StepPO: Step-Aligned Policy Optimization for Agentic Reinforcement Learning

Anonymous ACL submission

Abstract

Agentic reinforcement learning (RL) is emerging as a critical post-training paradigm for improving LLM agent capabilities. Existing RL algorithms for LLMs largely follow the token-centric paradigm as in RLHF and RLVR, where tokens serve as the basic units for modeling and optimization. However, this paradigm introduces a granularity mismatch in agentic RL, as it optimizes token-level predictions while LLM agents make step-level decisions through cycles of environmental observations and actions. To bridge this gap, we propose **StepPO**, a step-centric paradigm for agentic RL via step-aligned policy optimization. Specifically, we reformulate agentic RL from a token-level Markov Decision Process (MDP) into a step-level MDP, where interaction steps serve as the basic trajectory representations. We further propose step-level credit assignment to align policy optimization with the natural granularity of agent decisions. Together, StepPO optimizes agent policies at the step level for multi-turn agent-environment interaction. Experiments across multi-hop QA, academic paper search, and text-world action tasks show that StepPO consistently outperforms various RL algorithms. Further analyses provide insights into how step-centric paradigm improves agent training. We hope this step-centric paradigm offers a useful lens for understanding agent behavior and a practical path for training more capable LLM agents. Our code is available[1].

1 Introduction

Large language model (LLM)-based agents have given rise to phenomenal applications (e.g., OpenClaw, Claude Code) (Achiam et al., 2023; Bai et al., 2023; openclaw, 2026; Anthropic, 2025). These LLM agents are moving beyond single-turn question answering toward autonomous planning, multi-turn tool use and iterative interaction with external environment (Yao et al., 2022b; Schick et al., 2023; Cheng et al., 2026). Agentic reinforcement learning (RL) has therefore become a critical post-training paradigm for improving such capabilities (Cheng et al., 2025; Zhang et al., 2025). By optimizing policies over multi-turn interaction trajectories, agentic RL allows models to improve their decision-making from environmental feedback (Zhou et al., 2025; Wang et al., 2025).

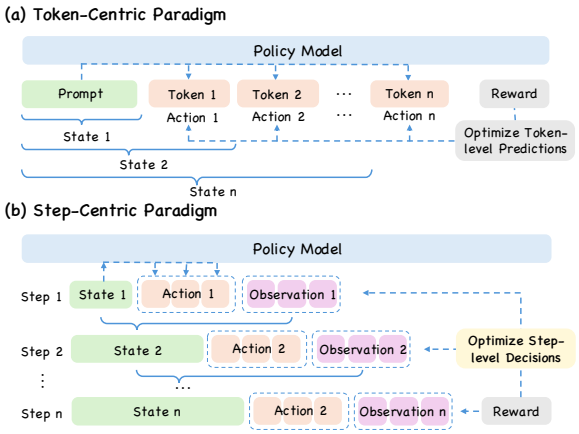

Figure 1: Comparison between token-centric and step-centric paradigms, where the basic modeling and optimization unit shifts from tokens to interaction steps.

Most existing agentic RL methods inherit the RL algorithms for LLMs. Representative methods include Proximal Policy Optimization (PPO) (Schulman et al., 2017) from the reinforcement learning from human feedback (RLHF) paradigm (Ouyang et al., 2022), and Group Relative Policy Optimization (GRPO) (Shao et al., 2024) from the reinforcement learning with verifiable rewards (RLVR) paradigm (Guo et al., 2025). These methods are largely token-centric. On the modeling side, they commonly view LLM generation as a token-level Markov Decision Process (MDP), where tokens are the basic units for formalizing states, actions, and policy updates. On the credit assignment side, PPO typically uses Generalized Advantage Estima-

[1] https://anonymous.4open.science/r/StepPO/

Table 1: Representative methods differ in how they place the MDP formulation and credit assignment units, revealing a granularity mismatch in agentic RL. StepPO reduces this mismatch by aligning both with the **step**.

MDP Formulation Granularity	Credit Assignment Granularity	Representative Methods
Token-level	Token-level	PPO (Schulman et al., 2017), Reinforce++ (Hu, 2025)
Token-level	Trajectory-level	GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024)
Step-level	Trajectory-level	GiGPO (Feng et al., 2026), LightningRL (Luo et al., 2025)
Step-level	**Step-level**	**StepPO**

tion (GAE) to estimate token-level advantages from critic-based temporal-difference residuals (Schulman et al., 2015), while GRPO uses Group-Relative Advantage Estimation (GRAE) to derive critic-free advantages from sampled trajectories and broadcasts each trajectory-level advantage to all tokens (Hu et al., 2026). These methods have brought substantial progress in preference alignment and verifiable-reward reasoning.

However, LLM agents do not interact with environments one token at a time. The actual medium of interaction is a step, where the agent receives an observation, generates a complete response that may be parsed as a tool call, and moves to the next state based on environmental feedback. This discrepancy introduces a granularity mismatch: existing RL algorithms organize modeling and optimization around tokens, whereas agent decisions take effect at the step level, as shown in Table 1. To bridge this gap, we propose **StepPO**, a step-centric paradigm for agentic RL that aligns RL modeling and optimization with the natural interaction granularity of LLM agents. Specifically, as illustrated in Figure 1, StepPO first reformulates agentic RL from a token-level MDP into a step-level MDP, where interaction steps serve as basic trajectory representations. Under this formulation, the policy receives the current step state, generates a complete action, obtains reward and observation, and transitions to the next step state.

StepPO further performs credit assignment at step granularity by propagating rewards across interaction steps, and finally applies PPO-style policy optimization over step-level actions. This step-level design enables more suitable credit assignment for multi-turn agent behaviors, since token-level credit is often too local to capture the effect of complete actions on subsequent states, while trajectory-level credit is too coarse to identify key intermediate decisions in long-horizon tasks. In this way, the MDP formulation, trajectory representation, and credit assignment unit are all aligned with the natural interaction unit of LLM agents.

We evaluate StepPO across multi-hop question answering (QA) (Yang et al., 2018), agentic academic paper search (He et al., 2025), and text-world action tasks including ALFWorld and WebShop (Shridhar et al., 2020; Yao et al., 2022a). Experimental results show that StepPO consistently outperforms representative RL baselines, including PPO, GRPO, and other methods with different MDP formulations and credit assignment strategies. Further analyses show that step-centric optimization improves decision quality in multi-turn interaction, offering a useful perspective for understanding agent behavior and a practical path toward training more capable LLM agents.

In summary, our contributions are as follows:

- We identify the granularity mismatch between token-level optimization and step-level agent decisions, and reformulate agentic RL as a step-level MDP.
- We propose StepPO, a step-aligned policy optimization method that combines step-native trajectory representation and step-level credit assignment over complete interaction steps.
- Experiments across various agentic scenes show consistent improvements over compared baselines and provide insights into step-centric training for LLM agents.

2 Related Work

2.1 Reinforcement Learning for LLMs

RL has become a major post-training paradigm for optimizing LLMs with feedback from human preferences and verifiable outcomes. Early LLM RL is largely built on PPO-based RLHF, where learned preference rewards guide policy improvement (Arjona-Medina et al., 2019; Ouyang et al., 2022). DPO offers a simpler preference-optimization alternative without online RL updates (Rafailov et al., 2023). More recent works prefer critic-free methods: RLOO estimates leave-one-out advantages using other samples from the

same prompt group (Ahmadian et al., 2024), REINFORCE++ adds practical stabilization while retaining a critic-free design (Hu, 2025), and GRPO estimates group-relative advantages for verifiable-reward reasoning (Shao et al., 2024). This path continues with reasoning-oriented variants of PPO and GRPO (Wang et al., 2026; Yu et al., 2026).

