
Beyond Trajectory-Level Attribution: Graph-Based Credit Assignment for Agentic Reinforcement Learning

Xin Cheng¹ Shuo He¹ Lang Feng¹ HaiYang Xu² Ming Yan² Lei Feng³ Bo An¹

Abstract

Group-based reinforcement learning (RL) methods have achieved remarkable success in improving the performance of large language models (LLMs) and have been rapidly extended to agentic tasks. However, their credit assignment relies heavily on coarse-grained trajectory-level attribution according to final outcomes, making it difficult to capture the contribution of individual steps, such as valuable steps obscured within failed trajectories. *To uncover latent information and enable more faithful step-level credit assignment*, we propose Graph-based Group Policy Optimization (GraphGPO), which first aggregates all rollout trajectories into a unified state-transition graph and then estimates the distance from each state to the task goal using the global information encoded in the graph. Finally, GraphGPO assigns credit to each edge by estimating a graph-based advantage, based on how much the transition reduces the distance to the task goal. In this way, GraphGPO significantly improves training efficiency and achieves state-of-the-art performance across a range of challenging benchmarks. Code is available at <https://github.com/langfengQ/verl-agent/tree/master/recipe/GraphGPO>.

1. Introduction

In recent years, Large Language Models (LLMs) (Achiam et al., 2024; Gemini et al., 2023; Yang et al., 2025; Liu et al., 2025a) have undergone rapid iteration and evolution, enabling them to move beyond static language understanding (Devlin et al., 2019; Radford et al., 2019) toward more complex reasoning and decision-making tasks (Wei et al., 2022; Ahn et al., 2022). As a result of these advances,

¹Nanyang Technological University, Singapore ²Tongyi Lab, Alibaba Group ³Southeast University, China. Correspondence to: Lei Feng <fenglei@seu.edu.cn>.

LLMs are increasingly deployed as agents that can perceive, reason, and act in complex, open-ended environments, enabling them to tackle tasks that require long-horizon planning and sequential decision making, spanning embodied tasks (Wang et al., 2023; Driess et al., 2023), web or mobile interactive environments (Furuta et al., 2024; Wang et al., 2024a; Zheng et al., 2024a; Gou et al., 2025; Feng et al., 2025a), as well as interactive games (Hafner et al., 2023; Xu et al., 2024; Wang et al., 2025d; Liu et al., 2025b) and tool-augmented reasoning scenarios (Schick et al., 2023; Paranjape et al., 2023; Qian et al., 2025; Xue et al., 2025).

Reinforcement Learning (RL) (Sutton et al., 1998) has demonstrated its ability to drive agents toward human-level performance through landmark achievements (Silver et al., 2018). As learning paradigms evolved toward large-scale models, RL has re-emerged as a crucial post-training stage for LLMs to enhance performance, resulting in frontier models such as OpenAI o1 (OpenAI, 2024) and DeepSeek R1 (Guo et al., 2025). Notably, group-based RL methods (Yu et al., 2025b; Zheng et al., 2025a; Cui et al., 2025), such as GRPO (Shao et al., 2024), have gained prominence by discarding the resource-intensive critic model used in traditional Actor-Critic frameworks (Konda & Tsitsiklis, 1999; Schulman et al., 2015a; 2017; Haarnoja et al., 2018). These methods rely on verifiable rewards and intra-group statistics, which reduces memory consumption and enables efficient scaling to LLMs. More recently, several studies (Wang et al., 2025e; Jin et al., 2025; Luo et al., 2025a; Yu et al., 2025a; Wang et al., 2025c; Feng et al., 2025b; He et al., 2026) have begun to extend group-based RL to multi-turn agentic tasks.

Although existing group-based RL methods have shown promising performance in multi-turn agentic tasks, they rely on an implicit but restrictive assumption: *the quality of each step can be inferred solely from the final success or failure of the trajectory it belongs to*. Under this assumption, all steps within a successful trajectory receive positive credit, while all steps within a failed trajectory are penalized, regardless of their actual contribution to task progress. However, this trajectory-level attribution is fundamentally misaligned with the multi-turn agentic tasks. A successful trajectory may contain redundant or erroneous steps that do not advance the agent toward the goal, while a failed trajectory may in-

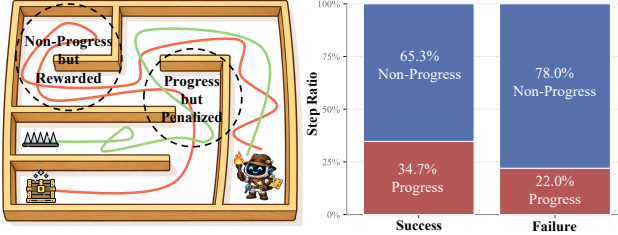


Figure 1. **Left:** When one successful trajectory and one failed trajectory are sampled, non-progress steps within the successful trajectory receive positive credit, while progress steps within the failed trajectory are penalized. **Right:** Step statistics showing the proportion of progress and non-progress steps in early-stage training of ALFWorld (rollout $M = 8$ and maximum step $T = 50$), reported separately for successful and failed trajectories.

clude decisive and valuable steps whose effects are negated by later mistakes. Figure 1 presents step-level statistics, showing that approximately 22.0% of the steps in failed trajectories contribute to task progress, whereas about 65.3% of the steps in successful trajectories do not meaningfully advance the task. As a result, credit assignment that relies solely on trajectory-level attribution fails to capture such latent information, leading to coarse and noisy step-level signals and, consequently, inefficient learning.

To address these problems, we propose Graph-based Group Policy Optimization (GraphGPO), a novel RL method that departs from trajectory-level attribution dependency and instead aggregates all rollout trajectories into a unified state-transition graph, enabling global, structure-aware finer-grained credit assignment. Concretely, GraphGPO represents environment states (e.g., the current prompt or historical interactions) as nodes, and models each step across all trajectories as a directed edge, thereby replacing isolated trajectory representations with a unified dynamic state-transition graph. Leveraging the global information from the graph, GraphGPO then captures the relationship between each state and the task goal to assign step-level rewards, without relying on trajectories. Finally, outgoing edges originating from the same state are grouped together to estimate graph-based advantages, which are then used to optimize the agent. Compared with previous methods, GraphGPO leverages global experiential information aggregated from all trajectories to capture latent relationships among steps, enabling more faithful step-level credit assignment and significantly improving training efficiency. Moreover, GraphGPO remains critic-free while introducing only a negligible amount of graph computation, achieving stable convergence without incurring additional overhead.

2. Related Work

Reinforcement learning for LLMs. Early applications of RL to LLMs primarily focused on RL from human feed-

back (RLHF) (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023; Zhang et al., 2025), where human preferences are used to align LLMs. More recently, RL with verifiable rewards (RLVR) (Kool et al., 2019; Ahmadian et al., 2024; Shao et al., 2024; Liu et al., 2025c; Lin et al., 2025; Su et al., 2025) has gained increasing attention, replacing human feedback with automatically computable rewards. This paradigm has been shown effective for improving reasoning capabilities (Team et al., 2025; Guo et al., 2025; Wen et al., 2025) in domains such as mathematics (Lightman et al., 2023), code generation (Le et al., 2022; Sun et al., 2024; Jiang et al., 2025), tool use (Schick et al., 2023; Qin et al., 2023; Wang et al., 2025a; Qian et al., 2025) and search (Sun et al., 2025; Song et al., 2025; Zheng et al., 2025b).

Reinforcement Learning for LLM-based Agents. Beyond static text generation, RL has increasingly been used to enhance the capabilities of LLM-based agents in dynamic, open-ended environments. Early studies (Mnih et al., 2015; Tan et al., 2024; Wen et al., 2024; Zhai et al., 2024; Bai et al., 2024; Wang et al., 2024c) typically employ value-based critics (Schulman et al., 2017; Peng et al., 2019) to guide agent learning in domains such as Android device control (Rawles et al., 2023), embodied environments (Shridhar et al., 2021), and card games (Brockman et al., 2016). More recent work has begun to extend RL to real-world agentic settings, including software engineering (Yang et al., 2024; Zheng et al., 2024b; Da et al., 2025), GUI control (Lu et al., 2025; Wang et al., 2025b; Lai et al., 2025; Shi et al., 2025), and Model Context Protocol (Luo et al., 2025b; Le et al., 2025; Team & Team, 2025).

3. Preliminaries

In this section, we introduce preliminary knowledge of multi-turn agentic tasks, group-based reinforcement learning, and relevant advantage estimation.

