sports Waffle Ribbed Polo Shirts Summer Long Sleeve Cotton Muscle Quarter-Zip Henley T-Shirt Price: $10.99 > Think[The item has color and size options. The instruction requires black color. I will select black first, then pick a size. **I'm not sure whether I need to select size even when the instruction doesn't specify one. I'll search relevant experiences. [RETRIEVE]]** [Retrieved Memory Items] When an item page shows selectable options (size, color, flavor, etc.), always click each option to select it before clicking Buy Now. If the instruction does not specify a particular option value, select any available one (e.g., the first listed). Skipping option selection typically results in a wrong item or task failure. [End of Retrieved Memory] > Click[black] > Observation: You have clicked black. > Think[Black color selected. The instruction does not specify size, so I will choose the first available size.] > Click[small] > Observation: You have clicked small. > Think[Both color and size are selected. Price $10.99 is within the $25.00 budget. Ready to buy.] > Click[Buy Now] > Observation: Your score (min 0.0, max 1.0): 1.0



(a) On ALFWorld. (b) On WebShop.

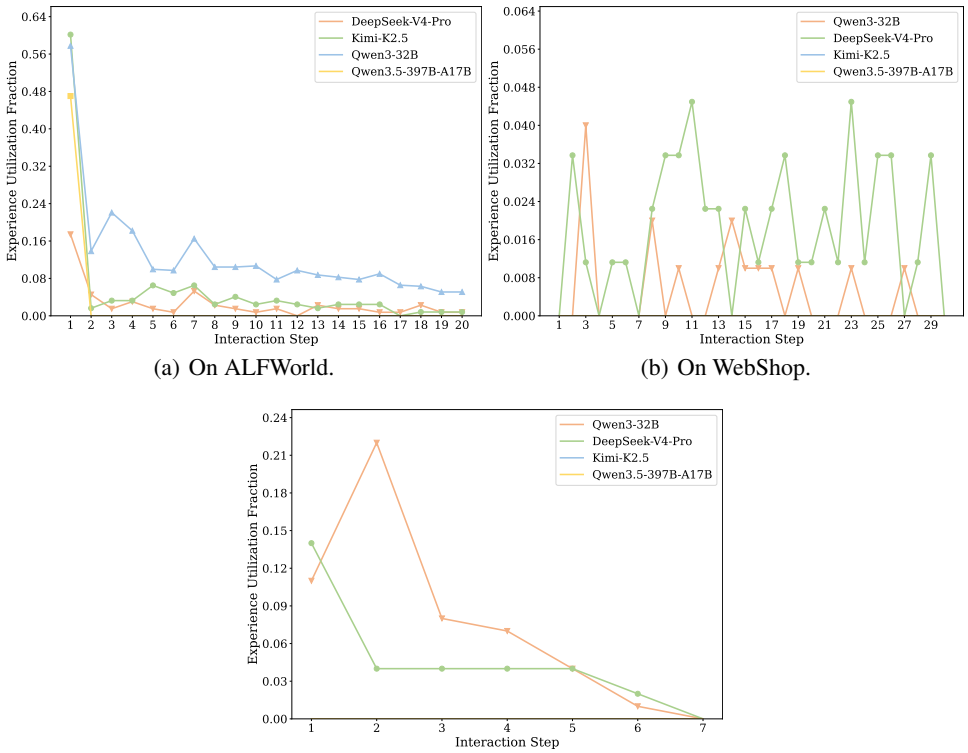


(c) On HotpotQA.

Figure 18: Temporal pattern of experience utilization under ExpWeaver with G-Memory.

Table 7: The example of ExpWeaver on QA task.