2.2 Agentic Reinforcement Learning

Agentic RL trains LLM agents through multi-turn interaction with tools and environmental feedback, where long horizons, sparse rewards, evolving observations, and branching traces become central challenges (Zhang et al., 2025; Wang et al., 2025). Early explorations instantiate these challenges in retrieval and multi-turn reasoning settings through end-to-end RL training (Jin et al., 2025; Wang et al., 2025; Cheng et al., 2025). Recent algorithms further adapt RL to agent execution structure: Tree-GRPO organizes exploration through tree-structured rollouts (Ji et al., 2025), PSPO uses process-aware trajectory-level optimization for academic paper search (Pan et al., 2026), and GiGPO introduces group-in-group credit assignment for agent trajectories (Feng et al., 2026). Nevertheless, existing methods still largely organize optimization around tokens, complete responses, or full trajectories, while agent decisions take effect through environment-facing interaction steps. StepPO treats this mismatch as the basis for a paradigm shift toward step-centric agentic RL, where the interaction step becomes the shared unit for the MDP formulation and credit assignment.

3 Preliminary

3.1 Token-Level MDP Formulation

Most RL algorithms for LLMs inherit the next-token prediction interface and therefore formulate policy optimization at token granularity. Given a prompt x and previously generated tokens $y_{<i}$, the policy model samples the next token y_i from its next-token distribution. In this formulation, each prefix defines a token-level state s_i, and each sampled token serves as a token-level action a_i:

$$s_i = (x, y_{<i}), \qquad a_i = y_i, \tag{1}$$

and the transition deterministically appends the sampled token to the prefix, yielding $s_{i+1} = (x, y_{\le i})$. The episode terminates when the model generates an end-of-sequence token y_L. Since rewards are usually provided at the end of a trajectory, token-level RL methods assign credit to individual tokens through advantage estimation.

3.2 Existing Credit Assignment Paradigms

For token-level policy optimization, existing RL methods mainly follow credit assignment in two paradigms. The first paradigm, represented by PPO (Schulman et al., 2017), uses a critic model to estimate token-level advantages through GAE (Schulman et al., 2015). Under the token-level formulation, the advantage $\hat{A}_i^{\mathrm{GAE}}$ can be written as:

$$\begin{aligned} \hat{A}_i^{\mathrm{GAE}} &= \sum_{l=0}^{L-i} (\gamma\lambda)^l \delta_{i+l}, \\ \delta_i &= r_i + \gamma V_\phi(s_{i+1}) - V_\phi(s_i). \end{aligned} \tag{2}$$

where V_ϕ estimates the value of each token-level state, δ_i measures its temporal-difference (TD) error under reward r_i, and γ, λ denote the discount factor and GAE trace parameter.

The second paradigm, represented by GRPO (Shao et al., 2024), avoids a critic by estimating trajectory-level advantages from grouped rollouts, termed GRAE. Given a prompt x, suppose the policy samples a group of N trajectories $\{\tau_j\}_{j=1}^N$ with returns $\{R_j\}_{j=1}^N$. The group-relative advantage $\hat{A}_k^{\mathrm{GRAE}}$ for trajectory τ_k can be written as:

$$\hat{A}_k^{\mathrm{GRAE}} = R_k - \bar{R}, \qquad \bar{R} = \frac{1}{N}\sum_{j=1}^{N} R_j. \tag{3}$$

Here, $\bar{R}$ is the mean return of rollouts sampled for the same prompt, while practical GRPO variants may include additional reward normalization. GRAE captures a trajectory-level credit signal, which is shared by all tokens within the trajectory.

4 StepPO

In this section, we present StepPO, which aligns agent decisions with MDP formulation and credit assignment at the step level.

4.1 Step-Level MDP Formulation

StepPO formulates agent execution as a step-level MDP. At interaction step t, the state $s_{1:M_t}^{(t)}$ contains the interaction history available to the agent, including previous observations and actions. The action $a_{1:L_t}^{(t)}$ is a complete environment-facing response generated by the policy. In a common ReAct-style setting, this action contains reasoning and tool calls tokens, such as search queries or text-world actions.

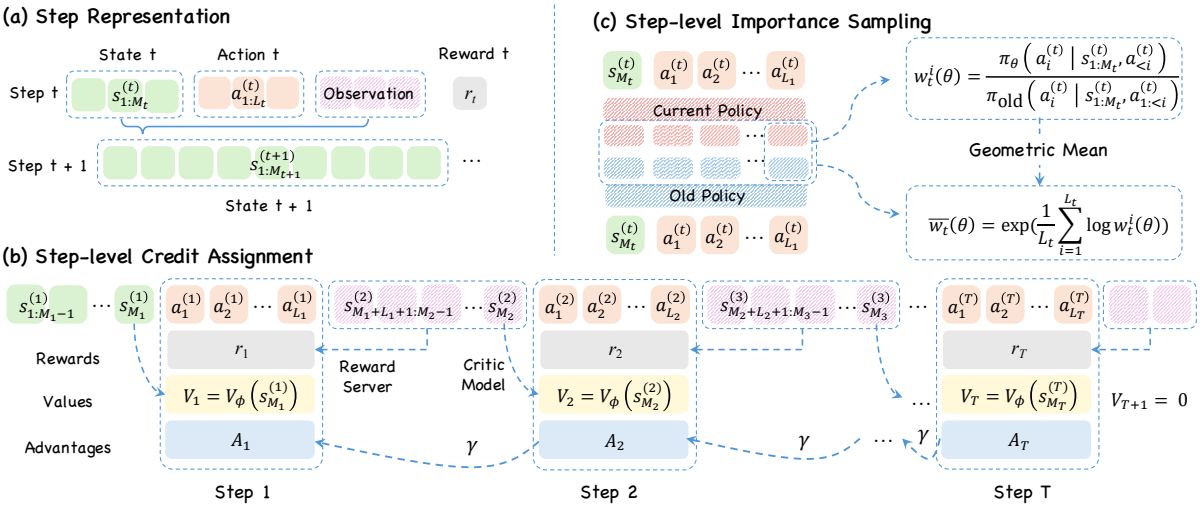

Figure 2: Overview of StepPO. StepPO reformulates agentic RL as a step-level MDP with step-native trajectory representation, step-level credit assignment, and step-level importance sampling for multi-turn optimization.

The policy observes $s^{(t)}_{1:M_t}$, emits $a^{(t)}_{1:L_t}$, receives reward r_t, and then transitions to the next state $s^{(t+1)}_{1:M_{t+1}}$. These form a step-level trajectory τ, and the resulting step-level policy objective $J(\theta)$ is:

$$\tau = \{(s^{(t)}_{1:M_t}, a^{(t)}_{1:L_t}, r_t, s^{(t+1)}_{1:M_{t+1}})\}_{t=1}^{T},$$
$$J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=1}^{T}\gamma^{t-1} r_t\right], \tag{4}$$

where T is the trajectory horizon and π_θ is the policy parameterized by θ. During autoregressive decoding, newly generated tokens are incrementally appended to the current action prefix until the action terminates. The environment transition occurs only after the complete action is executed and the returned observation tokens are incorporated into $s^{(t+1)}_{1:M_{t+1}}$. This differs from the token-level MDP in Eq. 1, where each newly generated token immediately induces a state transition.

