Productive State-Space Exploration for Multi-turn RL in LLM Agents

Anonymous ACL submission

Abstract

Reinforcement Learning (RL) for Large Language Model (LLM) agents in multi-turn, long-horizon environments remains challenging due to inefficient action selection and repetitive behavioral patterns. In particular, agents often become trapped in high-confidence yet unproductive behaviours, repeatedly executing invalid actions or entering action loops that hinder task completion. We observe that these frequently violate environment preconditions, revealing a mismatch between the agent's representation of the internal state and the dynamics of the underlying environment. In addition, repetitive high-confidence actions lead to cyclical behaviors that provide little to no progress towards the goal. We argue that addressing these challenges requires more *productive task-aligned exploration* to improve the agent's understanding of environment dynamics. Motivated by intrinsic curiosity-driven exploration, we propose a *conditional state-space exploration* framework that balances task completion with intrinsic information-seeking behavior. Specifically, we introduce an intrinsic reward that penalizes actions that do not increase information in the agent's state memory, thus encouraging exploration that maximizes coverage of the state-space, reducing redundant behaviors. Strong experimental results demonstrate that our approach substantially improves policy learning in multi-turn environments, mitigating invalid actions and repetitive loops.

1 Introduction

Reinforcement learning (RL) has become an important training paradigm for large language model (LLM) agents that must operate over multiple turns in partially observable, long-horizon environments (Carta et al., 2023; Shani et al., 2024; Guan et al., 2026). In contrast to single-step prediction tasks, these agents must select actions sequentially, update internal memory, and adapt to changing states while receiving sparse and delayed feedback. Recent work has shown that RL is increasingly relevant for improving reasoning, agentic control, and long-horizon decision making in LLMs, but it has also revealed a persistent difficulty: agents often become stuck in repetitive, high-confidence behaviors that are locally plausible yet unproductive for the task (Liu et al., 2025; Zhang et al., 2025; Xi et al., 2025).

In such settings, effective learning depends not only on exploiting promising actions but also on exploring the environment in a way that improves the agent's understanding of its dynamics. In practice, agents often overcommit to high-confidence actions that are locally plausible but globally unhelpful, leading to invalid interactions or repetitive loops that consume budget without advancing the task. Such failures, as shown in Figure 1, typically reflect a mismatch between the agent's internal state representation and the true environment dynamics, especially when the agent lacks sufficient feedback about which actions are feasible or informative. This motivates the need for a form of *productive state-space exploration* that helps the agent learn environment dynamics while remaining aligned with task completion.

A natural way to improve exploration is to encourage the agent to visit a broader range of states. In principle, increasing the entropy of the explored state space can improve coverage, reduce mode collapse, and expose the agent to more informative trajectories. However, directly maximizing exploration entropy can also be problematic. If encouraged too strongly, the agent may drift toward diverse but task-irrelevant behavior, thereby sacrificing progress on the underlying objective. In long-horizon environments, this tension between exploration and exploitation is especially important: *exploration should broaden the agent's experience, but it should not pull the policy away from useful action sequences that lead to task completion.*

This paper focuses on productive exploration

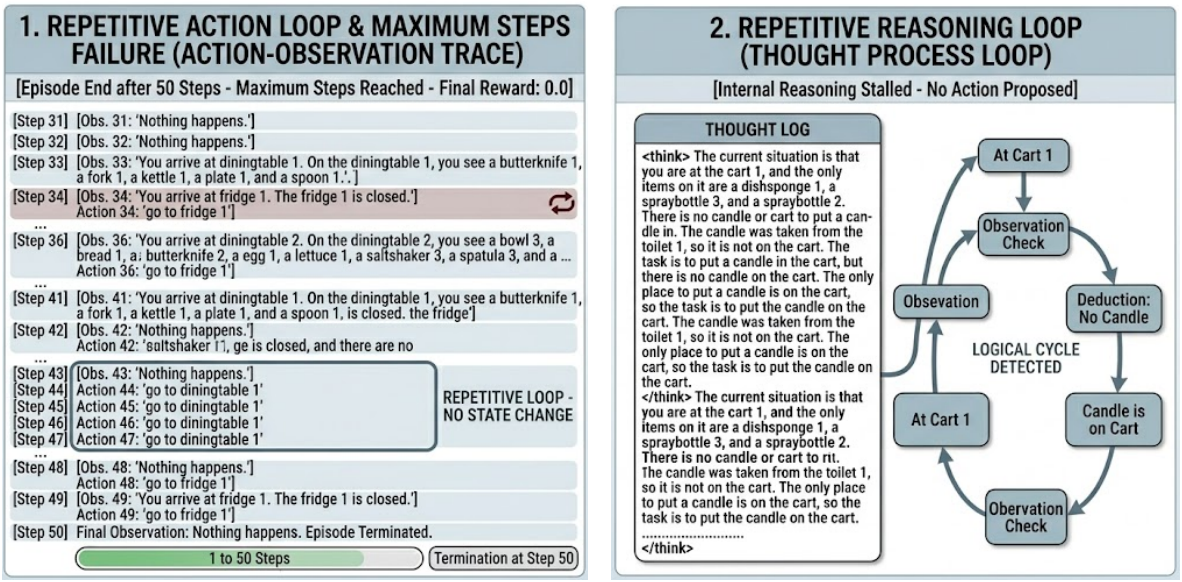

Figure 1: Recurring error patterns observed in LLM agents behaviour after RL training. In the first case, agent takes repetitive actions which consumes the maximum step budget. In the second pattern, the agent is unable to get out of a circular reasoning pattern leading to episode failures.

through *intrinsic curiosity-driven rewards*, but with a conservative formulation that preserves task alignment. Rather than positively rewarding state-space entropy increase in an unconstrained manner, we penalize actions that fail to contribute meaningfully to state-space diversification or information gain. The intuition is that unhelpful actions, such as invalid steps, redundant repetitions, or transitions that do not improve the agent's state knowledge, should reduce reward because they occupy probability mass without advancing exploration in a useful way. In this sense, negative reward acts as a shaping mechanism: it discourages locally repetitive behavior and encourages the policy to seek alternative trajectories that better expand the explored state space.