Task Goal: What profession does Nicholas Ray and Elia Kazan have in common?
> Thought 1: I need to search Nicholas Ray and Elia Kazan, find their professions, then find the profession they have in common. > Action 1: Search[Nicholas Ray] > Observation 1: Nicholas Ray (born Raymond Nicholas Kienzle Jr., August 7, 1911 – June 16, 1979) was an American film director, screenwriter, and actor best known for the 1955 film Rebel Without a Cause. > Thought 2: **I have one side's professions, but I should retrieve a quick comparison checklist to avoid missing overlaps. [RETRIEVE]** [Retrieved Memory Items] Strategy: For profession-comparison questions, extract profession sets for both entities separately, then return only the intersection. [End of Retrieved Memory] > Thought 3: Following the checklist, I now need Elia Kazan's professions and then compute the intersection. > Action 3: Search[Elia Kazan] > Observation 3: Elia Kazan was an American film and theatre director, producer, screenwriter and actor. > Thought 4: The intersection is director, screenwriter, and actor. > Action 4: Finish[director, screenwriter, actor]



(a) On ALFWorld. (b) On WebShop.

(c) On HotpotQA.

Figure 19: Temporal pattern of experience utilization under ExpWeaver with AWM.

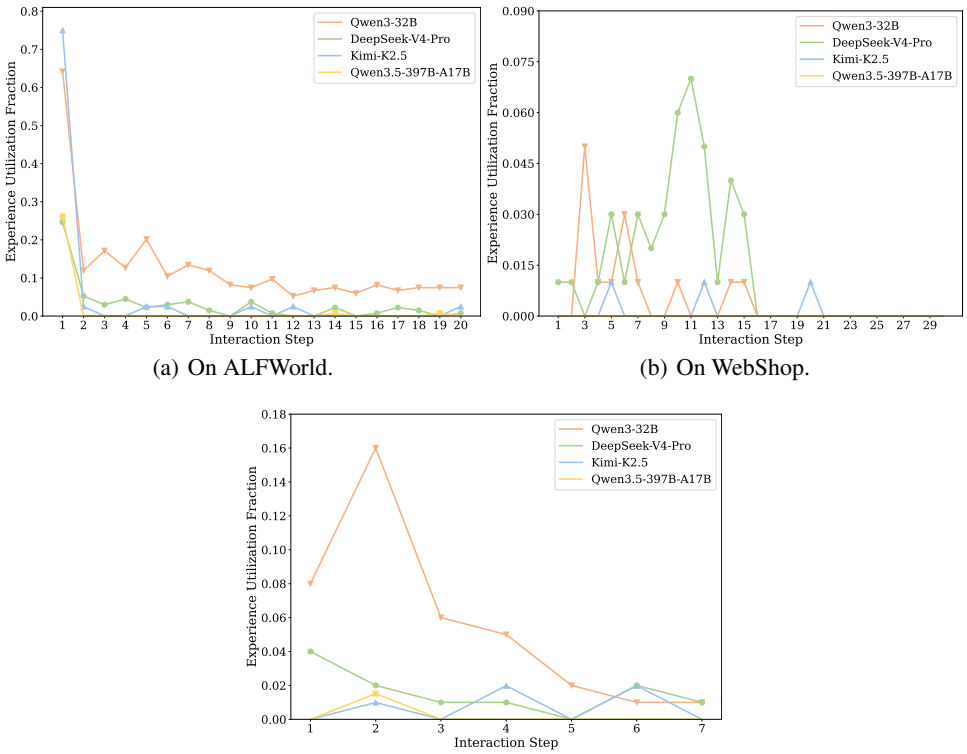
F.2 Additional Temporal Analysis

We provide additional temporal analyses of experience utilization across different self-evolving frameworks and tasks. Figures 17–20 report the fraction of trajectories that invoke experience at each interaction step under ReasoningBank, G-Memory, AWM, and SkillRL, respectively. Overall, these results show that the non-uniform temporal pattern observed in the main text is consistent across frameworks, tasks, and backbone models.