Step-level Trajectory Representation. In practice, StepPO stores a full trajectory as step-native records rather than flattening it into a single token sequence. As shown in Figure 2 (a), each record contains the state tokens $s^{(t)}_{1:M_t}$, the action tokens $a^{(t)}_{1:L_t}$, and the reward r_t, which is reserved for process rewards. Operationally, the environment observation returned after executing $a^{(t)}_{1:L_t}$ is stored by incorporating it into the next record's state tokens $s^{(t+1)}_{1:M_{t+1}}$. This design keeps each replay unit aligned with an MDP transition while preserving the token-level likelihoods required by the trainer.

4.2 Step-Level Credit Assignment

Given step-native records, StepPO computes credit at the same granularity as agent decisions. In RL, the advantage compares the expected return after taking a specific action with the expected return from the same state before that action is specified. The latter is exactly the value estimated by either a learned critic model or a critic-free estimator. Therefore, in a step-level MDP, the value of $s^{(t)}_{1:M_t}$ should also be estimated before $a^{(t)}_{1:L_t}$ is generated. StepPO uses $V_\phi(s^{(t)}_{M_t})$, namely the value at the final state token before the action starts, as the value of this state. The reward r_t denotes the reward assigned to the interaction step, which may come from intermediate environment feedback and aggregated token-level rewards. As shown in Figure 2 (b), arranging the records by their order within each trajectory then yields a step timeline on which step-level advantage $\hat{A}^{\text{Step}}_t$ can be computed as:

$$\hat{A}^{\text{Step}}_t = \sum_{l=0}^{T-t}(\gamma\lambda)^l \delta_{t+l},$$
$$\delta_t = r_t + \gamma V_\phi(s^{(t+1)}_{M_{t+1}}) - V_\phi(s^{(t)}_{M_t}), \tag{5}$$

where δ_t is the step-level TD residual. The resulting advantage $\hat{A}^{\text{Step}}_t$ is then broadcast back to the valid generated tokens of the same step for the PPO-style actor update. Compared with token-level GAE, this avoids spreading delayed reward over surface tokens that are not themselves decisions. Compared with trajectory-level relative advantages, it preserves the ability to distinguish useful and harmful intermediate steps.

Step-Level Actor Objective. The actor objective also follows step boundaries during policy optimization. Since a step action $a^{(t)}_{1:L_t}$ contains multiple generated tokens, directly multiplying token importance sampling ratios would make longer actions have systematically more extreme ratios. As presented in Figure 2 (c), StepPO therefore uses the geometric mean of token ratios as the length-normalized step-level importance ratio $\bar{w}_t(\theta)$:

$$\begin{aligned} \bar{w}_t(\theta) &= \exp\left(\frac{1}{L_t}\sum_{i=1}^{L_t}\log w_t^i(\theta)\right), \\ w_t^i(\theta) &= \frac{\pi_\theta(a_i^{(t)} \mid s^{(t)}_{1:M_t}, a^{(t)}_{<i})}{\pi_{\text{old}}(a_i^{(t)} \mid s^{(t)}_{1:M_t}, a^{(t)}_{1:<i})}, \end{aligned} \tag{6}$$

where π_{old} denotes the rollout policy and $w_t^i(\theta)$ is the i^{th} token importance sampling ratio in the t^{th} step. Let $\text{clip}_\epsilon(\cdot)$ denote clipping to $[1-\epsilon, 1+\epsilon]$. The clipped step-level actor objective $\mathcal{J}_{\text{actor}}(\theta)$ is:

$$\begin{aligned} \mathcal{J}_{\text{actor}}(\theta) = \mathbb{E}_{a^{(t)}\sim\pi_{\text{old}}(\cdot|s^{(t)})}\Big[\min\Big(\\ \bar{w}_t(\theta)\hat{A}_t^{\text{Step}}, \text{clip}_\epsilon\left(\bar{w}_t(\theta)\right)\hat{A}_t^{\text{Step}}\Big)\Big]. \end{aligned} \tag{7}$$

This design makes the step-level MDP operational throughout the RL pipeline. The rollout is organized as step-level transition records, the critic estimates values at $s^{(t)}_{M_t}$, GAE propagates rewards across interaction steps, and the actor loss remains compatible with token-level likelihood training in standard LLM RL frameworks. StepPO therefore directly optimizes agent policies at the step level for multi-turn agent-environment interaction.

5 Experiments

In this section, we evaluate StepPO across four representative agentic scenarios, present its overall performance, and conduct in-depth analyses to provide further insights.

5.1 Experimental Setup

Backbones. We instantiate StepPO on two backbone models, Qwen3-1.7B and Qwen3-4B-Instruct-2507 (Yang et al., 2025). Across all tasks, we adopt an OpenAI-compatible tool calling format, with task-specific tool schemas. The runtime prompt specifies the current observation, interaction history, available tools, and expected output format. Appendix B details the complete prompt templates.

Benchmarks. We evaluate four agentic scenes. For multi-hop QA, we use HotpotQA (Yang et al., 2018) as the in-domain benchmark and evaluate out-of-domain generalization on 2Wiki (Ho et al., 2020) and MuSiQue (Trivedi et al., 2022). The agent interacts through Wikipedia search actions and terminates by producing a short answer; we report answer accuracy (Acc). For academic paper search, we evaluate on RealResearchQuery (He et al., 2025) following the multi-turn retrieval setting of PaperScout (Pan et al., 2026). The action space consists of paper search and citation/reference expansion actions over a maintained paper pool; we report post-threshold F1@all and Recall@all. For ALFWorld (Shridhar et al., 2020), the agent completes household tasks in text-based embodied environments by selecting admissible textual commands at each step; we report win rates on seen and unseen validation splits. For WebShop (Yao et al., 2022a), the agent navigates a text-based shopping website through page-conditioned actions such as search, product clicks, option clicks, and purchase; we report the average task score (Score) and purchase success rate (Succ.).

Baselines. We compare against two families of baselines. The prompting baselines include the base model without RL fine-tuning and ReAct (Yao et al., 2022b). The RL baselines include PPO (Schulman et al., 2017), Reinforce++ (Hu, 2025), GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), and GiGPO (Feng et al., 2026). Within each benchmark, RL baselines share the same backbone model, training data, task description, tool interface, hyperparameter budget, and evaluation protocol; they differ only in their RL algorithm design.

Training Details. Our implementation follows the Agent-R1 agent-training framework (Cheng et al., 2025) and uses veRL as the backend RL framework (Sheng et al., 2025). Experiments are run on a server with 8 NVIDIA H100 GPUs. Unless otherwise specified, we set the actor learning rate to 1×10^{-6}, the critic learning rate to 1×10^{-5} for critic-based methods, the training batch size to 128, and the actor micro-batch size to 4 per GPU. Group-based baselines use 8 rollouts per prompt and a batch size of 16, yielding an effective batch size of 128 for fair comparison. We use the same discount factor $\gamma = 0.99$, GAE trace parameter $\lambda = 1.0$ across tasks and set the actor-side KL regularization coefficient to 0.001. Detailed parameters are provided in Appendix A.