3.1. Multi-turn Agentic Tasks

Let $\mathbf{x} \sim p(X)$ be the task example, and let π_θ denote a LLM-based policy parameterized by θ . In the general agentic setting (Xi et al., 2025; Wang et al., 2024b), the policy needs to interact with the environment multiple steps to accomplish the goal associated with task $\mathbf{x}$. At each step $t = 1, 2, \dots, T$, the policy π_θ observes the current environment state s_t and produces an action $\mathbf{a}_t \in \mathcal{V}^n$ sampled from conditioned distribution $\pi_\theta(\mathbf{a}_t \mid s_t, \mathbf{x})$, where $\mathcal{V}$ denotes the token vocabulary and n is the allowed maximum generation length. The environment then transitions to the next state s_{t+1} . Ultimately, all interactions form a trajectory $\tau = \{(s_1, \mathbf{a}_1), (s_2, \mathbf{a}_2), \dots, (s_T, \mathbf{a}_T)\}$. In many real-world tasks, the reward is provided only at the end of the trajectory depending on whether the goal is successfully accomplished,

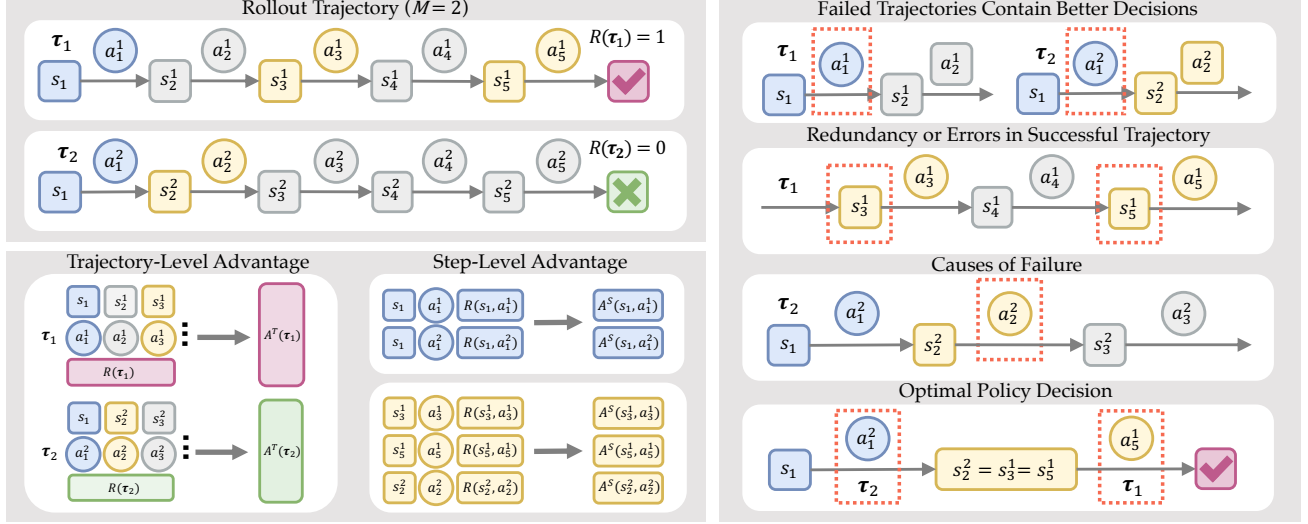


Figure 2. Overview of group-based advantage estimation and existing issues, where squares represent states and circles represent actions. **Top-left:** Rollout trajectories (τ_1 is a **successful** trajectory and τ_2 is a **failed** trajectory), where **blue** squares denote identical states among themselves, **yellow** squares denote another set of identical states, and **gray** represents independent states with no shared states. **Bottom-left:** Trajectory-level and step-level advantage estimation. **Right:** Issues with credit assignment that relies solely on trajectory success in both successful and failed trajectories.

so most intermediate rewards are zero. In this paper, we focus on such sparse, delayed reward settings.

3.2. Group-based Reinforcement Learning

The advantage function (Schulman et al., 2015b; Wang et al., 2016) in RL plays a central role by quantifying how favorable an action is and guiding policy updates. Group-based RL methods, such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024), estimate advantages in a critic-free manner by relying on verifiable rewards and intra-group statistics, thereby simplifying the training architecture. More formally, for a task instance x , a group of outputs $G^o = \{o_1, o_2, \dots, o_M\}$ is sampled from the policy model π_θ . Each output o_m receives a scalar reward $R(o_m)$ based on whether the goal of task is successfully completed by o_m . GRPO then estimates advantages solely from statistics computed within the group:

$$A(o_m) = (R(o_m) - \mu(G^o)) / \sigma(G^o), \quad (1)$$

where $\mu(o)$ and $\sigma(o)$ denote the mean and standard deviation of rewards $R(o)$ within the group G^o . GRPO was originally designed for single-turn settings, but it can be readily extended to multi-turn agentic tasks:

$$A^E(\tau_m) = (R(\tau_m) - \mu(G^E)) / \sigma(G^E), \quad (2)$$

where $G^E = \{\tau_1, \tau_2, \dots, \tau_M\}$ denotes a group of trajectories, and $R(\tau_m)$ is a scalar reward depending on whether the trajectory can successfully accomplish the goal.

For each step (s_t^m, a_t^m) within the same trajectory τ_m , GRPO assigns the same reward $R(\tau_m)$ and episode-level

advantage $A^E(\tau_m)$, which results in overly coarse credit assignment. To address fine-grained credit assignment, Group-in-Group Policy Optimization (GiGPO) (Feng et al., 2025b) introduces a step-level group advantage estimator which groups together all steps with the same state, regardless of whether they come from the same trajectory. The step-level group advantage is then computed as follows:

$$A^S(s_t^m, a_t^m) = (R^S(s_t^m, a_t^m) - \mu(G^S(s_t))) / \sigma(G^S(s_t)),$$

where $G^S(s_t) = \{(s_t^j, a_t^j) \mid s_t^j = s_t, 1 \leq i \leq T, 1 \leq j \leq M\}$ denotes the group of steps that share the same initial state s_t , and $R^S(s_t^m, a_t^m) = \lambda^{T-i} R(\tau_m)$ represents the standard RL discounted reward. Although the step-level credit assignment in GiGPO improves agent performance, it still relies on trajectory-level attribution based on the final outcome, i.e., $R(\tau)$.

4. Proposed Method

In this section, we first discuss the limitations of existing group-based RL methods. We then introduce a novel and effective method that enables faithful step-level credit assignment without incurring additional overhead.

4.1. Limitations of Trajectory-Level Attribution

Although existing group-based RL methods have employed multi-level advantage estimation and provide effective training signals in agentic tasks, their credit attribution still fundamentally relies on trajectory-level outcomes, limiting the fidelity of step-level credit assignment.

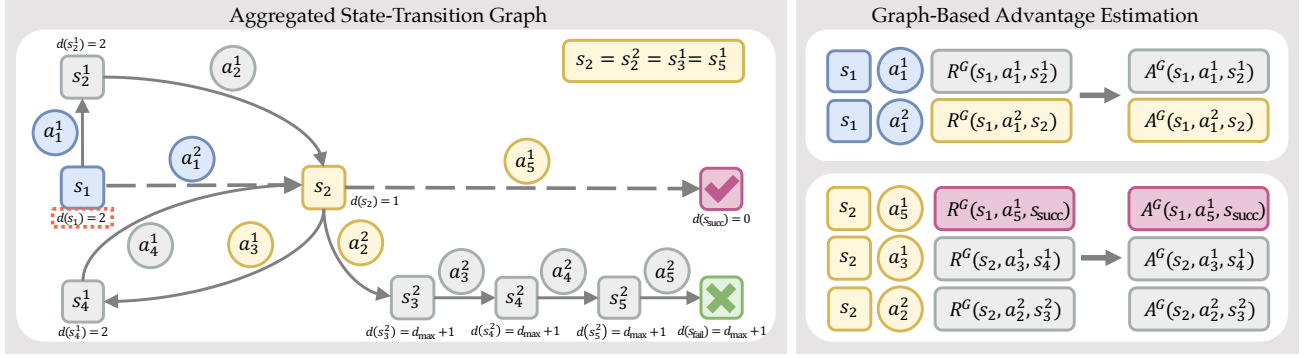


Figure 3. Overview of GraphGPO. For simplicity, we assume all transition costs are unitary, i.e., $c(s, \mathbf{a}) = 1$. **Left:** The aggregated state-transition graph constructed based on states from rollout trajectories, where identical states are merged (e.g., $s_1 = s_1^2 = s_1^3 = s_1^4$). **Right:** Graph-based advantage estimation, where the credit assignment relies on the distance $d(\cdot)$ of the next state. Taking the initial state s_1 as an example, the shortest path to the goal state s^* is $\{(s_1, \mathbf{a}_1^1, s_2), (s_2, \mathbf{a}_5^1, s_{\text{succ}})\}$, yielding $d(s_1) = 2$. Moreover, the maximum shortest-step distance among all states that can reach the goal is $d_{\text{max}} = 2$.

Specifically, current methods assign credit by attributing step quality solely to the final success or failure of a trajectory. As a result, redundant or erroneous steps within successful trajectories are rewarded, while correct steps in failed trajectories are consistently penalized. This attribution misalignment not only impedes effective policy updating but also obscures the underlying reasons for trajectory failures. Figure 2 illustrates these issues. For trajectory τ_1 , although the task is ultimately completed successfully, repeatedly visiting the yellow states (s_3^1 and s_5^1) indicates erroneous and redundant behavior. In contrast, although trajectory τ_2 eventually fails, it takes a more favorable action $\mathbf{a}_2^1$ than τ_1 at the blue state s_1 , enabling the environment to reach the yellow state more quickly. Moreover, the failure of τ_2 may stem from the action $\mathbf{a}_2^2$ in the yellow state s_2^2 , which leads to a successor state s_3^2 that cannot reach the goal state s_{succ} . Therefore, the steps preceding this state should not be directly penalized.

To address these issues, we propose Graph-based Group Policy Optimization (GraphGPO), a novel graph-based credit assignment method that leverages the global state-transition structure to redefine step-level credit attribution, thereby providing more faithful guidance for policy optimization.