This formulation reframes entropy increase as a consequence of avoiding unproductive actions rather than as an objective to be optimized directly. By penalizing actions that do not improve the state representation, the agent is pushed away from narrow loops and toward more diverse trajectories, while still being constrained by the task objective. As a result, the exploration policy remains productive: it increases coverage of the state space, but only insofar as that coverage is informative and compatible with downstream task success. This is particularly valuable in multi-turn LLM agents, where repeated invalid actions or low-information loops can rapidly degrade performance.

Our approach is consistent with the broader literature on intrinsic motivation in reinforcement learning, where curiosity, novelty seeking, and uncertainty reduction have been used to improve learning under sparse rewards (Chentanez et al., 2004; Aubret et al., 2019; Gao et al., 2025). It also aligns with recent efforts to train long-horizon LLM agents, where robustness depends not only on reasoning ability but also on the capacity to avoid repetitive or uninformative interactions with the environment (Kalyan and Andrews, 2025). Within this setting, penalizing unhelpful actions provides a principled compromise between unrestricted entropy maximization and overly conservative exploitation. The resulting behavior promotes broader state coverage without losing contact with the task.

In summary, we propose a *conditional intrinsic reward* framework that supports *productive exploration* by penalizing actions that do not improve *state-space entropy* or *information gain*. Our key argument is that entropy expansion is useful, but only when it is induced indirectly through the suppression of redundant or uninformative behavior. This allows the agent to explore more broadly while maintaining a strong coupling to task progress. We evaluate our proposed exploration strategy on two prominent multi-turn benchmark environments: `ALFWorld` and `WebShop`. Empirical results show that our approach surpasses recent baselines in performance metrics of success rates and significantly mitigates both invalid and repetitive action rates.

2 Related Work

2.1 Prompting-Based LLM Agents

Early LLM agent frameworks rely on prompting for multi-step reasoning and action selection without updating model weights. ReAct (Yao et al., 2023) interleaves reasoning (Thought) and actions (Act), outperforming pure reasoning or action prompting on tasks such as ALFWorld and WebShop. Reflexion (Shinn et al., 2023) extends this with verbal reinforcement, storing self-reflections from failed episodes in episodic memory to guide future decisions. Although ReAct and Reflexion improves performance on multi-turn tasks, both methods depend heavily on self-evaluation quality, which degrades for smaller LLMs, and cannot structurally suppress invalid actions since prompting alone cannot alter action priors.

2.2 Reinforcement Learning for LLM Agents

Recently, reinforcement learning (RL) has emerged as a principled approach for post-training LLMs, with growing focus on agent-based LLMs in multi-turn environments. PPO (Schulman et al., 2017) and its critic-free variant GRPO (Shao et al., 2024), which estimates baselines from group rollout scores, are widely adopted for RL-based LLM training. RLOO (Kool et al., 2019; Ahmadian et al., 2024) further reduces variance without a learned critic, and AGILE (Feng et al., 2024) applies PPO to LLM agents using a four-module architecture with proactive expert consultation, significantly outperforming prompting on long-horizon QA tasks. Retrospex (Xiang et al., 2024) combines an offline RL critic with LLM action likelihood through dynamic rescoring, improving performance on ALFWorld and WebShop over prompting baselines.

GIGPO (Feng et al., 2026) captures global trajectory quality and group level effectiveness without additional rollouts, providing rich signals for LLM agent learning. Recently, ProxMO (Fang et al., 2026) presents a framework to address GRPO's context-agnostic normalization and GiGPO's discrete state-grouping limitation. They explore the provision of gradient intensities based on task difficulty, and replace hard boundary partitioning with continuous semantic similarity weighting, so as to provide richer reward signals that are less task-agnostic. In totality, these works establish multiple avenues to improve vanilla RL for multi-turn LLM agents, such as stable baselines and supporting tools to provide more information of the world. However, they do not directly address the exploration failure modes we identify in this work. Specifically, we explore the notion of invalid actions that arise from environment precondition mismatches, resulting in action loops driven by high-confidence but unproductive behaviors. Our approach adds a conditional intrinsic reward that explicitly penalizes such behaviors during RL training.

Recent RL methods for multi-turn LLM agents focus on improving credit assignment, exploration, and training stability. ArCHer (Zhou and Zanette, 2024) introduces a hierarchical RL framework combining off-policy temporal difference learning at the utterance level with on-policy policy gradients at the token level, substantially improving sample efficiency over PPO while emphasizing information-seeking behavior in multi-turn settings. Tree-GRPO (Ding and Ye, 2026) improves training efficiency by replacing chain-based rollouts with tree-structured sampling, enabling step-level supervision from outcome rewards alone. Addressing exploration instability, EPO (Xu et al., 2025) identifies *exploration-exploitation cascade failure* in shared multi-turn policies and mitigates it via trajectory-level entropy regularization and entropy smoothing, yielding gains on ScienceWorld and ALFWorld. Similarly, RAGEN (Wang et al., 2025) studies multi-turn RL failure modes, identifying an "Echo Trap" caused by reward variance cliffs and gradient spikes, and proposes trajectory filtering with a critic for stabilization. While these works improve *how* agents learn—through efficient credit propagation, entropy stabilization, or robust optimization—our approach is complementary, focusing on *what* to explore by penalizing invalid or repetitive actions that fail to expand state-space knowledge and rewarding actions with positive entropy contribution.

2.3 Intrinsic Motivation and Exploration in RL

Intrinsic motivation has long been used to improve exploration under sparse rewards by encouraging state diversity and novelty. Curiosity-based methods like Pathak et al. (2017); Burda et al. (2019) reward novel states or high prediction errors, while count-based exploration formalizes revisitation penalties through information gain via KL-divergence between prior and posterior state beliefs (Bellemare et al., 2016). Extending state-entropy exploration to task-reward settings, Kim

et al. (2023) propose value-conditional state entropy (VCSE), addressing the tendency of naive entropy maximization to overexplore low-value regions by maximizing entropy within value partitions. Mutti et al. (2021); Ashlag et al. (2026) further provide a theoretical foundation for state entropy regularization, showing that it improves robustness and promotes broader coverage across multiple viable trajectories, unlike policy entropy which mainly injects noise along a dominant path. These works motivate our approach – by targeting state-space entropy, our conditional intrinsic penalty discourages revisiting already-observed states. This induces the same broader coverage properties as characterized in (Ashlag et al., 2026). Additionally, by tying the penalty to information gain rather than raw visitation counts, we ensure that only transitions that fail to expand the agent's knowledge are penalized, preserving the pull of the task reward toward goal-relevant states.