Table 2: Main results across multi-hop QA, academic paper search, ALFWorld, and WebShop. HotpotQA is in-domain (†), while 2Wiki and MuSiQue are out-of-domain (⋆). MDP and Credit Ass. denote the granularity of decision modeling and credit assignment. **Bold** means the best and underline is the second best.

Method	MDP	Credit Ass.	HotpotQA†	2Wiki⋆	MuSiQue⋆	RealResearchQuery		ALFWorld		WebShop	
						F1@all	Recall@all	Seen	Unseen	Score	Succ.
Qwen3-1.7B											
+ ReAct	–	–	3.62	2.24	0.50	0.005	0.012	2.86	2.24	2.03	0.80
+ PPO	Token	Token	38.00	50.12	16.55	0.284	0.514	67.14	69.40	59.12	34.60
+ Reinforce++	Token	Token	37.92	49.01	17.09	0.281	0.521	65.00	68.66	61.34	35.40
+ GRPO	Token	Traj.	36.76	48.30	16.88	0.275	0.530	73.57	75.37	63.15	36.20
+ RLOO	Token	Traj.	37.41	47.26	15.93	0.279	0.536	71.43	73.88	65.74	35.00
+ GSPO	Token	Traj.	39.12	46.83	17.46	0.264	0.541	67.86	67.91	59.03	32.40
+ GiGPO	Step	Traj.	40.85	52.43	18.37	0.298	0.545	70.00	69.40	66.92	41.80
+ StepPO	**Step**	**Step**	**44.86**	**56.17**	**21.56**	**0.314**	**0.551**	**75.00**	**79.10**	**69.88**	**45.00**
Qwen3-4B-Instruct-2507											
+ ReAct	–	–	37.45	48.59	10.26	0.171	0.193	7.14	2.99	51.58	23.80
+ PPO	Token	Token	56.75	58.92	19.82	0.303	0.531	76.43	72.39	70.18	46.00
+ Reinforce++	Token	Token	55.94	60.48	21.72	0.286	0.548	72.86	71.64	67.84	47.20
+ GRPO	Token	Traj.	56.61	63.33	25.07	0.294	0.572	81.43	74.63	65.83	44.20
+ RLOO	Token	Traj.	56.31	61.85	23.91	0.297	0.563	75.71	70.90	62.49	42.60
+ GSPO	Token	Traj.	57.08	56.14	22.59	0.289	0.552	79.29	77.61	69.78	48.80
+ GiGPO	Step	Traj.	58.14	61.27	23.50	0.306	0.567	88.57	79.10	67.13	50.00
+ StepPO	**Step**	**Step**	**63.78**	**66.16**	**29.87**	**0.327**	**0.585**	**92.14**	**85.82**	**77.52**	**57.80**

5.2 Main Results

Table 2 reports the main results, with all numbers averaged over three random seeds. StepPO achieves the best performance on every metric for both backbone models, showing consistent gains across tasks. On multi-hop QA, StepPO outperforms all baselines on in-domain HotpotQA and remains strongest on out-of-domain 2Wiki and MuSiQue, indicating that step-centric optimization transfers beyond training. This advantage extends to non-QA agentic environments: StepPO achieves the best paper-search results on RealResearchQuery, the highest seen and unseen win rates on ALFWorld, and the best task score and purchase success rate on WebShop. Regarding the low scores of Qwen3-1.7B under ReAct prompting, we observe that these scores mainly stem from the model's failure to follow our tool-calling format, while RL training substantially improves its performance. These results suggest that aligning MDP formulation and credit assignment with interaction steps is important for LLM agent training.

5.3 Ablation Study

We study the contribution of two key components in StepPO: step-level GAE for credit assignment and step-level importance sampling (IS) for policy optimization. As shown in Figure 3, removing either component consistently degrades performance across all tasks, indicating that both credit estimation and policy-ratio computation should follow

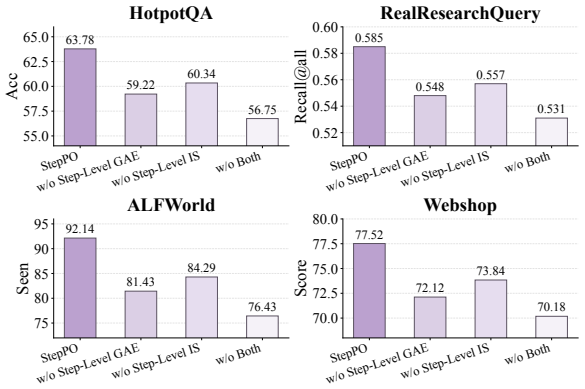

Figure 3: Ablation results across four benchmarks. Removing step-level GAE or step-level importance sampling consistently hurts performance.

interaction-step boundaries. The variant without step-level GAE suffers more on long-horizon tasks, suggesting that poorly aligned credit assignment weakens intermediate decision modeling. The variant without step-level IS also underperforms the full model, showing that step-level ratio aggregation stabilizes updates for multi-token actions. Removing both components yields the weakest results, confirming the benefit of joint step-level alignment.

5.4 In-Depth Analysis

Training Dynamics Analysis. Figure 4 further compares the optimization behavior of different RL methods on HotpotQA and ALFWorld. On both tasks, StepPO achieves the highest reward

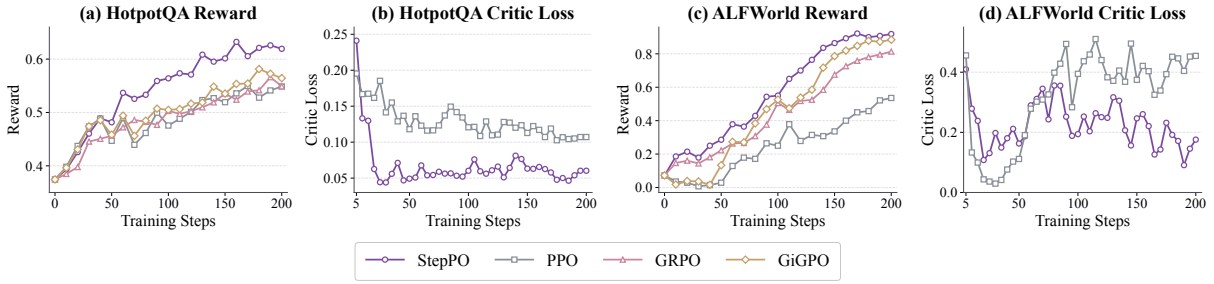

Figure 4: Training dynamics on HotpotQA and ALFWorld. We compare StepPO with PPO, GRPO, and GiGPO in terms of reward and with PPO in terms of critic loss across training steps. StepPO achieves higher rewards on both tasks while maintaining a lower critic loss, indicating more efficient learning and more accurate value estimation.

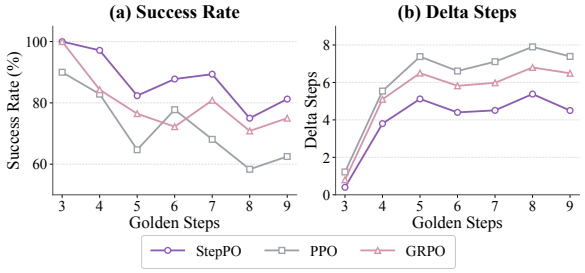

Figure 5: ALFWorld step-wise difficulty analysis grouped by human-annotated golden steps. Delta steps measure the gap between actual and golden step counts.

throughout most of training and continues to improve as training proceeds, while PPO, GRPO, and GiGPO either converge to lower rewards or improve more slowly. The critic-loss curves provide a complementary view of stability: on HotpotQA, StepPO keeps the critic loss consistently low after the initial updates, whereas PPO remains noticeably higher; on ALFWorld, PPO's critic loss fluctuates substantially as the interaction horizon grows, while StepPO maintains a more controlled loss despite achieving stronger rewards. These results indicate that modeling trajectories at the interaction-step level not only improves policy quality but also leads to accurate value estimation during training.