4.2. The Aggregated State-Transition Graph

Rather than treating each trajectory independently, we aggregate all rollout trajectories into a unified state-transition graph, which provides a structured representation of the environment dynamics induced by interactions of the policy. This representation enables credit assignment based on the global connectivity between states. Formally, given a set of rollout trajectories $\{\tau_1, \tau_2, \dots, \tau_M\}$ of size M , we construct a directed graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$. The node set $\mathcal{S}$ consists of all states visited in the trajectories, i.e., $\mathcal{S} = \bigcup_{m=1}^M \bigcup_{t=1}^T \{s_t^m\}$. For states that are identical, we merge

them into a single node in the graph. A directed edge $(s, \mathbf{a}, s', c(s, \mathbf{a})) \in \mathcal{E}$ exists if and only if there exists a time step t and a trajectory τ_m containing a transition $\{\dots (s_t^m, \mathbf{a}_t^m), (s_{t+1}^m, \mathbf{a}_{t+1}^m) \dots\}$ such that $s_t^m = s$, $\mathbf{a}_t^m = \mathbf{a}$, and $s_{t+1}^m = s'$, where $c(s, \mathbf{a}) > 0$ denotes the cost incurred when the policy takes action $\mathbf{a}$ at state s . In tool-use scenarios, the cost can be defined as the cost associated with tool usage, such as monetary or time costs. We denote the set of terminal states as $\mathcal{S}_{\text{term}}$, which includes both goal states s_{succ} corresponding to task successful completion and states s_{fail} which exceeds the maximum allowed length. In Figure 3, we show the aggregated state-transition graph constructed based on the successful trajectory τ_1 and failed trajectory τ_2 from Figure 2.

4.3. The Graph-Based Advantage Estimation

The graph-based structured representation provides global dynamic information between states. For example, it allows us to easily identify which states can be reached from any given state, and even how to reach them, even if it requires traversing multiple trajectories. Moreover, it also allows us to identify states from which cannot reach the goal state s_{succ} . These states should be avoided. Most importantly, it enables us to determine which states are closer to goal state s_{succ} . Building on this idea, for any state s , we use the following distance to measure the minimum cost required to reach the goal state s_{succ} :

$$d(s) = \begin{cases} 0, & \text{if } s = s_{\text{succ}}, \\ \min_{(s, \mathbf{a}, s', c) \in \mathcal{E}} (c(s, \mathbf{a}) + d(s')), & \text{if } s \rightsquigarrow s_{\text{succ}}, \\ +\infty, & \text{otherwise,} \end{cases} \quad (3)$$

where $s \rightsquigarrow s_{\text{succ}}$ denotes that there exists a path from s to the goal state s_{succ} in the graph. If $d(s)$ is close to 0, it indicates that state s is close to successfully completing the task. If $d(s)$ is large, it implies that significant cost is required to

complete the task. When $d(s) = +\infty$, it indicates that the state is empirically unable to complete the task, which indicates failure. Naturally, a good action is expected to reduce the distance to the goal while incurring minimal cost. Therefore, for an edge $(s, \mathbf{a}, s', c(s, \mathbf{a})) \in \mathcal{E}$ in the graph $\mathcal{G}$, we define the following graph-based step-level reward:

$$R^G(s, \mathbf{a}, s') = r_{\text{succ}} \omega^{d(s') + c(s, \mathbf{a})}, \quad (4)$$

where $r_{\text{succ}} > 0$ is a scalar denoting the reward for successfully completing the task, which is typically set to 1 or 10 in general agentic settings. $\omega \in (0, 1)$ is a distance discount factor that discounts the reward r_{succ} according to the distance between the resulting state s' and the goal. Additionally, for states with $d(s') = +\infty$, to avoid overly strong penalties that may restrict policy exploration, we replace $+\infty$ with the largest finite distance in the graph plus one, i.e., $d_{\max} + 1$, where $d_{\max} = \max_{s \in \mathcal{S}} d(s)$. The graph-based step-level reward discards feedback based on individual trajectories, and instead leverages the global connectivity of the aggregated graph constructed from all trajectories to provide more faithful step-level credit assignment. We then define the step-level group for each state s in graph as:

$$G^G(s) = \{(s_i, \mathbf{a}, s_j) \mid (s_i, \mathbf{a}, s_j) \in \mathcal{E}, s_i = s\}. \quad (5)$$

Intuitively, this group represents all candidate state transitions that originate from state s , i.e., all edges in the graph that start from s . Once this group is formed, we can estimate the following graph-based advantage for each edge $(s, \mathbf{a}, s') \in \mathcal{E}$ in the graph $\mathcal{G}$:

$$A^G(s, \mathbf{a}, s') = \frac{R^G(s, \mathbf{a}, s') - \mu(G^G(s))}{\sigma(G^G(s))}, \quad (6)$$

where $\mu(G^G(s))$ and $\sigma(G^G(s))$ denote the mean and standard deviation of graph-based step-level rewards within the group $G^G(s)$. For each state s with a step-level group size $|G^G(s)| = 1$, i.e., the state appears only once in collected experience trajectories, the graph-based advantage $A^G(s, \mathbf{a}, s')$ is set to 0.

Through graph-based advantage estimation, we derive step-level credit from global state-transition structure, enabling more faithful credit attribution. Specifically, for the start state s_1 in Figure 3, it has two successor states s_2^1 and s_2 . Since state s_2 is closer to the goal state s_{succ} than state s_2^1 (i.e., $d(s_2) < d(s_2^1)$), the resulting advantage satisfies $A^G(s_1, \mathbf{a}_1^2, s_2) > A^G(s_1, \mathbf{a}_1^1, s_2^1)$. This indicates that action $\mathbf{a}_1^2$ is preferred over $\mathbf{a}_1^1$ at state s_1 , even though $\mathbf{a}_1^2$ comes from the failed trajectory τ_2 . In addition, we can identify action $\mathbf{a}_2^2$ as the cause of the failure, since it transitions to state s_2^2 , which empirically has no path to the goal state s_{succ} (i.e., $d(s_2^2) = d_{\max} + 1$). As a result, the corresponding advantage $A^G(s_2, \mathbf{a}_2^2, s_2^3)$ is the smallest within the group $G^G(s_2)$. Moreover, redundancy or erroneous behavior is reflected as cycles in the graph, such as the loop

$(s_2, \mathbf{a}_3^1, s_4^1), (s_4^1, \mathbf{a}_4^1, s_2)$ in Figure 3. Such transitions inevitably increase the distance ($d(s_4^1) > d(s_2)$) to the goal state s_{succ} and thus receive lower advantages than actions that reduce the distance, e.g., $\mathbf{a}_5^1$. We then present the following proposition, which demonstrates that graph-based advantage estimation assigns higher advantages to actions that more effectively reduce the distance to the goal state.

Proposition 4.1 (Monotonicity of graph-based advantage). *Consider the state s that can reach the goal in the deterministic environment. There are two actions a_{good} and a_{bad} leading state s to next states s'_{good} and s'_{bad} , respectively. Assume that a_{good} moves closer to the goal than a_{bad} , i.e., $d(s'_{\text{good}}) + c(s, a_{\text{good}}) < d(s'_{\text{bad}}) + c(s, a_{\text{bad}})$. The proposed graph-based advantage satisfies*

$$A^G(s, a_{\text{good}}, s'_{\text{good}}) > A^G(s, a_{\text{bad}}, s'_{\text{bad}}). \quad (7)$$

The proof of Proposition 4.1 is provided in Appendix B.1. In particular, credit attribution based on final trajectory outcomes favors actions that merely appear in successful trajectories, whereas graph-based credit attribution prefers actions that move closer to the goal.

4.4. Policy Optimization

To prevent the advantage from degenerating to zero for states with only one outgoing edge (i.e., $|G^G(s)| = 1$), we still incorporate the episode-level advantage, resulting in the following combined advantage:

$$A(s, \mathbf{a}, s') = \beta^G A^G(s, \mathbf{a}, s') + \beta^E A^E(\tau), \quad (8)$$

where the edge $(s, \mathbf{a}, s')$ belongs to the trajectory τ , meaning that there exists a time step t such that τ contains the transition $\{\dots (s_t, \mathbf{a}_t), (s_{t+1}, \mathbf{a}_{t+1}) \dots\}$ with $s_t = s$, $\mathbf{a}_t = \mathbf{a}$, and $s_{t+1} = s'$. β^G and β^E are balancing factors for the advantages A^G and A^E , respectively. Then the final policy optimization objective of GraphGPO is:

$$\mathbb{E} \left[\frac{1}{NT} \sum_{m=1}^M \sum_{t=1}^T \min \left(\rho_{\theta}(\mathbf{a}_t^m) A(s_t^m, \mathbf{a}_t^m, s_{t+1}^m), \right. \right. \quad (9)$$

$$\left. \left. \text{clip}(\rho_{\theta}(\mathbf{a}_t^m), 1 \pm \epsilon) A(s_t^m, \mathbf{a}_t^m, s_{t+1}^m) \right) \right] - \beta \mathbb{D}_{\text{KL}}(\pi_{\theta} \parallel \pi_{\theta_{\text{ref}}}),$$

where $\rho_{\theta}(\mathbf{a}_t^m) = \frac{\pi_{\theta}(\mathbf{a}_t^m | s_t^m, x)}{\pi_{\text{old}}(\mathbf{a}_t^m | s_t^m, x)}$ is the importance sampling ratio and β controls the strength of the KL penalty. We present the pseudo code in Appendix A.

Compared to previous methods, GraphGPO provides more faithful step-level credit assignment by deriving credit from the global state-transition structure rather than trajectory outcomes. We then present the following proposition, which shows that the conditional variance of graph-based step-level feedback is lower than that of trajectory-based step-level feedback.