Across ALFWorld, experience utilization is highly concentrated at the beginning of trajectories. This trend appears consistently under G-Memory, AWM, and SkillRL, where most retrievals occur in the first few interaction steps and then rapidly decrease. This suggests that agents tend to consult experience when initial task understanding, high-level planning, and environment exploration are most needed. Notably, some models continue to invoke experience sporadically in later steps, especially in ALFWorld, which often corresponds to object-search or recovery phases under partial observability.

On WebShop, the temporal patterns are more dispersed and generally sparser. Compared with ALFWorld, agents invoke experience less frequently, and retrieval events are often scattered across later interaction steps. This reflects the different nature of web navigation, where the agent may not need experience at the beginning but may consult it when encountering uncertain product search, comparison, or navigation decisions. The scattered peaks indicate that experience is used as situational guidance rather than as a fixed initialization signal.

On HotpotQA, experience utilization is typically concentrated in early reasoning steps and quickly diminishes afterward. This is especially clear for models with higher retrieval frequency, where experience is often invoked during initial question decomposition or evidence-seeking. Once the agent establishes a reasoning direction, later retrieval becomes less frequent. This pattern supports the view that experience is mainly used to guide early reasoning or resolve uncertainty, rather than being repeatedly injected throughout the entire reasoning process.



(a) On ALFWorld. (b) On WebShop.

(c) On HotpotQA.

Figure 20: Temporal pattern of experience utilization under ExpWeaver with SkillRL.

Importantly, these trends hold across different experience representations, including distilled insights in ReasoningBank, raw trajectories and distilled insights in G-Memory, workflows in AWM, and reusable skills in SkillRL. Although the absolute retrieval frequency varies across frameworks and models, the temporal structure remains consistent: ExpWeaver invokes experience selectively, with strong early-stage usage and occasional later-stage reactivation when the agent faces uncertainty or difficulty. These results further support our claim that interweaving experience utilization with decision-making enables agents to regulate when experience enters the reasoning process, rather than following rigid initialization-only or always-on usage strategies.

F.3 Qualitative Examples

We provide qualitative examples to illustrate how ExpWeaver invokes experience during decision-making across different environments. These cases show that experience retrieval is not triggered uniformly, but is typically invoked when the agent encounters uncertainty, ambiguity, or repeated failure during task execution.

ALFWorld. Table 5 presents an example from ALFWorld, where the agent needs to heat an apple and place it in the garbage can. The agent first searches several plausible locations, such as cabinets and the fridge, but fails to find the target object. After repeated unsuccessful search attempts, ExpWeaver triggers experience retrieval to seek guidance about alternative object locations. The retrieved experience suggests that the garbage can should be checked, which directly helps the agent locate the apple and continue the task. This example shows that ExpWeaver invokes experience when the agent is stuck in a partially observable environment and needs corrective guidance.

WebShop. Table 6 shows a WebShop example where the agent must purchase a black long-sleeve men's t-shirt under a price constraint. After finding a candidate item, the agent becomes uncertain about whether unspecified options such as size should also be selected before buying. ExpWeaver

retrieves relevant experience indicating that all selectable options should be clicked before purchasing, even when some options are not specified in the instruction. Guided by this experience, the agent selects the required color, chooses an available size, and successfully completes the purchase. This case shows that experience is useful for resolving procedural uncertainty in web environments.

Knowledge-intensive QA. Table 7 provides an example from a QA task asking for the profession shared by Nicholas Ray and Elia Kazan. After retrieving information about one entity, the agent recognizes a risk of missing overlapping professions and invokes experience to obtain a comparison checklist. The retrieved experience reminds the agent to extract profession sets for both entities separately and return their intersection. Following this, the agent searches for the second entity and correctly computes the shared professions. This illustrates how ExpWeaver helps regulate reasoning in knowledge-intensive tasks by retrieving experience when the agent finds a reasoning pitfall.

Overall, these examples support our temporal and entropy-based analyses: ExpWeaver tends to invoke experience at moments where the agent faces uncertainty or requires additional guidance. The retrieved experience then serves as targeted support for the current decision, rather than as static context uniformly injected throughout the trajectory.