Step-Wise Difficulty Analysis. We analyze ALFWorld validation tasks by their human-annotated golden steps, where golden steps denote the minimum human-verified action sequence required to complete each task. Figure 5 evaluates two aspects for each step-count group: the success rate and the delta steps between the actual number of agent steps and the golden step count. As the required number of golden steps increases, tasks become harder and all methods show some degradation. However, StepPO exhibits a smaller drop in success rate than PPO and GRPO, especially in longer-horizon groups. It also maintains a consistently smaller step gap, indicating that its trajectories remain closer to the human-annotated golden plans. These trends suggest that StepPO learns trajectories that are not only more successful but also closer to human-verified solution paths, especially as tasks require more interaction steps.

Table 3: Tool-use behavior on the academic paper search task. We report the average response tokens of complete trajectory, search calls, expansion calls, and F1@all.

Method	Res. Tokens	Search	Expand	F1@all
PPO	2719.78	1.98	16.76	0.284
GRPO	2951.57	2.74	15.48	0.294
GiGPO	2814.79	**4.57**	16.54	0.306
StepPO	**2646.35**	3.21	**19.84**	**0.314**

Tool-Use Behavior Analysis. Table 3 analyzes how different RL methods use the retrieval tools in the academic paper search. StepPO achieves the best F1@all while using the fewest response tokens, indicating that its gains come from more effective interaction decisions rather than simply longer reasoning. Compared with PPO, GRPO, and GiGPO, StepPO performs substantially more citation and reference expansion actions, suggesting that step-level optimization encourages the agent to exploit the citation graph more effectively for paper exploration. Although GiGPO issues more search calls, StepPO obtains higher retrieval coverage by learning a more balanced search-and-expand strategy. This pattern shows that step-level credit assignment better aligns policy optimization with the behavior required by academic paper search.

Credit Assignment Analysis. Table 4 studies how GAE granularity affects WebShop under different λ values. When λ decreases from 0.99 to 0.95, token-level GAE shows a much larger score drop than step-level GAE (13.09% vs. 5.53%). In

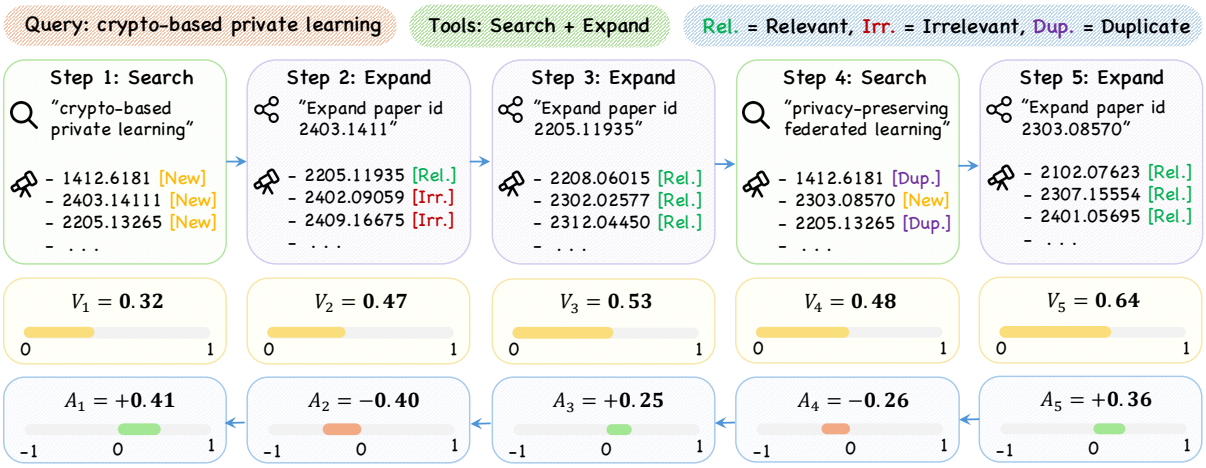

Figure 6: Case study of StepPO on the academic paper search task. Step-level values and advantages assign positive credit to productive decisions while down-weighting less useful steps, enabling efficient paper search exploration.

Table 4: Effect of GAE granularity under $\lambda = 0.99$ and 0.95 on WebShop. Step-level GAE shows a smaller score drop when λ decreases.

Params.	Res. Tokens	Avg. Steps	Score
Token-level GAE			
$\lambda = 0.99$	249.18	4.48	72.12
$\lambda = 0.95$	276.86	4.23	62.68
Rel. change	+11.11%	-5.58%	**-13.09%**
Step-level GAE			
$\lambda = 0.99$	142.96	4.91	77.52
$\lambda = 0.95$	184.23	4.15	73.23
Rel. change	+28.87%	-15.48%	**-5.53%**

GAE, a delayed reward propagated backward by n positions is weighted by roughly λ^n. Token-level GAE propagates rewards across generated tokens, yielding a decay scale of about $\lambda^{L\times T}$, where L is the average response length and T is the number of interaction steps. Step-level GAE instead propagates rewards across steps, yielding about λ^T. Thus, reducing λ weakens token-level advantages more sharply. These results show that step-level credit assignment better preserves delayed reward signals in long-text, multi-step agent interactions.

Efficiency Analysis. StepPO introduces negligible additional training overhead over PPO. The rollout and environment interaction remain unchanged, since both methods execute the same agent trajectories. The actor update also follows the same PPO-style objective. For the critic, StepPO estimates values only at step boundaries rather than all generated tokens, making supervision more compact. Advantage computation is lighter because it operates over interaction steps, though it accounts for only a small fraction of training time. Overall, StepPO keeps a similar per-iteration time to PPO, while its improved optimization can reduce the total cost needed to reach the same performance. Full comparison results are provided in Appendix C.

5.5 Case Study

Figure 6 shows an academic paper search example from StepPO. The state value generally increases as the paper pool becomes richer and more relevant. StepPO also assigns different advantages to different steps: effective search and expansion steps receive positive advantages, while low-yield steps receive smaller or negative advantages. This case shows that StepPO assigns credit to specific interaction decisions, better matching multi-turn tool-augmented agent behavior. The full case is provided in Appendix D.

6 Conclusion

In this work, we propose StepPO, a step-centric paradigm for agentic RL that addresses the granularity mismatch between token-level optimization and step-level agent decisions. StepPO reformulates agent execution as a step-level MDP and aligns credit assignment and policy optimization with interaction steps. Experiments across multi-hop QA, academic paper search, and text-world action tasks show that StepPO consistently improves performance over representative RL baselines. Further analyses show that step-level optimization improves training stability, value estimation, and long-horizon decision making. We hope this step-centric perspective provides a useful path for training more capable LLM agents.