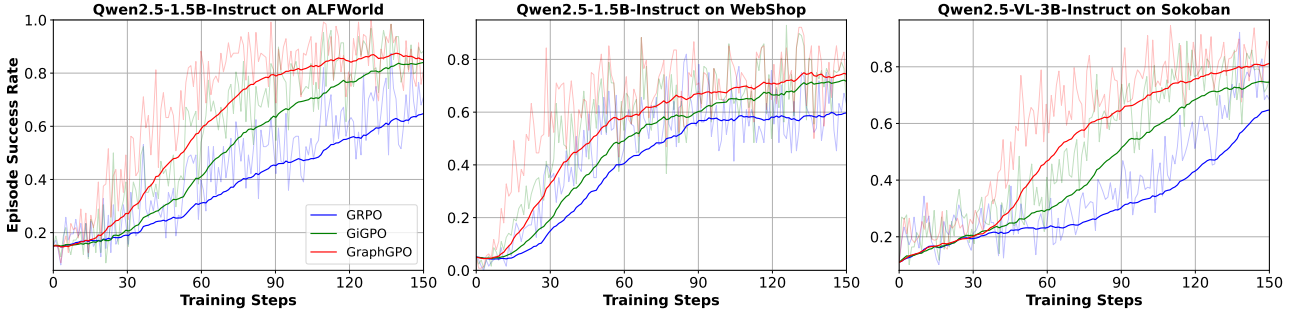


Figure 4. Training episode success rate versus steps for GraphGPO (red), GiGPO (green), and GRPO (blue) on the ALFWorld, WebShop, and Sokoban benchmarks. The lighter curves show the original curves, while the darker curves correspond to exponential moving average (EMA) smoothing with decay $\alpha = 0.95$, highlighting the overall training trends.

Proposition 4.2 (Conditional variance reduction). *Given rollouts sampled from a fixed policy π in the deterministic environment. For a state-action pair (s, a, s') that is visited with non-zero probability, Assume that both successful and failed trajectories pass through (s, a, s') with positive probability and (s, a, s') is independent of step t . For the trajectory-style step feedback $X^S = \eta(t)R(\tau)$ and graph-style step feedback $X^G = R^G(s, a, s)$, the conditional variance hold almost surely:*

$$\text{Var}(X^G|s, a, s') \leq \text{Var}(X^S|s, a, s').$$

where $\eta(t) = \lambda^{T-t}$ for GiGPO, $\eta(t) = 1$ for GRPO.

The proof of Proposition 4.2 is provided in Appendix B.2.

5. Experiments

In this section, we present extensive experiments to demonstrate the effectiveness of our proposed method.

5.1. Experimental Setup

Benchmarks. We evaluate GraphGPO on a set of challenging multi-turn agentic benchmarks, including ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022a). ALFWorld is an embodied environment designed to evaluate the ability of an agent to perform long-horizon, multi-step decision-making. The benchmark contains 3,827 task instances spanning six categories of common household activities. WebShop is a large-scale, web-based interactive environment that tests agents in realistic online shopping scenarios. The environment includes over 1.1 million products and approximately 12,000 user instructions. In addition to text-based agentic environments for LLMs, we also consider vision-language model (VLM) settings in interactive game environments: Sokoban benchmark (Schrader, 2018), where agents must reason over visual observations and plan multi-step actions. Detailed descriptions of benchmark you can find in Appendix C.

Compared Methods. For LLMs, we compare GraphGPO with several competitive baselines, including: 1) Closed-source LLMs: GPT-4o (Achiam et al., 2024) and Gemini-2.5-Pro (Gemini et al., 2023), which are widely adopted advanced models in general-purpose settings. 2) Prompting-based agents: ReAct (Yao et al., 2022b) and Reflexion (Shinn et al., 2023), which rely entirely on in-context information to accomplish multi-turn tasks without parameter updates. 3) RL-based training methods: PPO (Schulman et al., 2017), a widely used critic-based RL algorithm, as well as group-based RL methods, including RLOO (Kool et al., 2019; Ahmadian et al., 2024), GRPO (Shao et al., 2024), and GiGPO (Feng et al., 2025b). For VLMs, we focus on comparisons with group-based RL methods.

Implementation Details. Following prior work (Feng et al., 2025b), we use Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Qwen, 2025) as the base LLMs, and Qwen2.5-VL-3B-Instruct (Bai et al., 2025) as the base VLM. For consistency with existing agent frameworks, the agent retains only the most recent two interaction steps as memory and discards earlier history. In addition, the agent is prompted to first generate its reasoning enclosed within `<think></think>` tags, followed by the action enclosed within `<action></action>` tags (Wei et al., 2022). For all methods, we adopt the same training hyperparameters to ensure fair comparisons. For group-based methods, the rollout group size is set to 8 across all benchmarks. Moreover, the balancing factors β^G and β^E are both set to 1 for GraphGPO. For simplicity, we consider all transition costs are unitary, i.e., $c(s, a) = 1$. Detailed implementation details, including training settings and hyperparameters, are provided in the Appendix C.

5.2. Experimental Results

Performance on Agentic Benchmarks. Table 1 presents the performance comparison between GraphGPO and several competitive baselines on the ALFWorld and WebShop

Table 1. The test performance on ALFWorld and WebShop. For ALFWorld, we report the average success rate (%) for each subtask as well as the overall result. For WebShop, we report the average task score and the average success rate (%). Most results are averaged over 3 random seeds during testing. The best performances are highlighted in bold.

Type	Method	ALFWorld							WebShop	
		Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Succ.
Closed-Source Models										
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	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	82.89(3.62)	82.14(6.37)	73.86(6.84)	78.57(0.00)	77.78(4.54)	71.43(3.89)	77.86(1.33)	84.73(0.49)	71.35(2.05)
RL Training	GiGPO	98.81(1.68)	95.16(3.89)	81.46(0.56)	78.57(0.00)	94.44(0.00)	93.65(5.94)	90.88(0.97)	87.94(0.43)	73.83(2.30)
RL Training	GraphGPO	95.15(1.62)	100.0(0.00)	85.26(2.58)	85.71(5.83)	96.30(2.61)	93.65(2.24)	92.71(1.32)	89.29(1.48)	78.65(3.86)
Qwen2.5-7B-Instruct										
Prompting	Qwen2.5	33.4	21.6	19.3	6.9	2.8	3.2	14.8	26.4	7.8
Prompting	ReAct	48.5	35.4	34.3	13.2	18.2	17.6	31.2	46.2	19.5
Prompting	Reflexion	62.0	41.6	44.9	30.9	36.3	23.8	42.7	58.1	28.8
RL Training	PPO	92.3(4.0)	64.0(8.4)	92.5(2.4)	89.5(7.0)	80.3(2.0)	68.8(8.3)	80.4(2.7)	81.4(3.1)	68.7(5.1)
RL Training	RLOO	87.6(4.3)	78.2(8.3)	87.3(5.8)	81.3(7.6)	71.9(5.2)	48.9(8.4)	75.5(4.6)	80.3(3.2)	65.7(4.0)
RL Training	GRPO	88.98(5.30)	91.98(4.43)	77.89(4.58)	78.57(0.00)	90.74(5.24)	71.43(3.89)	83.33(2.05)	84.31(1.27)	75.00(2.78)
RL Training	GiGPO	97.53(1.75)	100.0(0.00)	83.98(1.32)	90.48(6.73)	94.44(0.00)	100.0(0.00)	94.27(1.33)	86.72(1.44)	78.38(1.94)
RL Training	GraphGPO	100.0(0.00)	100.0(0.00)	91.40(1.50)	92.86(5.83)	94.44(0.00)	92.06(2.24)	95.31(1.10)	86.94(0.68)	80.31(1.33)

Table 2. The test performance of VLM agents using Qwen2.5-VL-3B-Instruct on the interactive game environment Sokoban. We report the average task score and the average success rate (%).

Type	Method	Sokoban [6×6]
Prompting	Qwen2.5-VL	11.7
RL Training	GRPO	67.1
RL Training	GiGPO	76.92
RL Training	GraphGPO	86.98(0.73)

benchmarks. As shown in the table, GraphGPO consistently outperforms all baselines across both benchmarks. In particular, on ALFWorld, GraphGPO achieves improvements on nearly all subtasks, resulting in average success rate gains of 14.85% and 11.98% over GRPO for the 1.5B and 7B models, respectively. On WebShop, GraphGPO not only attains higher task scores but also improves the average success rate over GRPO by 7.30% and 5.31% for the 1.5B and 7B models, respectively. Furthermore, Table 2 reports the comparison results of GraphGPO with group-based RL baselines in the interactive game environment Sokoban. In this deterministic maze-based setting, GraphGPO outperforms GRPO by 19.88% and surpasses GiGPO by 10.06%. These results validate the effectiveness of GraphGPO in improving agent performance.

Training Dynamics. Figure 4 illustrates the evolution of the success rate over training steps on ALFWorld, Web-

Shop, and Sokoban, comparing GraphGPO with GRPO and GiGPO. From Figure 4, we have the following observations: 1) While all three methods exhibit a clear upward trend during training, GraphGPO and GiGPO consistently outperform GRPO across all benchmarks. This suggests that relying solely on coarse-grained episode-level signals is not effective enough for multi-turn agentic tasks. 2) GraphGPO maintains a leading position throughout the training process and achieves the highest peak performance on all benchmarks. This improvement can be attributed to finer-grained credit assignment enabled by graph-based advantage estimation, which leverages global information aggregated from the rollout data. 3) GraphGPO shows a substantially faster improvement in the early stages of training, and the performance gap between GraphGPO and the baselines is most pronounced during the mid-training phase. This behavior indicates that the state-transition graph allows more effective more informative step-level signals, particularly when the overall rollout success rate is low. The training dynamic of validation you can find in Appendix D.2.