In LLM-based RL, intrinsic signals have recently been explored to complement sparse task rewards. MIRA (Nourzad and Joe-Wong, 2026) improves exploration by using memory graphs from agent trajectories and semantic reasoning to shape PPO advantages, mitigating poor early critic calibration. Similarly, our approach leverages accumulated experience for exploration, but instead defines an information-gain-based intrinsic signal that penalizes observations failing to expand state knowledge. This is simpler to compute, grounded in Shannon entropy (Proposition 1), and aligned with repetitive and invalid behaviors in text-based environments. Further recent work on moral intrinsic rewards (Tennant et al., 2025), has shown that intrinsic signals can shape agent behavior toward desired norms without task-specific reward engineering, reinforcing the broader applicability of intrinsic motivation for LLM alignment. Our contribution fits within this emerging direction: intrinsic rewards as a principled mechanism for shaping LLM agent behavior in ways that extrinsic rewards alone cannot achieve.

3 Methodology

3.1 Problem Setup

We consider reinforcement learning for LLM agents operating in partially-observable and long-horizon interactive environments. At each timestep t, the agent is in state s_t, takes an action $a_t \in \mathcal{A}$, and receives an observation o_t from the environment. We define the agent's state s_t as the set of all observations received so far along the trajectory, i.e., the information the agent has accumulated about the environment up to time t. The objective is to learn a policy $\pi(a_t \mid s_t)$ that maximizes expected cumulative reward while completing the task within a finite horizon T.

Unlike standard short-horizon decision problems, this setting introduces two recurring failure modes. First, the agent may produce *invalid actions* whose preconditions are not satisfied, causing the transition to fail or yielding no meaningful progress. Second, the agent may enter *action loops*, repeatedly selecting a small set of actions that do not improve task state or move the agent closer to completion. Both issues reduce learning efficiency and often reflect an imperfect internal model of the environment dynamics. In practice, these failures are especially harmful in LLM agents because they waste interaction budget and reinforce unproductive behavior patterns.

3.2 State Representation

To support long-horizon decision making, we define the agent's state as the accumulated observation history:

$$s_t = \{o_1, o_2, \ldots, o_t\}.$$

This representation captures the information available to the agent at time t. Instead of relying only on the most recent observation, the agent maintains a memory of the full trajectory of *unique observations*, enabling it to reason about what has already been explored and what remains uncertain. This accumulated state serves as the basis for evaluating whether a new observation provides additional information about the environment.

At each step, the agent observes the current state s_t, selects an action a_t, and receives the next observation o_{t+1}. The state is then updated by appending the new observation:

$$s_{t+1} = s_t \cup \{o_{t+1}\}.$$

The intrinsic reward is computed from the change in information content between s_t and o_{t+1}. If the new observation does not improve the agent's knowledge of the environment, a negative intrinsic reward is assigned. This signal is combined with the task reward and used to update the policy.

3.3 Conditional State-Space Exploration

We propose a conditional state-space exploration mechanism that rewards informative transitions and penalizes unhelpful ones. After taking action a_t, the agent receives a new observation o_{t+1}, and the information gain is computed by comparing this new observation with the previous state s_t. Let $I(\cdot)$ denote a measure of information content or entropy associated with the agent's accumulated state representation. We define the change in information as

$$\Delta I(s_t, o_{t+1}) = I(s_t \cup \{o_{t+1}\}) - I(s_t).$$

This quantity measures whether the new observation expands the agent's knowledge of the environment.

The intrinsic reward is then defined as:

$$r_t^{\text{intrinsic}} = \begin{cases} -\gamma & \text{if } \Delta I(s_t, o_{t+1}) \leq 0, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $\gamma > 0$ controls the penalty magnitude. This formulation penalizes actions that do not lead to a meaningful increase in the agent's information about the environment, thereby discouraging redundant or cyclic behavior.

3.4 Final Reward

The final reward used for policy optimization combines the task reward and the intrinsic reward:

$$r_t = r_t^{\text{task}} + \lambda r_t^{\text{intrinsic}},$$

where $\lambda \geq 0$ balances task performance and exploration pressure. The resulting objective is

$$J(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{T-1} \rho^t r_t \right],$$

where $\rho \in [0, 1]$ is the discount factor. This objective encourages the agent to maximize task reward while avoiding actions that fail to improve state-space coverage.

3.5 Entropy-Oriented Exploration Objective

The motivation for this design is that directly maximizing state-space entropy can sometimes distract the agent from the task objective. An overly strong incentive for novelty may lead the policy to explore diverse but irrelevant trajectories. By contrast, penalizing non-informative actions creates a softer and more task-aligned mechanism for increasing state-space entropy. The policy is discouraged from remaining in low-diversity action cycles, but it is still constrained by the external task reward, which ensures that exploration remains useful for task completion.

This conditional structure is especially important in long-horizon environments, where the agent must balance exploration and exploitation over many sequential decisions. Actions that expand state coverage or contribute to task progress receive *no* intermediate positive incentive. However, actions that actively fail to improve the agent's state knowledge are *penalized*. As a result, the learned policy is pushed toward trajectories that are both diverse and productive.

This penalizing mechanism promotes maximization of the information density of the agent's history and connects directly to the established principles of information-theoretic exploration. Consequently, the underlying framework inherits the structural benefits of entropy-maximizing exploration. Below we describe the connection between state-space entropy maximization and our proposed productive exploration strategy.