Limitations

Our work has several limitations. First, although StepPO does not introduce additional computational overhead compared with PPO, it remains less efficient than critic-free algorithms such as GRPO under the same batch setting due to value estimation. Extending step-level optimization to critic-free RL algorithms is an important direction for future work. Second, the current study mainly focuses on single ReAct-style agent settings and does not explore more complex multi-agent or hierarchical systems. Whether step-level optimization remains effective in these settings requires further investigation. Third, we have not validated StepPO on very large backbone models, leaving its scalability to stronger LLM agents for future study. Finally, we only validate StepPO on interactive agent tasks, including multi-hop QA, academic paper search, and embodied text environments. Its effectiveness for broader post-training scenarios, such as alignment and reasoning tasks, remains to be explored.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267.

Anthropic. 2025. Claude code: Build, debug, and ship from your terminal. https://claude.ai/product/claude-code.

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. 2019. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, and 1 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Mingyue Cheng, Jie Ouyang, Shuo Yu, Ruiran Yan, Yucong Luo, Zirui Liu, Daoyu Wang, Qi Liu, and Enhong Chen. 2025. Agent-r1: Training powerful llm agents with end-to-end reinforcement learning. *arXiv preprint arXiv:2511.14460*.

Mingyue Cheng, Daoyu Wang, Shuo Yu, Qingchuan Li, Jie Ouyang, Yucong Luo, Yiju Zhang, Qi Liu, and Enhong Chen. 2026. A comprehensive survey of the llm-based agent: The contextual cognition perspective.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, and 1 others. 2025. Pasa: An llm agent for comprehensive academic paper search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11663–11679.

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

Jian Hu. 2025. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv e-prints*, pages arXiv–2501.

Tianyi Hu, Qingxu Fu, Yanxi Chen, Zhaoyang Liu, and Bolin Ding. 2026. Seeupo: Sequence-level agentic-rl with convergence guarantees. *arXiv preprint arXiv:2602.06554*.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. 2025. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. 2025. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

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

openclaw. 2026. openclaw: Your own personal ai assistant. any os. any platform. the lobster way. https://github.com/openclaw/openclaw.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Tingyue Pan, Jie Ouyang, Mingyue Cheng, Qingchuan Li, Zirui Liu, Daoyu Wang, Mingfan Pan, Shuo Yu, and Qi Liu. 2026. Paperscout: An autonomous agent for academic paper search with process-aware sequence-level policy optimization. *arXiv preprint arXiv:2601.10029*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in neural information processing systems*, 36:68539–68551.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pages 1279–1297.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

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

Tianyi Wang, Yixia Li, Long Li, Yibiao Chen, Shaohan Huang, Yun Chen, Peng Li, Yang Liu, and Guanhua Chen. 2026. Sppo: Sequence-level ppo for long-horizon reasoning tasks. *arXiv preprint arXiv:2604.08865*.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, and 1 others. 2025. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, and 1 others. 2025. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and 1 others. 2025. Agentfly: Fine-tuning llm agents without fine-tuning llms. *arXiv preprint arXiv:2508.16153*.

A Detailed Experimental Settings

A.1 Datasets Preparation

Multi-hop QA. The multi-hop QA setting uses HotpotQA (Yang et al., 2018) in the distractor setting as the in-domain benchmark. We train on the official HotpotQA training split with 90,447 questions and evaluate in-domain performance on the development split with 7,405 questions, where each question is paired with 10 Wikipedia context paragraphs. To evaluate cross-dataset generalization, we further test on the development splits of 2Wiki (Ho et al., 2020) and MuSiQue (Trivedi et al., 2022), containing 12,576 and 2,417 questions respectively; these examples are used only for evaluation and do not participate in parameter updates on HotpotQA. The reward is the normalized exact-match score between the predicted answer and the gold answer, and we report answer accuracy (Acc).

Academic Paper Search. Academic paper search follows the multi-turn paper discovery setting studied in PaperScout (Pan et al., 2026). The query data are built from RealResearchQuery (He et al., 2025), with 33,551 research queries for training and 50 queries for testing. Retrieved papers are evaluated against query-level relevance annotations. Reward and inference scoring use pasa-7b-selector[2], a Qwen2.5-7B based model optimized for relevance assessment, which is not exposed as an agent tool. For each newly discovered paper, the selector takes the research query together with the paper title and abstract, and returns a relevance score. During training, papers with near-zero selector scores are discarded, and each search or expansion step receives the sum of the top three newly discovered relevance scores, minus optional action cost; repeated search queries and invalid or repeated expansions receive a penalty of -0.5. During inference, the same selector scores the final paper pool, and we report post-threshold F1@all and Recall@all against the annotated arXiv IDs.

ALFWorld. ALFWorld (Shridhar et al., 2020) is a text-based embodied household task benchmark. We filter the official household task data to solvable tasks from supported task types and use 3,553 training tasks, 140 valid-seen tasks, and 134 valid-unseen tasks, covering six task families: examining objects under light, pick-and-place, clean-and-place, cool-and-place, heat-and-place, and pick-two-and-place. Each episode provides a goal instruction and is evaluated by whether the agent completes the specified household task. The terminal reward follows the environment's task-success signal, and we report win rate on the seen and unseen validation splits.

WebShop. WebShop (Yao et al., 2022a) is a text-based e-commerce navigation benchmark in which each goal specifies a shopping instruction. We use the full product setting, which contains approximately 1.18 million products and 12,087 predefined shopping goals. The split uses 11,587 goals for training and 500 goals for development. The reward is the WebShop task score computed from whether the purchased item satisfies the instruction and matches required product attributes, with purchase completion as an additional success signal. We report both average task score (Score) and purchase success rate (Succ.).

A.2 RL Environment Construction

Multi-hop QA. The multi-hop QA environment provides a dense retrieval interface over benchmark-specific Wikipedia paragraph indexes. For HotpotQA, the retrieval corpus is built by deduplicating all distractor context paragraphs from the training and development splits, yielding 509,308 passages. For cross-dataset evaluation, 2Wiki and MuSiQue use separate indexes built from the official full Wikipedia paragraph corpus with 5,902,082 passages and the deduplicated union of MuSiQue question-linked passages with 139,416 passages, respectively. We encode passages with `BAAI/bge-large-en-v1.5` and index them with FAISS. At each step, the agent observes the question, retrieved passages, previous search queries, and format feedback. It can issue up to 4 parallel `search(query)` calls per step, and the maximum horizon is 5 steps. When the agent has enough evidence, it terminates by outputting a short answer inside `<answer>...</answer>` tags.

Academic Paper Search. The academic paper search environment maintains a paper pool for each research query and exposes two tools: `search(query)`, which retrieves papers from an academic search service, and `expand(paper_id)`, which expands the citations and references of a paper already in the pool. The paper corpus is constructed from a January 2026 Semantic Scholar

[2] https://huggingface.co/bytedance-research/pasa-7b-selector

snapshot[3] by retaining arXiv papers with abstracts, resulting in approximately 3 million papers and 30 million in-corpus citation edges. The retrieval API supports sparse retrieval with a SQLite FTS5 full-text index using BM25 ranking, dense retrieval with Qdrant and BGE-M3 embeddings, and hybrid retrieval through reciprocal-rank fusion. The default search returns 10 papers; citation expansion returns up to 30 citing papers, and reference expansion returns up to 99 referenced papers. At each step, the agent observes the user query, the current paper pool, and previous search or expansion actions. It can make up to 5 parallel tool calls per step, and the maximum horizon is 5 steps.