Ablation Study. Table 3 presents ablation results comparing GraphGPO with GiGPO. First, we evaluate variants without episode-level advantages ($-A^E$), where only step-level advantages (A^G or A^S) are used. Under this setting, both GraphGPO and GiGPO exhibit performance degradation across all benchmarks. This indicates that relying

Table 3. Ablation study comparing variants without episode-level advantages ($-A^E$) and with dynamic sampling (+DS). The best performance under the same setting is highlighted in bold. For ALFWorld and WebShop, agents use Qwen2.5-1.5B-Instruct as the base LLM. We report the average success rate (%) for all settings.

Method	ALFWorld							WebShop	Sokoban
	Pick	Clean	Cool	Look	Heat	Pick2	All		
GiGPO	98.81 _(1.68)	95.16 _(3.89)	81.46 _(0.56)	78.57 _(0.00)	94.44 _(0.00)	93.65 _(5.94)	90.88 _(0.97)	73.83 _(2.30)	76.92
- A^E	96.34 _(0.06)	100.0 _(0.00)	83.98 _(1.32)	83.33 _(3.36)	98.15 _(2.61)	85.71 _(7.78)	91.41 _(2.30)	73.18 _(2.08)	62.5
+ DS	97.57 _(1.72)	100.0 _(0.00)	90.21 _(4.53)	95.24 _(6.73)	98.00 _(2.23)	98.41 _(2.24)	96.35 _(1.95)	81.25 _(2.78)	-
GrpahGPO	95.15 _(1.62)	100.0 _(0.00)	85.26 _(2.58)	85.71 _(5.83)	96.30 _(2.61)	93.65 _(2.24)	92.71 _(1.32)	78.65 _(3.86)	86.98 _(0.73)
- A^E	97.57 _(1.71)	100.0 _(0.00)	85.22 _(2.78)	85.71 _(0.00)	94.44 _(0.00)	85.71 _(0.00)	91.67 _(0.97)	75.00 _(2.76)	83.07 _(2.41)
+ DS	100.0 _(0.00)	98.41 _(2.24)	95.15 _(3.43)	97.62 _(3.36)	100.0 _(0.00)	100.0 _(0.00)	98.43 _(1.28)	85.68 _(3.52)	90.36 _(1.60)

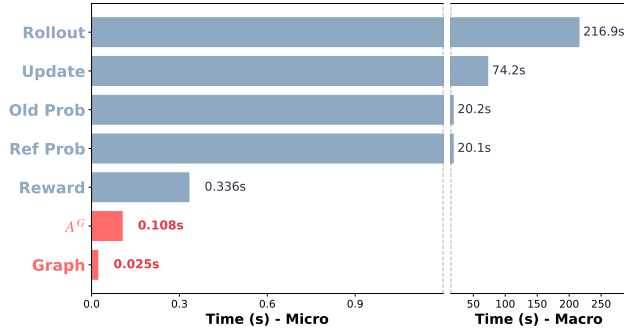


Figure 5. Per-iteration runtime breakdown of training stages, including rollout, graph aggregation, reward estimation, graph-based advantage computation, recomputation of old and reference policy probabilities, and policy update. Gray bars denote stages shared by group-based methods, while red bars indicate the additional overhead introduced by GraphGPO. A broken x-axis is used to accommodate stages with smaller runtimes.

solely on step-level advantages is insufficient to provide informative credit signals for all steps in multi-turn agentic tasks. Nevertheless, GraphGPO consistently outperforms GiGPO across all tasks, achieving a notably larger margin on Sokoban, where it surpasses GiGPO by 20.57%. Second, we compare GraphGPO with GiGPO under dynamic sampling (Yu et al., 2025b). For each example x , a set of rollout trajectories $\{\tau_1, \tau_2, \dots, \tau_M\}$ is resampled if all trajectories in the set are either successful or failed. This resampling process continues until a sufficient number of trajectories is collected or a maximum of 10 attempts is reached. Under this setting, both GraphGPO and GiGPO benefit from dynamic sampling. However, GraphGPO consistently achieves superior performance.

Computational Overhead. For memory overhead, GraphGPO constructs states using only the deterministic components of environment observations (see Appendix for details), without storing additional models or auxiliary datasets. During training, the state-transition graph is maintained using a hash table whose size is $|\mathcal{E}|$, which scales with the rollout parameters M and T and remains

negligible in practice. Regarding per-iteration time overhead, GraphGPO performs a single shortest-path search using Dijkstra algorithm (Haeupler et al., 2024), with a time complexity of $O((|\mathcal{V}| + |\mathcal{E}|) \log |\mathcal{V}|)$. Compared with the computational cost of updating LLMs, this overhead is negligible. The per-iteration runtime breakdown is shown in Figure 5. As illustrated, the most time-consuming stage in training is rollout, which samples trajectories from the policy to guide optimization, followed by policy updates involving backpropagation. In comparison, the additional costs introduced by GraphGPO mainly come from graph construction (0.108s) and graph-based advantage computation (0.025s). These costs are negligible when compared with rollout (216.9s) and policy update (74.2s), accounting for only 0.04% of the total per-iteration runtime. Since rollout is the most time-consuming component, improving RL training efficiency crucially depends on recovering high-fidelity step-level supervision from collected rollouts, especially when successes are sparse.

Additional experimental results, including training dynamics of validation, hyperparameter studies and case study, are provided in Appendix D.

6. Conclusion

In this work, we introduced Graph-based Group Policy Optimization (GraphGPO), a simple and effective method for enabling faithful step-level credit assignment in multi-turn agentic tasks. Unlike existing methods that rely on trajectory-level attribution according to final outcomes, GraphGPO aggregates rollout trajectories into a unified state-transition graph and assigns step-level credit based on global progress toward the task goal. Overall, extensive experiments across diverse multi-turn agentic benchmarks demonstrate that GraphGPO consistently outperforms prior group-based methods by exploiting global structural information across rollouts for step-level credit assignment. The proposed formulation complements existing group-based RL methods and provides a practical mechanism for improving credit assignment in long-horizon agentic tasks.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

This research is supported by the RIE2025 Industry Alignment Fund – Industry Collaboration Projects (IAF-ICP) (Award I2301E0026), administered by A*STAR, as well as supported by Alibaba Group and NTU Singapore through Alibaba-NTU Global e-Sustainability CorpLab (ANGEL).

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.
- Ahmadian, A., Cremer, C., Gallé, M., Fadaee, M., Kreutzer, J., Pietquin, O., Üstün, A., and Hooker, S. Back to basics: Revisiting reinforce style optimization for learning from human feedback in LLMs. *arXiv preprint arXiv:2402.14740*, 2024.
- Ahn, M., Brohan, A., Brown, N., Chebotar, Y., Cortes, O., David, B., Finn, C., Fu, C., Gopalakrishnan, K., Hausman, K., et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.
- Bai, H., Zhou, Y., Pan, J., Cemri, M., Suhr, A., Levine, S., and Kumar, A. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. In *NeurIPS*, 2024.
- Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.
- Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., Das-Sarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.
- Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.
- Cui, G., Zhang, Y., Chen, J., Yuan, L., Wang, Z., Zuo, Y., Li, H., Fan, Y., Chen, H., Chen, W., et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.
- Da, J., Wang, C., Deng, X., Ma, Y., Barhate, N., and Hendryx, S. Agent-RLVR: Training software engineering agents via guidance and environment rewards. *arXiv preprint arXiv:2506.11425*, 2025.
- Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. In *ACL*, 2019.
- Driess, D., Xia, F., Sajjadi, M. S. M., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., Huang, W., Chebotar, Y., Sermanet, P., Duckworth, D., Levine, S., Vanhoucke, V., Hausman, K., Toussaint, M., Greff, K., Zeng, A., Mordatch, I., and Florence, P. Palm-e: an embodied multimodal language model. In *ICML*, 2023.
- Feng, L., Tan, W., Lyu, Z., Zheng, L., Xu, H., Yan, M., Huang, F., and An, B. Towards efficient online tuning of VLM agents via counterfactual soft reinforcement learning. In *ICML*, 2025a.
- Feng, L., Xue, Z., Liu, T., and An, B. Group-in-group policy optimization for LLM agent training. *arXiv preprint arXiv:2505.10978*, 2025b.
- Furuta, H., Lee, K.-H., Nachum, O., Matsuo, Y., Faust, A., Gu, S. S., and Gur, I. Multimodal web navigation with instruction-finetuned foundation models. In *ICLR*, 2024.
- Gemini, Anil, R., Borgeaud, S., Alayrac, J.-B., Yu, J., Soricut, R., Schalkwyk, J., Dai, A. M., Hauth, A., Millican, K., et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.
- Gou, B., Wang, R., Zheng, B., Xie, Y., Chang, C., Shu, Y., Sun, H., and Su, Y. Navigating the digital world as humans do: Universal visual grounding for gui agents. In *ICLR*, 2025.
- Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.
- Haeupler, B., Hladík, R., Rozhoň, V., Tarjan, R. E., and Tetěk, J. Universal optimality of dijkstra via beyond-worst-case heaps. In *FOCS*, 2024.
- Hafner, D., Pasukonis, J., Ba, J., and Lillicrap, T. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