Proposition 1. *Let the explored state space at time t be represented by the empirical visitation distribution*

$$P_t = (p_1^{(t)}, \ldots, p_n^{(t)}), \qquad p_i^{(t)} = \frac{c_i^{(t)}}{\sum_{j=1}^n c_j^{(t)}},$$

where $c_i^{(t)}$ denotes the visitation count of state i, $n = n_t$ is the number of distinct states observed so far, and P_t be the probability of the sequence of actions leading to time t. Thus, the Shannon entropy of the agent's actions can be defined as $H(P_t) = -\sum_{i=1}^n p_i^{(t)} \log p_i^{(t)}$. Then,

1. *If the new observation introduces a previously unseen state, then $H(P_{t+1}) > H(P_t)$.*
2. *If the new observation repeats an already-visited state k satisfying $p_k^{(t)} \geq e^{-H(P_t)}$, then $H(P_{t+1}) \leq H(P_t)$.*

Proof. Let

$$N = \sum_{i=1}^n c_i^{(t)}, \qquad \alpha = \frac{1}{N+1}.$$

The update induced by a new observation can be written as

$$P_{t+1} = (1 - \alpha) P_t + \alpha e,$$

where e is a point mass on the observed state.

Case 1: Introduction of a new state.
Suppose the observation corresponds to a previously unseen state. Then the support expands by one dimension and

$$P_{t+1} = \big((1-\alpha)P_t, \alpha\big).$$

The entropy of the updated distribution decomposes as

$$H(P_{t+1}) = (1-\alpha)H(P_t) + h(\alpha),$$

where $h(\alpha) = -(1-\alpha)\log(1-\alpha) - \alpha \log \alpha$ is the binary entropy term arising from introducing the new state. Therefore,

$$H(P_{t+1}) - H(P_t) = h(\alpha) - \alpha H(P_t).$$

Since the support size before the update is at most N,

$$H(P_t) \leq \log N.$$

Substituting $\alpha = 1/(N+1)$ gives

$$h(\alpha) - \alpha H(P_t) \geq \log(N+1) - \log N > 0.$$

Hence $H(P_{t+1}) > H(P_t)$.

Case 2: Repetition of an existing state.
Suppose the observation revisits an already-seen state $k \in \{1, \ldots, n\}$. Then

$$P_{t+1} = (1-\alpha)P_t + \alpha e_k,$$

where e_k is the point mass on state k. Define $f(\alpha)$:

$$f(\alpha) = H\big((1-\alpha)P_t + \alpha e_k\big).$$

Since Shannon entropy is concave, f is concave in α. Its directional derivative at $\alpha = 0$ is

$$f'(0) = -H(P_t) - \log p_k^{(t)}.$$

Thus,

$$f'(0) \leq 0 \quad \Longleftrightarrow \quad p_k^{(t)} \geq e^{-H(P_t)}.$$

By concavity, a non-positive initial slope implies

$$f(\alpha) \leq f(0)$$

for all $\alpha > 0$, with equality only in the degenerate case $P_t = e_k$. Therefore,

$$H(P_{t+1}) \leq H(P_t),$$

with strict inequality whenever the update is non-trivial. □

Remark 1. *Proposition 1 formalizes the intuition underlying our intrinsic penalty. Introducing novel states strictly increases visitation entropy, whereas repeatedly revisiting sufficiently frequent states reduces entropy and concentrates probability mass. Penalizing repeated observations therefore discourages unproductive revisitation and promotes broader state-space exploration.*

4 Experiments

In this section, we report the experimental results of our proposed approach of productive state-space exploration. We evaluate our method against several recent baselines across two interactive agent benchmarks: `ALFWorld` (Shridhar et al., 2020) and `WebShop` (Yao et al., 2022a). With this, we aim to demonstrate two main benefits of our proposed approach. First, the performance improvements due to the intrinsic reward signals. Second, the exploration behaviour and its usefulness in overcoming the repeated actions and looping behaviour of RL trained LLM agents.

4.1 Baselines

For the performance comparison, we benchmark against several state-of-the-art recent baseline models. These include prompting frameworks and those trained with RL. The prompting baselines are Geminie-2.5 (Team et al., 2023), GPT-4o (Achiam et al., 2023), Qwen-2.5 (Qwen et al., 2025), ReAct (Yao et al., 2022b) and Reflexion (Shinn et al., 2023) frameworks. The RL trained baselines include PPO (Schulman et al., 2017), RLOO (Kool et al., 2019; Ahmadian et al., 2024), GRPO (Shao et al., 2024), GiGPO (Feng et al., 2026), ProxMo (Fang et al., 2026). The most recent and relevant baselines for our approach are GiGPO and ProxMo, since we aim to address challenges in multi-turn LLM agent training with RL.

4.2 Training Details:

We train Qwen-2.5-1.5B-Instruct model with the proposed state-space exploration rewards along with Proximal Policy Optimization (PPO) based RL training. We run all our experiments on top of *verl-agent* framework released by GiGPO (Feng et al., 2026). We train for 300 steps with all other hyperparameters kept same as in the framework. We set the maximum steps at 50 for `ALFWorld` and 15 for `WebShop` as with GiGPO. The strength of state-space exploration penalty, γ, as defined in Equation 1, is set $\frac{1}{40}$ of the final success reward.

Table 1: Performance on ALFWorld and WebShop. Following GiGPO, for ALFWorld environment we report the average success rate (%) for each subtask along with the overall result. For WebShop environment, we report the average score and the average success rate (%).

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Closed-Source Model										
Prompting	GPT-4o	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen2.5-1.5B-Instruct										
Prompting	Qwen2.5	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL Training	PPO (with critic)	$64.8_{(3.5)}$	$40.5_{(6.9)}$	$57.1_{(4.9)}$	$60.6_{(6.6)}$	$46.4_{(4.0)}$	$47.4_{(1.9)}$	$54.4_{(3.1)}$	$73.8_{(3.0)}$	$51.5_{(2.9)}$
RL Training	RLOO	$88.3_{(3.0)}$	$52.8_{(8.6)}$	$71.0_{(5.9)}$	$62.8_{(8.7)}$	$66.4_{(5.5)}$	$56.9_{(4.7)}$	$69.7_{(2.5)}$	$73.9_{(5.6)}$	$52.1_{(6.7)}$
RL Training	GRPO	$85.3_{(1.5)}$	$53.7_{(8.0)}$	$84.5_{(6.8)}$	$78.2_{(7.9)}$	$59.7_{(5.0)}$	$53.5_{(5.6)}$	$72.8_{(3.6)}$	$75.8_{(3.5)}$	$56.8_{(3.8)}$
RL Training	GiGPO	$94.4_{(5.9)}$	$67.5_{(4.6)}$	$94.8_{(3.8)}$	$94.4_{(7.8)}$	$79.8_{(4.7)}$	$76.4_{(5.4}$	$86.7_{(1.7)}$	$83.1_{(1.6)}$	$65.0_{(3.2)}$
RL Training	ProxMO	$94.3_{(0.0)}$	$92.9_{(0.0)}$	$89.3_{(0.0)}$	$92.2_{(0.0)}$	$\mathbf{89.5}_{(0.0)}$	$87.0_{(0.0)}$	$90.6_{(0.0)}$	$85.3_{(0.0)}$	$67.1_{(0.0)}$
RL Training	**OURS**	$\mathbf{99.5}_{(0.7)}$	$\mathbf{95.5}_{(6.4)}$	$\mathbf{99.7}_{(0.4)}$	$\mathbf{97.1}_{(2.9)}$	$87.4_{(1.6)}$	$\mathbf{83.9}_{(3.7)}$	$\mathbf{94.0}_{(1.19)}$	$\mathbf{89.2}_{(1.1)}$	$\mathbf{78.3}_{(1.2)}$