ALFWorld. The ALFWorld environment is built on TextWorld. Each episode corresponds to a packaged TextWorld game file, and the runtime wrapper loads one game as an independent interaction instance. At each step, the wrapper returns the current textual observation and the admissible-command list, and the agent must submit exactly one command that matches the list through the environment-step tool. These commands include navigation and household operations such as moving, opening or closing objects, taking objects, and placing objects. Task success is determined by the `won` signal returned by the environment. For each agent step, the prompt is reconstructed from the current observation and history actions rather than accumulated as a full multi-turn dialogue, matching the step-level MDP formulation. We set the maximum horizon to 20 interaction steps.

WebShop. The WebShop environment is implemented as a self-hosted HTTP shopping simulator. It builds a SQLite product store and a Lucene full-text search index over the full product catalog. During training, the client interacts with the server through reset and step calls: reset initializes a shopping goal, and each step applies one executable action to the current page state. Available actions are generated dynamically from the current page, including keyword search, product clicks, option selection, back navigation, and purchase. The agent issues one action through the environment-step tool and receives the next page observation. Each agent step reconstructs the prompt from the current observation and recent action history rather than carrying a full dialogue transcript. We set the maximum horizon to 15 interaction steps.

[3] https://api.semanticscholar.org/api-docs/datasets

A.3 Baselines

ReAct-style Prompting. The prompting baseline evaluates the pretrained backbone without RL fine-tuning. The model uses the same task prompts and tool schemas as the RL methods, and follows a ReAct-style format (Yao et al., 2022b) that interleaves reasoning traces and environment-facing actions. This baseline measures the capability of the pretrained policy under the target interaction protocol before policy optimization.

PPO. PPO (Schulman et al., 2017) is implemented as the token-level GAE baseline. It uses a learned critic and estimates advantages at token granularity. This baseline keeps the conventional LLM RL view that generated tokens are the optimization unit, even though the environment state changes only after a complete interaction response.

Reinforce++. Reinforce++ (Hu, 2025) is a critic-free token-level return baseline. It computes discounted returns over valid generated tokens and applies masked whitening. We also include the Reinforce++ baseline variant, which subtracts the same-prompt rollout-group average return before broadcasting the resulting trajectory advantage to valid tokens.

GRPO. GRPO (Shao et al., 2024) is a critic-free group-relative baseline. For each prompt, it samples multiple rollouts and computes a trajectory-level relative advantage from the group rewards. We use 8 rollouts per prompt by default and reduce the effective prompt batch size accordingly to keep the update budget comparable.

RLOO. RLOO (Ahmadian et al., 2024) is another trajectory-level group baseline. It computes each rollout's baseline from the average reward of the other rollouts in the same prompt group, leaving out the current rollout's own reward. The resulting advantage is still assigned at trajectory granularity.

GSPO. GSPO (Zheng et al., 2025) is used as a sequence-level policy-loss baseline. In our implementation, it keeps the GRPO trajectory-level advantage estimator and replaces the token-level policy ratio with a sequence-level ratio objective. This isolates the effect of the sequence-level policy loss from StepPO's step-level credit assignment.

GiGPO. GiGPO (Feng et al., 2026) is a critic-free baseline that combines trajectory-level group comparison with grouped step-level information.

Table 5: Key optimization hyperparameters used in the main experiments. The critic learning rate applies to critic-based methods, including StepPO and PPO.

Hyperparameter	HotpotQA	RealResearchQuery	ALFWorld	WebShop
Actor learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}	1×10^{-6}
Critic learning rate	1×10^{-5}	1×10^{-5}	1×10^{-5}	1×10^{-5}
Max prompt length	10240	10240	8192	16384
Max response length	1024	4096	4096	4096
StepPO train batch	128	128	128	128
GRPO rollout number	8	8	8	8
Actor micro batch	4	4	4	4
Max environment steps	5	5	20	15

It introduces a step-advantage component grouped by anchor observations, but does not learn a value function. We include it as a strong comparison for finer-grained agent credit assignment.

A.4 Training Settings

Our implementation follows the Agent-R1 agent-training framework (Cheng et al., 2025) and uses veRL as the backend RL framework (Sheng et al., 2025), with vLLM generation (Kwon et al., 2023). Experiments are run on a server with 8 NVIDIA H100 GPUs. Unless otherwise specified, we set the actor learning rate to 1×10^{-6}, the critic learning rate to 1×10^{-5} for critic-based methods, the training batch size to 128, and the actor micro-batch size to 4 per GPU. Group-based baselines use 8 rollouts per prompt and a batch size of 16, yielding an effective batch size of 128 for fair comparison. We use the same discount factor $\gamma = 0.99$ and GAE trace parameter $\lambda = 1.0$ across tasks, and set the actor-side KL regularization coefficient to 0.001. Within each benchmark, all RL methods share the same backbone model, training data, task description, tool interface, hyperparameter budget, and evaluation protocol; they differ only in their RL algorithm design. StepPO and PPO enable the critic, while GRPO, RLOO, Reinforce++, GSPO, and GiGPO are critic-free baselines according to their estimator definitions.

B Prompt Templates

This appendix lists the prompt templates used to instantiate the tool-calling interface described in Section 5. The `prompt` field stored in each dataset row contains the task input, such as the HotpotQA question, paper-search query, ALFWorld goal, or WebShop shopping instruction. During rollout, the agent flow reconstructs a step prompt from the current observation or retrieved evidence, action history, available tools or admissible actions, and expected output format. This construction matches the step-level MDP formulation: each prompt represents the current interaction state, and each model response is parsed as one environment-facing action. We preserve placeholders such as `{observation}` and `{history_actions}` because they are populated at rollout time.

Table 6: Training time per iteration. Units are seconds per iteration (s/iteration).

Method	Rollout Time	Actor Update	Critic Update
PPO	14.12	76.65	104.60
StepPO	14.01	74.35	93.18

C Training Efficiency

Table 6 reports the measured training time of PPO and StepPO under the same rollout and update setting. StepPO keeps the rollout process and actor update unchanged relative to PPO, and its critic update remains comparable because values are estimated only at interaction-step boundaries. These results show that StepPO does not introduce additional training computation in practice.

D Representative Trajectory

To complement the case study in Section 5, we include a shortened academic paper search trajectory. The example illustrates how the agent grows a paper pool through search and citation/reference expansion, while selecting actions at the same interaction-step granularity used by StepPO. It is lightly edited from a real rollout: long observations, full paper lists, and repetitive parallel tool calls are abbreviated with ellipses, while the step-level decision pattern is preserved.

HotpotQA Prompt

You are a research agent. Your goal is to answer the User Query using Wikipedia search evidence.

User Query

{user_query}

History Actions

{history_actions}

Retrieved Passages

{passage_list}

Recent tool / format issues

{tool_feedback}

Instructions

- Analyze the Retrieved Passages and History Actions to determine the next set of actions. Enclose your analysis of the state and decision logic within `<analysis>...</analysis>` tags.
- You support parallel tool calling. You should output multiple tool calls in a single step if several independent actions are valuable at the current state.
- Attend to the history actions and avoid repeating the same search queries.

When you can answer the question from the current passages, put the short final answer inside `<answer></answer>` tags instead of further tool calls.