- He, S., Feng, L., Wei, Q., Cheng, X., Feng, L., and An, B. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. *arXiv preprint arXiv:2602.22817*, 2026.
- Jiang, X., Dong, Y., Liu, M., Deng, H., Wang, T., Tao, Y., Cao, R., Li, B., Jin, Z., Jiao, W., et al. Coderl+: Improving code generation via reinforcement with execution semantics alignment. *arXiv preprint arXiv:2510.18471*, 2025.
- Jin, B., Zeng, H., Yue, Z., Yoon, J., Arik, S., Wang, D., Zamani, H., and Han, J. Search-rl: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.
- Konda, V. and Tsitsiklis, J. Actor-critic algorithms. In *NeurIPS*, 1999.
- Kool, W., van Hoof, H., and Welling, M. Buy 4 reinforce samples, get a baseline for free! In *ICLR*, 2019.
- Lai, H., Liu, X., Zhao, Y., Xu, H., Zhang, H., Jing, B., Ren, Y., Yao, S., Dong, Y., and Tang, J. Computerrl: Scaling end-to-end online reinforcement learning for computer use agents. *arXiv preprint arXiv:2508.14040*, 2025.
- Le, H., Wang, Y., Gotmare, A. D., Savarese, S., and Hoi, S. C. H. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. In *NeurIPS*, 2022.
- Le, Q. M., Luu, M. S. K., Tran, K.-T., Nguyen, D.-H., Pham, H.-Q.-V., Le, Q., Lam, H. T., and Nguyen, H. D. Toolbrain: A flexible reinforcement learning framework for agentic tools. *arXiv preprint arXiv:2510.00023*, 2025.
- Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let’s verify step by step. In *ICLR*, 2023.
- Lin, Z., Lin, M., Xie, Y., and Ji, R. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.
- Liu, A., Mei, A., Lin, B., Xue, B., Wang, B., Xu, B., Wu, B., Zhang, B., Lin, C., Dong, C., et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.
- Liu, B., Guertler, L., Yu, S., Liu, Z., Qi, P., Balcells, D., Liu, M., Tan, C., Shi, W., Lin, M., et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. *arXiv preprint arXiv:2506.24119*, 2025b.
- Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding rl-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.
- Lu, Z., Ye, J., Tang, F., Shen, Y., Xu, H., Zheng, Z., Lu, W., Yan, M., Huang, F., Xiao, J., et al. Ui-s1: Advancing gui automation via semi-online reinforcement learning. *arXiv preprint arXiv:2509.11543*, 2025.
- Luo, X., Zhang, Y., He, Z., Wang, Z., Zhao, S., Li, D., Qiu, L. K., and Yang, Y. Agent lightning: Train any ai agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025a.
- Luo, Z., Shen, Z., Yang, W., Zhao, Z., Jwalapuram, P., Saha, A., Sahoo, D., Savarese, S., Xiong, C., and Li, J. Mcp-universe: Benchmarking large language models with real-world model context protocol servers. *arXiv preprint arXiv:2508.14704*, 2025b.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.
- OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.
- Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.
- Paranjape, B., Lundberg, S., Singh, S., Hajishirzi, H., Zettlemoyer, L., and Ribeiro, M. T. Art: Automatic multi-step reasoning and tool-use for large language models. *arXiv preprint arXiv:2303.09014*, 2023.
- Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.
- Qian, C., Acikgoz, E. C., He, Q., Wang, H., Chen, X., Hakkani-Tür, D., Tur, G., and Ji, H. ToolRL: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.
- Qin, Y., Liang, S., Ye, Y., Zhu, K., Yan, L., Lu, Y., Lin, Y., Cong, X., Tang, X., Qian, B., et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. *arXiv preprint arXiv:2307.16789*, 2023.
- Qwen. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.
- Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

- Rafailov, R., Sharma, A., Mitchell, E., Manning, C. D., Ermon, S., and Finn, C. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, pp. 53728–53741, 2023.
- Rawles, C., Li, A., Rodriguez, D., Riva, O., and Lillicrap, T. Androidinthewild: A large-scale dataset for android device control. In *NeurIPS*, 2023.
- Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Hambro, E., Zettlemoyer, L., Cancedda, N., and Scialom, T. Toolformer: Language models can teach themselves to use tools. In *NeurIPS*, 2023.
- Schrader, M.-P. B. gym-sokoban. <https://github.com/mpSchrader/gym-sokoban>, 2018.
- Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. Trust region policy optimization. In *ICML*, 2015a.
- Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.
- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y., Wu, Y., et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Shi, Y., Yu, W., Li, Z., Wang, Y., Zhang, H., Liu, N., Mi, H., and Yu, D. Mobilegui-rl: Advancing mobile gui agent through reinforcement learning in online environment. *arXiv preprint arXiv:2507.05720*, 2025.
- Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023.
- Shridhar, M., Yuan, X., Côté, M.-A., Bisk, Y., Trischler, A., and Hausknecht, M. Alfworld: Aligning text and embodied environments for interactive learning. In *ICLR*, 2021.
- Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018.
- Song, H., Jiang, J., Min, Y., Chen, J., Chen, Z., Zhao, W. X., Fang, L., and Wen, J.-R. R1-searcher: Incentivizing the search capability in llms via reinforcement learning. *arXiv preprint arXiv:2503.05592*, 2025.
- Stiennon, N., Ouyang, L., Wu, J., Ziegler, D., Lowe, R., Voss, C., Radford, A., Amodei, D., and Christiano, P. F. Learning to summarize with human feedback. In *NeurIPS*, 2020.
- Su, Y., Yu, D., Song, L., Li, J., Mi, H., Tu, Z., Zhang, M., and Yu, D. Crossing the reward bridge: Expanding rl with verifiable rewards across diverse domains. *arXiv preprint arXiv:2503.23829*, 2025.
- Sun, H., Qiao, Z., Guo, J., Fan, X., Hou, Y., Jiang, Y., Xie, P., Zhang, Y., Huang, F., and Zhou, J. Zeroscore: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.
- Sun, Z., Lyu, C., Li, B., Wan, Y., Zhang, H., Li, G., and Jin, Z. Enhancing code generation performance of smaller models by distilling the reasoning ability of llms. In *LREC/COLING*, 2024.
- Sutton, R. S., Barto, A. G., et al. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.
- Tan, W., Zhang, W., Liu, S., Zheng, L., Wang, X., and An, B. True knowledge comes from practice: Aligning llms with embodied environments via reinforcement learning. *arXiv preprint arXiv:2401.14151*, 2024.
- Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.
- Team, M. F. M. and Team, M. A. I. Mirorl: An mcp-first reinforcement learning framework for deep research agent. <https://github.com/MiroMindAI/MiroRL>, 2025.
- Wang, G., Xie, Y., Jiang, Y., Mandlekar, A., Xiao, C., Zhu, Y., Fan, L., and Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.
- Wang, H., Qian, C., Zhong, W., Chen, X., Qiu, J., Huang, S., Jin, B., Wang, M., Wong, K.-F., and Ji, H. Acting less is reasoning more! teaching model to act efficiently. *arXiv preprint arXiv:2504.14870*, 2025a.
- Wang, H., Zou, H., Song, H., Feng, J., Fang, J., Lu, J., Liu, L., Luo, Q., Liang, S., Huang, S., et al. Ui-tars-2 technical report: Advancing gui agent with multi-turn reinforcement learning. *arXiv preprint arXiv:2509.02544*, 2025b.
- Wang, J., Xu, H., Jia, H., Zhang, X., Yan, M., Shen, W., Zhang, J., Huang, F., and Sang, J. Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration. In *NeurIPS*, 2024a.

- Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., et al. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345, 2024b.
- Wang, T., Wu, Z., Liu, J., Hao, J., Wang, J., and Shao, K. Distrl: An asynchronous distributed reinforcement learning framework for on-device control agents. *arXiv preprint arXiv:2410.14803*, 2024c.
- Wang, W., Xiong, S., Chen, G., Gao, W., Guo, S., He, Y., Huang, J., Liu, J., Li, Z., Li, X., et al. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*, 2025c.
- Wang, Z., Schaul, T., Hessel, M., Hasselt, H., Lanctot, M., and Freitas, N. Dueling network architectures for deep reinforcement learning. In *ICML*, 2016.
- Wang, Z., Li, X., Ye, Y., Fang, J., Wang, H., Liu, L., Liang, S., Lu, J., Wu, Z., Feng, J., et al. Game-tars: Pretrained foundation models for scalable generalist multimodal game agents. *arXiv preprint arXiv:2510.23691*, 2025d.
- Wang, Z., Wang, K., Wang, Q., Zhang, P., Li, L., Yang, Z., Jin, X., Yu, K., Nguyen, M. N., Liu, L., et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025e.
- Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.
- Wen, M., Wan, Z., Wang, J., Zhang, W., and Wen, Y. Reinforcing llm agents via policy optimization with action decomposition. In *NeurIPS*, 2024.
- Wen, X., Liu, Z., Zheng, S., Ye, S., Wu, Z., Wang, Y., Xu, Z., Liang, X., Li, J., Miao, Z., et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.
- Xi, Z., Chen, W., Guo, X., He, W., Ding, Y., Hong, B., Zhang, M., Wang, J., Jin, S., Zhou, E., et al. The rise and potential of large language model based agents: A survey. *Science China Information Sciences*, 68(2):121101, 2025.
- Xu, Z., Yu, C., Fang, F., Wang, Y., and Wu, Y. Language agents with reinforcement learning for strategic play in the werewolf game. In *ICML*, 2024.
- Xue, Z., Zheng, L., Liu, Q., Li, Y., Zheng, X., Ma, Z., and An, B. Simpletir: End-to-end reinforcement learning for multi-turn tool-integrated reasoning. *arXiv preprint arXiv:2509.02479*, 2025.
- Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., and Press, O. Swe-agent: Agent-computer interfaces enable automated software engineering. In *NeurIPS*, 2024.
- Yao, S., Chen, H., Yang, J., and Narasimhan, K. Web-shop: Towards scalable real-world web interaction with grounded language agents. In *NeurIPS*, 2022a.
- Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K. R., and Cao, Y. React: Synergizing reasoning and acting in language models. In *ICLR*, 2022b.
- Yu, H., Chen, T., Feng, J., Chen, J., Dai, W., Yu, Q., Zhang, Y.-Q., Ma, W.-Y., Liu, J., Wang, M., et al. Memagent: Reshaping long-context llm with multi-conv rl-based memory agent. *arXiv preprint arXiv:2507.02259*, 2025a.
- Yu, Q., Zhang, Z., Zhu, R., Yuan, Y., Zuo, X., Yue, Y., Dai, W., Fan, T., Liu, G., Liu, L., et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025b.
- Zhai, S., Bai, H., Lin, Z., Pan, J., Tong, P., Zhou, Y., Suhr, A., Xie, S., LeCun, Y., Ma, Y., et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In *NeurIPS*, 2024.
- Zhang, K., Zuo, Y., He, B., Sun, Y., Liu, R., Jiang, C., Fan, Y., Tian, K., Jia, G., Li, P., et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025.
- Zheng, B., Gou, B., Kil, J., Sun, H., and Su, Y. Gpt-4v (ision) is a generalist web agent, if grounded. *arXiv preprint arXiv:2401.01614*, 2024a.
- Zheng, C., Liu, S., Li, M., Chen, X.-H., Yu, B., Gao, C., Dang, K., Liu, Y., Men, R., Yang, A., et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025a.
- Zheng, T., Zhang, G., Shen, T., Liu, X., Lin, B. Y., Fu, J., Chen, W., and Yue, X. Opencodeinterpreter: Integrating code generation with execution and refinement. *arXiv preprint arXiv:2402.14658*, 2024b.
- Zheng, Y., Fu, D., Hu, X., Cai, X., Ye, L., Lu, P., and Liu, P. Deepresearcher: Scaling deep research via reinforcement learning in real-world environments. *arXiv preprint arXiv:2504.03160*, 2025b.
- Ziegler, D. M., Stiennon, N., Wu, J., Brown, T. B., Radford, A., Amodei, D., Christiano, P., and Irving, G. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A. Pseudo Code.

Algorithm 1 The pseudo-code of GraphGPO

```

1: Require: Initial policy  $\pi_{\theta_{\text{old}}}$ , task distribution  $p(X)$ , distance discount factor  $\omega$ , clipping parameter  $\epsilon$ , KL penalty  $\beta$ ,
   group size  $M$ , maximum step  $T$ 
2: for each training iteration do
3:   Update the old policy model:  $\theta_{\text{old}} \leftarrow \theta$ 
4:   // Multi-step rollout phase
5:   Sample task  $\mathbf{x} \sim p(X)$  and initialize  $M$  identical environments
6:   for  $t = 1$  to  $T$  do
7:     Sample actions  $\{\mathbf{a}_t^m \sim \pi_{\theta_{\text{old}}}(\cdot | \mathbf{s}_t^m, \mathbf{x})\}_{m=1}^M$ 
8:     Execute actions, observe the next state  $\{\mathbf{s}_{t+1}^m\}_{m=1}^M$ 
9:   end for
10:  // Graph aggregation phase
11:  Construct directed graphs  $\mathcal{G} = (\mathcal{S}, \mathcal{E})$  by Section (4.2)
12:  Compute distance  $d(s)$  for each nodes  $s \in \mathcal{S}$  by Eq. (3)
13:  // Grouping phase
14:  Build step-level groups  $G^G(s)$  by Eq. (5)
15:  // Advantage computation phase
16:  Compute episode-level advantages by Eq. (2)
17:  Compute graph-based advantages within each group by Eq. (6)
18:  // Policy update phase
19:  Update policy  $\theta$  by maximizing objective Eq. (9)
20: end for
    
```

Algorithm 1 summarizes the full training procedure. We highlight the key stages of the training pipeline, with the runtime of each stage reported in Figure 5. As such, GraphGPO preserves the critic-free, low-memory, and stable convergence properties, as well as the high efficiency of group-based RL, while aggregating all rollout trajectories into a global structure to capture latent relationships across steps, thereby enabling more fine-grained and informative credit assignment.

B. Proofs of Theorem

B.1. Proof of Proposition 4.1

By definition of the graph-based reward, we have graph-based step-level rewards for edge $(s, \mathbf{a}_{\text{good}}, s'_{\text{good}})$ and $(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}})$ as follows:

$$R^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) = r_{\text{succ}} \omega^{d(s'_{\text{good}}) + c(s, \mathbf{a}_{\text{good}})}, \quad (10)$$

$$R^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}}) = r_{\text{succ}} \omega^{d(s'_{\text{bad}}) + c(s, \mathbf{a}_{\text{bad}})}. \quad (11)$$

where $r_{\text{succ}} > 0$ is a scalar. let $D_{\text{good}} = d(s'_{\text{good}}) + c(s, \mathbf{a}_{\text{good}})$ and $D_{\text{bad}} = d(s'_{\text{bad}}) + c(s, \mathbf{a}_{\text{bad}})$. By the assumption that $\mathbf{a}_{\text{good}}$ moves closer to the goal than $\mathbf{a}_{\text{bad}}$, we have:

$$D_{\text{good}} < D_{\text{bad}}. \quad (12)$$

Since $\omega \in (0, 1)$, the function $x \rightarrow \omega^x$ is strictly decreasing on $\mathcal{R}$, so we have:

$$D_{\text{good}} < D_{\text{bad}} \implies \omega^{D_{\text{good}}} > \omega^{D_{\text{bad}}} \implies R^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) > R^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}}). \quad (13)$$

Then we recall the graph-based advantages $A^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) = \frac{R^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) - \mu(G^G(s))}{\sigma(G^G(s))}$ and $A^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}}) = \frac{R^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}}) - \mu(G^G(s))}{\sigma(G^G(s))}$ where $\mu(G^G(s))$ and $\sigma(G^G(s))$ are the same for all actions at state s . Since there are at least the two edges considered above and their rewards are different, we have $\sigma(G^G(s)) > 0$. Hence,

$$A^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) - A^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}}) = \frac{R^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) - \mu(G^G(s))}{\sigma(G^G(s))} - \frac{R^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}}) - \mu(G^G(s))}{\sigma(G^G(s))} \quad (14)$$

$$= \frac{R^G(s, \mathbf{a}_{\text{good}}, s'_{\text{good}}) - R^G(s, \mathbf{a}_{\text{bad}}, s'_{\text{bad}})}{\sigma(G^G(s))} \quad (15)$$

$$> 0 \quad (16)$$

Which concludes the proof of Proposition 4.1. $\square$

B.2. Proof of Proposition 4.2

We analyze the conditional variance in each step-level feedback.

For a state-action pair $(s, \mathbf{a}, s')$ that is visited with non-zero probability, the trajectory-style step feedback at time t as follows:

$$X^S = \eta(t)R(\tau), \quad (17)$$

where $\eta(t) = 1$ for GRPO and $\eta(t) = \lambda^{T-t}$ for GRPO. The graph-style step feedback at time t as follows:

$$X^G = R^G(s, \mathbf{a}, s'), \quad (18)$$

Suppose positive probability of both successful and failed trajectories passing through $(s, \mathbf{a}, s')$. Denote by $\tau_{succ}(s, \mathbf{a}, s')$ and $\tau_{fail}(s, \mathbf{a}, s')$ the sets of successful and failed trajectories that visit $(s, \mathbf{a}, s')$ with positive probability, i.e.,

$$p_{succ} = \mathbb{P}(\tau \in \tau_{succ}(s, \mathbf{a})), \quad (19)$$

$$p_{fail} = 1 - p_{succ} > 0. \quad (20)$$

Suppose $(s, \mathbf{a}, s')$ is independent with step t , that is, $p(s, \mathbf{a}, s'|t) = p(s, \mathbf{a}, s')$. Then, for a $\tau \in \tau_{succ}(s, \mathbf{a}, s')$ we have $R(\tau) > 0$, so:

$$X^S = \eta(t)R(\tau) > 0. \quad (21)$$

It is worth noting that X^S is not independent with step t . X^S is different when $(s, \mathbf{a}, s')$ observed in different t , that is:

$$\text{Var}(X^S|s, \mathbf{a}, s', \tau_{succ}) \geq 0. \quad (22)$$

For a fail trajectory $\tau \in \tau_{fail}(s, \mathbf{a}, s')$, the reward $R(\tau) = 0$, so $X^S = 0$ and $\text{Var}(X^S|s, \mathbf{a}, s', \tau_{fail}) = 0$. Combining the above Eq. 22, we have the final conditional variance for the trajectory-style step feedback:

$$\text{Var}(X^S|s, \mathbf{a}, s') \geq 0. \quad (23)$$

Now consider the graph-style step feedback. The empirical state-transition graph and distance function $d(\cdot)$ are constructed from the current batch of rollouts. Once the graph is fixed, the reward assigned to any observed transition $(s, \mathbf{a}, s')$ is a deterministic function $R^G(s, \mathbf{a}, s')$. Since the environment is deterministic, the empirical graph is deterministic. The successor state s' and thus the feedback X^G are deterministic. Therefore

$$\text{Var}(X^G|s, \mathbf{a}, s') = 0. \quad (24)$$

Combining the Eq. 23, we have:

$$\text{Var}(X^G|s, \mathbf{a}, s') \leq \text{Var}(X^S|s, \mathbf{a}, s'). \quad (25)$$

Which concludes the proof of Proposition 4.2. $\square$

C. Implementation Details

C.1. Computing Details

All experiments are conducted for 150 training epochs across all benchmarks and model configurations. For experiments using Qwen2.5-1.5B-Instruct and Qwen2.5-VL-3B-Instruct, training is performed on 2 NVIDIA H100 GPUs with 80 GB memory each. For experiments using the larger Qwen2.5-7B-Instruct model, we employ 4 NVIDIA H100 GPUs with 80 GB memory each to accommodate the increased computational and memory requirements.