Table 2: Exploration efficiency on ALFWorld and WebShop. We compare our method with GiGPO using three metrics: mean episode length, invalid action rate, and repetitive action rate. Lower values indicate more productive exploration.

Method	ALFWorld			WebShop		
	Mean Episode Length	Invalid Action Rate	Repetitive Action Rate	Mean Episode Length	Invalid Action Rate	Repetitive Action Rate
GiGPO	32.27	4.08	45.31	5.53	0.14	43.25
Ours	**22.02**	**2.79**	**11.53**	**5.18**	**0.09**	**40.83**

4.3 Results

Table 1 compares our proposed method (labelled **Ours**) against existing prompting-based and reinforcement learning (RL) training approaches on the ALFWorld and WebShop benchmarks. Following prior work, we report the average success rate for each ALFWorld subtask together with the overall success rate (*All*). For WebShop, we report both the average task score and the average success rate. For RL-based methods, results are averaged across 3 runs and reported with standard deviation to evaluate both effectiveness and optimization stability.

We primarily compare our approach against recent RL optimization methods, namely GiGPO (Group-in-Group Policy Optimization) and ProxMO (Fang et al., 2026), which represent strong baselines for LLM agent training. Compared to GiGPO, our method achieves consistent gains across all ALFWorld subtasks, improving the overall success rate from 86.7% to 94.0%. The improvements are particularly pronounced on challenging long-horizon tasks such as *Cool* (+7.6%) and *Pick2* (+7.5%), indicating stronger sequential reasoning and more effective action planning. On WebShop, our intrinsic-rewards approach improves the average task score from 83.1 to 89.2 and substantially increases the success rate from 65.0% to 78.3%, demonstrating superior capability in complex web-based interaction tasks.

Compared to ProxMO, a recent proximity-based multi-turn optimization framework designed for practical credit assignment in LLM agent training, our approach still achieves better overall performance across both benchmarks. Specifically, it improves the overall ALFWorld success rate from 90.6% to 94.0%, while increasing the WebShop success rate from 67.1% to 78.3%. The improvements on ALFWorld are especially significant, suggesting that our method generalizes more effectively in environments requiring long-horizon exploration, recovery from intermediate mistakes, and consistent multi-turn decision making.

4.4 Intrinsic Productive Exploration

To look beyond overall success rates, we evaluate the behavioral dynamics of our agent against the competitive baseline GiGPO (Feng et al., 2026). This analysis focuses on *intrinsic productive exploration*—the agent's capacity to autonomously discover optimal state-action trajectories through targeted, purposeful environment interactions, rather than wasting computational steps on redundant or non-vindicating behaviors. We define three behav-

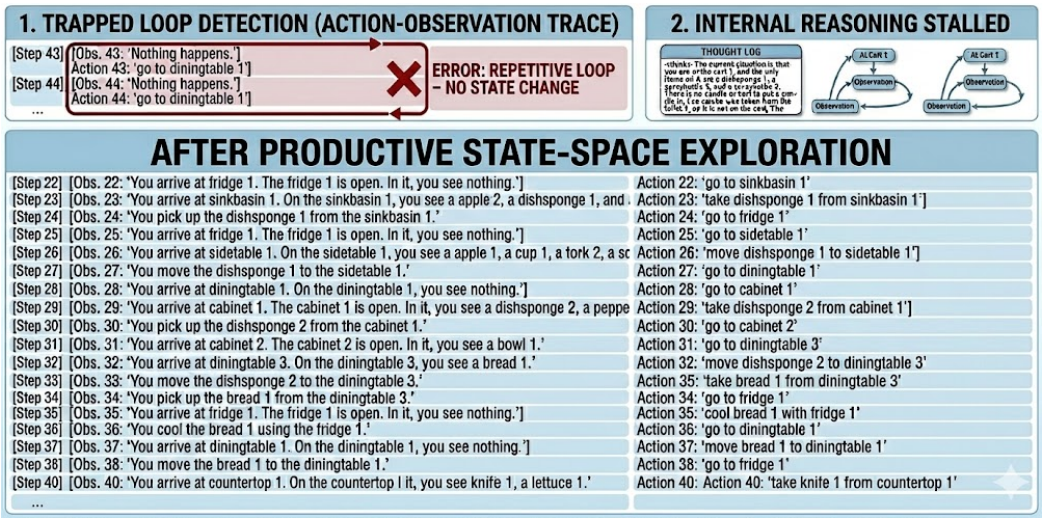

Figure 2: An example output of the multi-turn interaaction of the llm agent after training with conditional state-space exploration with intrinsic rewards. The qualitative outputs show that much of the repetitive actions and looping behaviour is absent from the trajectory.

ioral diagnostics across the `ALFWorld` and `WebShop` benchmarks to quantify the efficiency and productivity of the agent's intrinsic exploration strategy:

- **Mean Episode Length:** The average number of sequential steps to complete a task instance.
- **Invalid Action Rate:** The frequency of actions that fail (violate action preconditions).
- **Repetitive Action Rate:** The frequency of steps spent executing identical actions. We consider an action repetitive when either the action or the resulting observation is repeated in the trajectory.