Tool Definition

```
{
  "type": "function",
  "function": {
    "name": "search",
   "description": "Search Wikipedia for passages relevant to the user question. Use
      natural-language or keyword queries; must differ from prior history queries
       when possible.",
    "parameters": {
      "type": "object",
      "properties": {
        "query": {
          "type": "string",
       "description": "A single search query (natural language or keywords). Must
            differ from all history queries when seeking new evidence."
        }
      },
      "required": ["query"]
    }
  }
}
```

Paper Search Prompt

You are a research agent. Your goal is to find papers relevant to the User Query.

User Query

{user_query}

History Actions

{history_actions}

Paper List

{paper_list}

Instructions

- Analyze the Paper List and History Actions to determine the next set of actions. Enclose your analysis of the state and decision logic within `<analysis>...</analysis>` tags.
- You support parallel tool calling. You should output multiple tool calls in a single step if several independent actions are valuable at the current state.
- Attend to the history actions and avoid repeating the same search query or expanding the same paper.

Tool Definitions

```
[
  {
    "type": "function",
    "function": {
      "name": "search",
     "description": "Search for relevant papers with the hybrid retrieval API.",
      "parameters": {
        "type": "object",
        "properties": {
          "query": {
            "type": "string",
          "description": "A single search query in natural language or keywords.
              Must differ from all history queries."
          }
        },
        "required": ["query"]
      }
    }
  },
  {
    "type": "function",
    "function": {
      "name": "expand",
     "description": "Expand from an existing paper by merging its citations and
        references to surface more related works.",
      "parameters": {
        "type": "object",
        "properties": {
          "paper_id": {
```

```
            "type": "string",
            "description": "The paper identifier of a paper already present in the
                current paper list."
            }
          },
          "required": ["paper_id"]
        }
      }
    }
]
```

Paper Search Selector Prompt

```
You are an elite researcher in the field of AI, conducting research on {user_query}.
Evaluate whether the following paper fully satisfies the detailed requirements of
the user query and provide your reasoning. Ensure that your decision and reasoning
are consistent.

Searched Paper:
Title: {title}
Abstract: {abstract}

User Query: {user_query}

Output format: Decision: True/False
Reason:...
Decision:
```

ALFWorld Prompt

You are acting in ALFWorld TextWorld. Choose exactly one command from the provided admissible commands each turn. You may reason briefly inside <think></think> before acting. Call the env_step tool with a exact command. Do not output a final answer.

Current Observation

{observation}

History Actions

{history_actions}

Admissible Commands

{admissible_commands}

Instructions

```
- Think briefly about the current state inside `<think>...</think>` tags.
- Use exactly one command through the `env_step` tool.
- The command must exactly match one item from `Admissible Commands`.
- Follow ALFWorld TextWorld command style such as `go to dresser 1`,
  `take mug 1 from cabinet 3`, `use desklamp 1`.
- Use the official observation text as the source of truth.
- Do not output explanations or a final natural-language answer.
```

Output Format

```
<think>
[Your brief reasoning about the current state and next command.]
</think>
<tool_call>
{{"name": "env_step", "arguments": {{"command": "[one admissible command]"}}}}
</tool_call>
```

Tool Definition

```
{
  "type": "function",
  "function": {
    "name": "env_step",
    "description": "Execute one ALFWorld TextWorld command and return the next
       official observation.",
    "parameters": {
      "type": "object",
      "properties": {
        "command": {
          "type": "string",
        "description": "A single ALFWorld TextWorld command such as `go to dresser
           1`, `open cabinet 3`, `take mug 1 from cabinet 3`, `use desklamp 1`. It
            must exactly match one currently admissible command."
        }
      },
      "required": ["command"]
    }
  }
}
```

WebShop Prompt

You are acting in the WebShop text environment. Your goal is to find and buy the product that best satisfies the shopping instruction. Use exactly one executable WebShop action each turn through the env_step tool. Do not explain.

Shopping Instruction

{instruction}

Current Observation

{observation}

Recent History

{recent_history}

Available Actions

{available_actions}

Instructions

- Use exactly one action through the `env_step` tool.
- The `command` must be one available action exactly, except replace `<your query>` in `search[<your query>]` with concise product keywords.
- Click product ASINs, option values, `Description`, `Features`, `Reviews`, `Back to Search`, `Back to Item`, or `Buy Now` only when listed above.
- Buy only when the selected product and options satisfy the shopping instruction.

Tool Definition

```
{
  "type": "function",
  "function": {
    "name": "env_step",
   "description": "Execute one WebShop action and return the next  observation.",
    "parameters": {
      "type": "object",
      "properties": {
        "command": {
          "type": "string",
        "description": "A single action such as `search[wireless headphones]` or
          `click[Buy Now]`."
        }
      },
      "required": ["command"]
    }
  }
}
```

Paper Search Trajectory

```
User Query:
crypto-based private learning

==================== Step 1 ====================
<analysis>
The paper pool is empty. The query concerns privacy-preserving learning with
cryptographic techniques, so the first step should build a broad seed pool. Use
parallel searches that cover homomorphic encryption, federated learning,
secure computation, and differential privacy. No previous queries or expanded
papers need to be avoided yet.
</analysis>
<tool_call>
{"name": "search", "arguments": {"query": "crypto-based private learning"}}
</tool_call>
<tool_call>
{"name": "search", "arguments": {"query": "homomorphic encryption ML"}}
</tool_call>
...

==================== Step 2 ====================
Observation summary:
The pool now contains seed papers on encrypted inference, private training,
federated learning, and privacy-preserving neural networks.

<analysis>
Several retrieved papers directly match the query, especially works involving
homomorphic encryption and encrypted neural-network inference. To turn the seed
pool into a connected research graph, expand the most relevant new papers. Avoid
repeating the four search queries from Step 1.
</analysis>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "1412.6181"}}
</tool_call>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "2403.14111"}}
</tool_call>
...

==================== Step 3 ====================
Observation summary:
Expansion surfaces papers on FHE-based inference, encrypted training, transfer
learning, transformers, and activation-function approximations.

<analysis>
The pool is now richer but still concentrated around encrypted inference. Expand
new papers that diversify the graph toward training, transformer models, secure
pipelines, and systems. Skip papers already expanded in previous steps.
```

```
</analysis>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "2205.11935"}}
</tool_call>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "2402.09059"}}
</tool_call>
...

==================== Step 4 ====================
Observation summary:
New papers include CryptoDL, THE-X, MedBlindTuner, CURE, Pencil, SPEED, SoK
surveys, and secure split-learning or hybrid-encryption methods.

<analysis>
The graph now covers both foundational and recent work. The next useful action
is not another broad search, but targeted citation/reference expansion from
new high-signal papers. Prioritize papers that connect FHE with training,
fine-tuning, transformers, secret sharing, or split learning. Avoid expanding
papers already marked as explored.
</analysis>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "1711.05189"}}
</tool_call>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "2107.14338"}}
</tool_call>
...

==================== Step 5 ====================
Observation summary:
The pool contains a broad set of relevant candidates: FHE-based neural-network
inference, private training, graph neural networks, RNN training, hybrid
HE/secret-sharing systems, functional encryption for federated learning, and
survey papers.

<analysis>
At this point, most broad directions have been covered. Expand only the remaining
new papers that add missing subtopics, such as graph learning, recurrent models,
hybrid cryptographic protocols, or comparative surveys. This final step improves
recall while avoiding repeated searches and already-expanded papers.
</analysis>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "2012.09364"}}
</tool_call>
<tool_call>
{"name": "expand", "arguments": {"paper_id": "2209.11904"}}
</tool_call>
...
```