The empirical results detailing these behavioral metrics are shown in Table 2. In `ALFWorld`, our method compresses the Mean Episode Length from ≈ 32 steps down to ≈ 22 relative to GiGPO. Similar trend persists in the `WebShop`, where the gains are not as profound as in `ALFWorld`, however help in score improvement compared to GIGPO. More broadly, the results suggest:

- **Learning of Environment Dynamics:** The significant improvement in Mean Episode Length and Invalid Action Rate shows that the agent is able to better understand the environment dynamics and can often avoid taking actions which fail to satisfy action preconditions.
- **Loop Resolution and Novelty Seeking:** The most supporting evidence of productive exploration is revealed by the *Repetitive Action Rate* in `ALFWorld`. While `GiGPO` spends 45.31% of its execution trajectory repeating identical commands, our approach drops this rate dramatically to 11.53%. This absolute reduction of 33.78% demonstrates that our fine-grained credit assignment inherently disincentivizes stale, repeated states. Instead, it forces the agent to continuously break out of loops and seek novel, progressive environmental interactions.

Both the above inferences can also be seen qualitative action trajectories as shown in Figure 2. The stark contrast in the decrease in the number of invalid actions relative to Figure 1 suggests our approach of penalizing actions which repeat states is indeed helpful in reducing the redundancy of the agent's trajectory.

5 Conclusion

In this paper, we addressed the tendency of RL-trained LLM agent policies to repeat invalid actions and fall into action-loops in multi-turn long horizon environments. We introduced a novel conditional state-space exploration framework driven by an information-aware intrinsic reward. Instead of directly maximizing unconstrained state entropy which risks driving the agent toward task-irrelevant trajectories, our approach elegantly reframes exploration by penalizing actions that produce redundant state information. Our experimental evaluation demonstrates that this framework achieves substantial performance gains and significantly mitigates action repetition and circular reasoning patterns.

Limitations

We discuss some of the limitations of our work below.

Our experiments are conducted primarily on smaller reasoning models. Because exploration strategies are inherently influenced by a model's overall capacity and pre-trained priors, larger instruction-tuned models may exhibit fewer failures than those documented in this study. Growth in model scale often mitigates baseline reasoning bottlenecks; nevertheless, optimizing smaller models remains highly valuable. Enhancing the capabilities of these smaller architectures enables more practical, targeted applications, particularly in hardware-constrained environments where deployment efficiency is critical.

Additionally, this work focuses specifically on text-based environments within AlfWorld and WebShop, which feature relatively small, well-defined action spaces that lend themselves well to set representations. Environments with larger, continuous, or stochastic action spaces may exhibit vastly different dynamics under this set-based state representation, potentially limiting direct generalizability. Additionally, our framework explicitly penalizes looped actions, such as state revisitation, as suboptimal behaviors. While this assumption holds in the tested benchmarks, certain environments legitimately require revisitation, such as those involving continuous task monitoring or iterative status verification. In such complex scenarios, our current reward design may lack the necessary robustness and could inadvertently degrade agent performance.

Regardless of these limitations, our approach provides a structurally sound framework for agent reasoning and state representation in multi-turn settings. Ultimately, the insights gained from our set-based representation and reward design establish a strong foundation for future scaling and adaptation across more complex, open-ended environments.

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.

Yonatan Ashlag, Uri Koren, Mirco Mutti, Esther Derman, Pierre-Luc Bacon, and Shie Mannor. 2026. State entropy regularization for robust reinforcement learning. *Advances in Neural Information Processing Systems*, 38:21018–21046.

Arthur Aubret, Laetitia Matignon, and Salima Hassas. 2019. A survey on intrinsic motivation in reinforcement learning. *arXiv preprint arXiv:1908.06976*.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. 2016. Unifying count-based exploration and intrinsic motivation. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, page 1479–1487, Red Hook, NY, USA. Curran Associates Inc.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. 2019. Large-scale study of curiosity-driven learning. In *ICLR*.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2023. Grounding large language models in interactive environments with online reinforcement learning. In *International conference on machine learning*, pages 3676–3713. PMLR.

Nuttapong Chentanez, Andrew Barto, and Satinder Singh. 2004. Intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 17.

Zheng Ding and Weirui Ye. 2026. TreeGRPO: Tree-advantage GRPO for online RL post-training of diffusion models. In *The Fourteenth International Conference on Learning Representations*.

Yangyi Fang, Jiaye Lin, Xiaoliang Fu, Cong Qin, Haolin Shi, Chang Liu, and Peilin Zhao. 2026. Proximity-based multi-turn optimization: Practical credit assignment for llm agent training. *arXiv preprint arXiv:2602.19225*.

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.

Peiyuan Feng, Yichen He, Guanhua Huang, Yuan Lin, Hanchong Zhang, Yuchen Zhang, and Hang Li. 2024. Agile: A novel reinforcement learning framework of llm agents. *Advances in Neural Information Processing Systems*, 37:5244–5284.

Jingtong Gao, Ling Pan, Yejing Wang, Rui Zhong, Chi Lu, Maolin Wang, Qingpeng Cai, Peng Jiang, and Xiangyu Zhao. 2025. Navigate the unknown: Enhancing llm reasoning with intrinsic motivation guided exploration. *arXiv preprint arXiv:2505.17621*.

Shengyue Guan, Jindong Wang, Jiang Bian, Bin Zhu, Jian-Guang Lou, and Haoyi Xiong. 2026. Evaluating llm-based agents for multi-turn conversations: A survey. *ACM Transactions on Intelligent Systems and Technology*, 17(4):1–40.

Vivek Kalyan and Martin Andrews. 2025. Reinforcement learning for long-horizon multi-turn search agents. *arXiv preprint arXiv:2510.24126*.

Dongyoung Kim, Jinwoo Shin, Pieter Abbeel, and Younggyo Seo. 2023. Accelerating reinforcement learning with value-conditional state entropy exploration. *Advances in Neural Information Processing Systems*, 36:31811–31830.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 reinforce samples, get a baseline for free!

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. 2025. Reinforcement learning meets large language models: A survey of advancements and applications across the llm lifecycle. *arXiv preprint arXiv:2509.16679*.

Mirco Mutti, Lorenzo Pratissoli, and Marcello Restelli. 2021. Task-agnostic exploration via policy gradient of a non-parametric state entropy estimate. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 9028–9036.

Narjes Nourzad and Carlee Joe-Wong. 2026. MIRA: Memory-integrated reinforcement learning agent with limited LLM guidance. In *The Fourteenth International Conference on Learning Representations*.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pages 2778–2787. PMLR.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, and 1 others. 2024. Multi-turn reinforcement learning with preference human feedback. *Advances in Neural Information Processing Systems*, 37:118953–118993.

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*.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.

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*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Elizaveta Tennant, Stephen Hailes, and Mirco Musolesi. 2025. Moral alignment for llm agents. In *International Conference on Learning Representations*, volume 2025, pages 97477–97510.

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*.

Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, and 1 others. 2025. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*.

Yufei Xiang, Yiqun Shen, Yeqin Zhang, and Nguyen Cam-Tu. 2024. Retrospex: Language agent meets offline reinforcement learning critic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4650–4666.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N Metaxas. 2025. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning. *arXiv preprint arXiv:2509.22576*.

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*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Yifei Zhou and Andrea Zanette. 2024. Archer: training language model agents via hierarchical multi-turn rl. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

A Appendix

A.1 Datasets:

ALFWorld. `ALFWorld` (Shridhar et al., 2020) is a text-based interactive environment designed for studying embodied decision making and long-horizon reasoning in language agents. Derived from the ALFRED benchmark, it converts household manipulation tasks into a partially observable textual setting where agents interact through natural language actions and observations. Tasks typically involve navigation, object manipulation, and sequential planning over multiple turns, making the environment particularly suitable for evaluating reinforcement learning methods for LLM agents under sparse rewards and delayed feedback. Concretely, `ALFWorld` contains 6 tasks types of various difficulty. The tasks are:

- Pick & Place: agent must find an object of a specified type, pick it up, find the correct location to place it, and place it down there.
- Examine in Light: agent must find an object of a specified type, locate and turn on a light source with the object still in hand.
- Clean & Place: agent must find an object of a specified type, pick it up, move to a sink or a basin, turn on the faucet and wash the object, find the correct location to place it, and place it down there.
- Heat & Place: agent must find an object of a specified type, pick it up, move to the microwave, heat the object by turning on the microwave, find the correct location to place it, and place it down there.
- Cool & Place: agent must find an object of a specified type, pick it up, move to the fridge, place the object in the fridge to cool it, find the correct location to place it, and place it down there.
- Pick Two & Place: agent must find an object of a specified type, find the correct location to place it, place it down there, look for another object of a specified type, pick up the object, return to the previous location, place it down there together with the first object.

WebShop. `WebShop` (Yao et al., 2022a) is an e-commerce interaction benchmark that evaluates language agents on realistic online shopping tasks. In this environment, agents must search through large collections of products, inspect item descriptions and reviews, and iteratively refine decisions to satisfy user instructions. Unlike embodied environments, WebShop emphasizes retrieval, preference understanding, and multi-step reasoning in semi-structured web settings. The benchmark is widely used for evaluating the effectiveness of sequential decision-making and reinforcement learning approaches for interactive LLM agents. Concretely, the `WebShop` dataset contains 1,000,000 products, 670 unique attributes, 842,849 unique options, extracted publicly from Amazon. It presents over 12,000 annotated trajectories and an action space of 8 valid actions.

A.2 Additional Qualitative Results:

In Figure 4, we further show some of the example outputs of the LLM agent trained with intrinsic productive exploration rewards framework.

A.3 Declaration

We have cited the models and code used in Section 4. We have used ChatGPT, Gemini and Claude for polishing writing and generation of some figures.

Prompt Template for ALFWorld

You are an expert agent operating in the ALFRED embodied Environment. Your task is to: {task_description}. Prior to this stepyou have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}. You are now at step {current_step} and your current observent observation is: {current_observation}. Your admissible actions for the current situation [{admissible_actions}].
Now it's your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Prompt Template for WebShop

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: {task_description}. Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history}. You are now at step {current_step} and your current observation is: {current_observation}. Your admissible actions for the current situation are: [{available_actions}].
Now it's your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process MUST be enclosed within <think> </think> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Figure 3: Prompts used for both `ALFWorld` and `WebShop` environments.

AFTER PRODUCTIVE STATE-SPACE EXPLORATION

[Step 6]	[Obs. 6: 'You pick up the vase 3 from the diningtable 1.']	Action 6: 'go to desk 1'
[Step 7]	[Obs. 7: 'You arrive at desk 1. On the desk 1, you see a box 2.']	Action 7: 'move vase 3 to desk 1'
[Step 8]	[Obs. 8: 'You move the vase 3 to the desk 1.']	Action 8: 'go to sofa 1'
[Step 9]	[Obs. 9: 'You arrive at sofa 1. On the sofa 1, you see a pillow 2, and a pillow 1.']	Action 9: 'go to desk 1'
[Step 10]	[Obs. 10: 'You arrive at desk 1. On the desk 1, you see a box 2, and a vase 3.']	Action 10: 'take vase 3 from desk 1'
[Step 11]	[Obs. 11: 'You pick up the vase 3 from the desk 1.']	Action 11: 'go to desk 1'
[Step 12]	[Obs. 12: 'Nothing happens.']	Action 12: 'go to shelf 1'
[Step 13]	[Obs. 13: 'You arrive at shelf 1. On the shelf 1, you see a remotecontrol 1, a vase 2, and a vase 1.']	Action 13: 'move vase 3 to shelf 1'
[Step 14]	[Obs. 14: 'You move the vase 3 to the shelf 1.']	Action 14: 'go to shelf 2'
[Step 15]	[Obs. 15: 'You arrive at shelf 2. On the shelf 2, you see a creditcard 1, and a statue 1.']	Action 15: 'go to shelf 3'
[Step 16]	[Obs. 16: 'You arrive at shelf 3. On the shelf 3, you see a statue 2.']	Action 16: 'go to sofa 2'
[Step 17]	[Obs. 17: 'You arrive at sofa 2. On the sofa 2, you see a laptop 2.']	Action 17: 'go to tvstand 1'
[Step 18]	[Obs. 18: 'You arrive at tvstand 1. On the tvstand 1, you see a cellphone 3, a television 1, and a vase 4.']	Action 18: 'take vase 4 from tvstand 1'
[Step 19]	[Obs. 19: 'You pick up the vase 4 from the tvstand 1.']	Action 19: 'go to desk 1'
[Step 20]	[Obs. 20: 'You arrive at desk 1. On the desk 1, you see a box 2.']	Action 20: 'move vase 4 to desk 1'
[Step 21]	[Obs. 21: 'You move the vase 4 to the desk 1.']	Action 21: 'go to diningtable 1'
[Step 22]	[Obs. 22: 'You arrive at diningtable 1. On the diningtable 1, you see a box 1, a cellphone 1, a laptop 3, and a laptop 1.']	Action 22: 'go to shelf 1'
[Step 23]	[Obs. 23: 'You arrive at shelf 1. On the shelf 1, you see a remotecontrol 1, a vase 3, a vase 2, and a vase 1.']	Action 23: 'take vase 1 from shelf 1'
[Step 24]	[Obs. 24: 'You pick up the vase 1 from the shelf 1.']	Action 24: 'go to desk 1'
[Step 25]	[Obs. 25: 'You arrive at desk 1. On the desk 1, you see a box 2, and a vase 4.']	Action 25: 'move vase 1 to desk 1'

✔ TASK ACHIEVED
You have put two vase in desk.

AFTER PRODUCTIVE STATE-SPACE EXPLORATION

[Step 15]	[Obs. 15: 'You arrive at fridge 1. The fridge 1 is open. In it, you see a bowl 2, a bread 3, a egg 1, a mug 2, and a plate 1.']	Action 15: 'go to cabinet 3'
[Step 16]	[Obs. 16: 'You arrive at cabinet 3. The cabinet 3 is closed.']	Action 16: 'open cabinet 3'
[Step 17]	[Obs. 17: 'You open the cabinet 3. The cabinet 3 is open. In it, you see nothing.']	Action 17: 'go to countertop 1
[Step 18]	[Obs. 18: 'You arrive at countertop 1. On the countertop 1, you see a bread 2, a butterknife 2, a fork 2, a fork 1, a knife 3, a knife 1, a lettuce 2, a potato 1, a saltshaker 3, a soapbottle 3, a tomato 3, a tomato 2, and a tomato 1.']	Action 18: 'go to fridge 1'
[Step 19]	[Obs. 19: 'You arrive at fridge 1. The fridge 1 is open. In it, you see a bowl 2, a bread 3, a egg 1, a mug 2, and a plate 1.']	Action 19: 'go to countertop 2'
[Step 20]	[Obs. 20: 'You arrive at countertop 2. On the countertop 2, you see a bread 1, a cup 1, a dishsponge 2, a dishsponge 1, a knife 2, a lettuce 1, a mug 1, a peppershaker 3, a peppershaker 1, a spoon 2, and a spoon 1.']	Action 20: 'take cup 1 from countertop 2'
[Step 21]	[Obs. 21: 'You pick up the cup 1 from the countertop 2.']	Action 21: 'go to fridge 1'
[Step 22]	[Obs. 22: 'You arrive at fridge 1. The fridge 1 is open. In it, you see a bowl 2, a bread 3, a egg 1, a mug 2, and a plate 1.']	Action 22: 'cool cup 1 with fridge 1'
[Step 23]	[Obs. 23: 'You cool the cup 1 using the fridge 1.']	Action 23: 'go to countertop 1'
[Step 24]	[Obs. 24: 'You arrive at countertop 1. On the countertop 1, you see a bread 2, a butterknife 2, a fork 2, a fork 1, a knife 3, a knife 1, a lettuce 2, a potato 1, a saltshaker 3, a soapbottle 3, a tomato 3, a tomato 2, and a tomato 1.']	Action 24: 'move cup 1 to countertop 1'
[Step 25]	[Obs. 25: 'You move the cup 1 to the countertop 1.']	Action 25: 'go to fridge 1'
[Step 26]	[Obs. 26: 'You arrive at fridge 1. The fridge 1 is open. In it, you see a bowl 2, a bread 3, a egg 1, a mug 2, and a plate 1.']	Action 26: 'go to countertop 2'
[Step 27]	[Obs. 27: 'You arrive at countertop 2. On the countertop 2, you see a bread 1, a dishsponge 2, a dishsponge 1, a knife 2, a lettuce 1, a mug 1, a peppershaker 3, a peppershaker 1, a spoon 2, and a spoon 1.']	Action 27: 'take mug 1 from countertop 2'
[Step 28]	[Obs. 28: 'You pick up the mug 1 from the countertop 2.']	Action 28: 'go to fridge 1'
[Step 29]	[Obs. 29: 'You arrive at fridge 1. The fridge 1 is open. In it, you see a bowl 2, a bread 3, a egg 1, a mug 2, and a plate 1.']	Action 29: 'go to microwave 1'
[Step 30]	[Obs. 30: 'You arrive at microwave 1. The microwave 1 is open. In it, you see a apple 1.']	Action 30: 'move mug 1 to microwave 1'
[Step 31]	[Obs. 31: 'You move the mug 1 to the microwave 1.']	Action 31: 'take apple 1 from microwave 1'
[Step 32]	[Obs. 32: 'You pick up the apple 1 from the microwave 1.']	Action 32: 'go to fridge 1'
[Step 33]	[Obs. 33: 'You arrive at fridge 1. The fridge 1 is open. In it, you see a bowl 2, a bread 3, a egg 1, a mug 2, and a plate 1.']	Action 33: 'cool apple 1 with fridge 1'
[Step 34]	[Obs. 34: 'You cool the apple 1 using the fridge 1.']	Action 34: 'go to countertop 2'
[Step 35]	[Obs. 35: 'You arrive at countertop 2. On the countertop 2, you see a bread 1, a dishsponge 2, a dishsponge 1, a knife 2, a lettuce 1, a peppershaker 3, a peppershaker 1, a spoon 2, and a spoon 1.']	Action 35: 'move apple 1 to countertop 2'

✔ TASK ACHIEVED
You have cooled some apple and put it in countertop.

Figure 4: Additional example outputs of the multi-turn interaction of the llm agent on the `ALFWorld` datset after training with conditional state-space exploration with intrinsic rewards.