Prompt Template for ALFWorld

You are an expert agent operating in the ALFRED embodied 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 of the current situation are: `[{admissible_actions}]`. Location: `{current_location}`. Items in hand (status): `{current_holding}`. History moving list by you: `{history_moving}`. Now it's your turn to take an action. You should first reason step-by-step about the current situation. 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 6. Prompt template used for ALFWorld experiments.

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: `[{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.

Figure 7. Prompt template used for WebShop experiments.

C.2. Prompts

The prompt templates used for agents in ALFWorld, WebShop, and Sokoban are shown in Figure 6, Figure 7, and Figure 8, respectively. All prompts are constructed using Python-style string formatting, where placeholders enclosed in curly braces (`{}`) indicate semantic slots that are dynamically instantiated at each interaction step.

Specifically, `{task_description}` specifies the task definition, and `{step_count}` denotes the number of actions already executed. The placeholder `{history_length}` indicates the length of the visible interaction history, which is set to 2 in all experiments. The agent's recent action-observation history is represented by `{action_history}`. The current interaction step is denoted by `{current_step}`, while `{current_observation}` corresponds to the observation returned by the environment at the current step. The placeholder `{admissible_actions}` enumerates the set of valid actions available to the agent under the current observation.

In addition, several environment-specific placeholders are introduced to provide richer state information in ALFWorld. `{current_location}` indicates the agent's current location, `{current_holding}` describes the object(s) currently held by the agent along with their states, and `{history_moving}` records the agent's object manipulation history. When no object movement has occurred, the `{history_moving}` field is omitted from the prompt.

To explicitly structure the model's reasoning and outputs, we employ a set of control tags. The step-by-step reasoning process is enclosed within `<think>` and `</think>` tags, while the final selected action is wrapped within `<action>` and `</action>` tags. For vision-language settings, `<image>` serves as a placeholder token representing the visual observation.

C.3. Comparing Methods

GPT-4o. A closed-source, large-scale large language model used as a strong baseline for multi-turn agentic tasks, demonstrating advanced reasoning and instruction-following capabilities.

Gemini-2.5-Pro. A closed-source large language model comparable in scale and overall capability to GPT-4o, serving as another competitive proprietary baseline.

ReAct. A prompting-based agent framework that interleaves reasoning and acting by explicitly generating intermediate thoughts and actions in a chain-of-thought manner.

Prompt Template for Sokoban

You are an expert agent operating in the Sokoban environment. Your goal is to push all the boxes onto the target spots. Once all boxes are on the targets, you win!

Rules
 You can only push boxes. You can't pull them, so plan ahead to avoid getting stuck.
 You can't walk through or push boxes into walls.
 To avoid traps, do not push boxes into corners or against walls where they can't be moved again.

Visual Elements in the Image:
 Character: A small, green alien-like figure with two antennae and black eyes. It represents you.
 Box: A yellow crate marked with an orange "X" across its front. It is the box you need to push.
 Target: A black tile outlined in red, with a small red diamond shape in the center. It marks the destination where a box should be pushed.

Current Step
 Your current observation is shown in the image: `<image>`
 Your admissible actions are ["up", "down", "left", "right"].
 Now it's your turn to make a move (choose ONE action only for the current step).
 You should first reason step-by-step about the current situation — observe the positions of boxes and targets, plan a path to push a box toward a target, and avoid traps like corners or walls. 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 8. Prompt template used for Sokoban experiments.

Reflexion. A prompting-based agent that enhances performance through self-reflection and iterative refinement over previously generated trajectories.

PPO. Proximal Policy Optimization, a widely used actor–critic reinforcement learning algorithm that relies on a learned value function for stable policy updates.

RLOO. Reinforcement Learning with Offline Observations, a group-based reinforcement learning approach that estimates advantages using group-level statistics without training an explicit value network.

GRPO. Group Relative Policy Optimization, a group-based reinforcement learning method that performs trajectory-level advantage estimation and is designed to scale RL training to multi-step and reasoning-intensive tasks.

GiGPO. Grouped Incremental Group Policy Optimization, a hierarchical group-based reinforcement learning method that performs groupwise step-level advantage estimation for LLM-based agents.

For the main group-based baselines (GRPO and GiGPO), we reproduce their experimental results under the same settings. For other comparison methods, we directly adopt the reported results from GiGPO.

C.4. ALFWorld

All methods are configured with identical hyperparameters for fair comparison. The maximum prompt length is set to 2048 tokens, and the maximum response length is 512 tokens. Each episode allows up to 50 environment steps. The learning rate is set to 1×10^{-6} for the actor and 1×10^{-5} for the critic, where the critic is used only in PPO. We adopt a rule-based reward scheme, assigning a reward of 10 for successful task completion and 0 otherwise. To handle invalid actions generated by the agent, a reward penalty of -0.1 is applied.

For all group-based RL methods, we use a group size of 8 and sample 16 groups per rollout, resulting in a total of $16 \times 8 = 128$ environments. In contrast, PPO uses 128 independent environments for rollouts. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 256, and the KL-divergence loss coefficient is set to 0.01. For GiGPO, the discount factor λ is set to 0.95.

It is worth noting that we identify incomplete observations and issues in the default ALFWorld environment. To address this, we augment the observations with additional information to improve environment fidelity. Specifically, for each environment step, we add only the agent's current location and the objects it is holding. Detailed descriptions of these modifications are provided in Appendix E. For all group-based RL methods, including GRPO and GiGPO, we reproduce their experimental results under this improved environment. Compared to the original setting, these methods consistently achieve noticeably

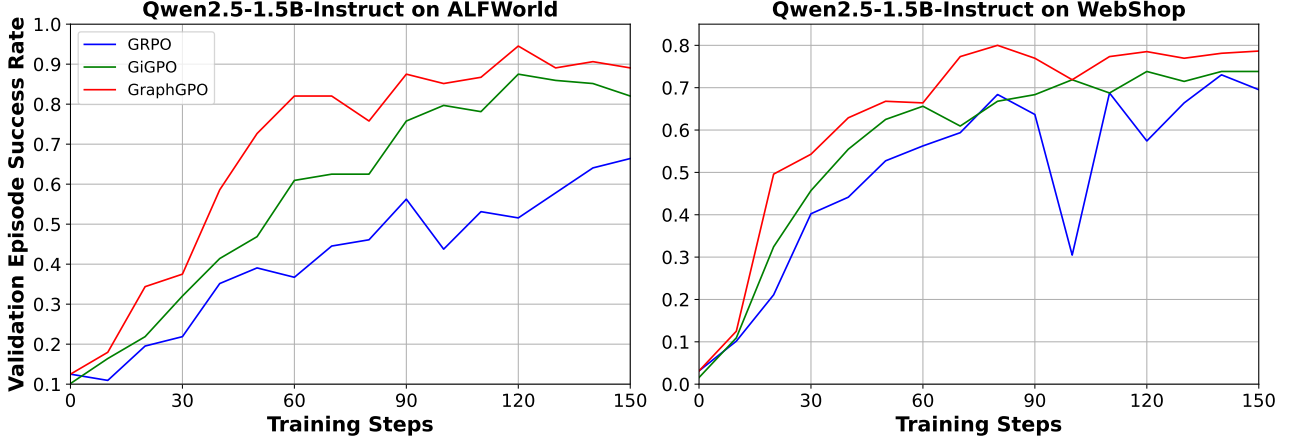


Figure 9. Validation episode success rate versus training steps for GraphGPO (red), GiGPO (green), and GRPO (blue) on the ALFWorld and WebShop benchmarks. The success rate is recorded every 10 training steps.

better performance. Finally, the distance discount factor ω in GraphGPO is set to 0.10.

C.5. WebShop

All methods are configured with identical hyperparameters to ensure fair comparison. The maximum prompt length is set to 5120 tokens, and the maximum response length is 512 tokens. Each episode is limited to 15 environment steps. The learning rate is 1×10^{-6} for the actor and 1×10^{-5} for the critic, where the critic is used only in PPO. We adopt a rule-based reward scheme, assigning a reward of 10 for successful task completion and 0 otherwise. Invalid actions are penalized with a reward of -0.1 .

As with ALFWorld, all group-based RL methods use a group size of 8 and sample 16 groups per rollout, resulting in a total of $16 \times 8 = 128$ environments. In contrast, PPO uses 128 independent environments for rollouts. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 64, and the KL-divergence loss coefficient is set to 0.01. For GiGPO, the discount factor λ is set to 0.95.

It is worth noting that we adopt a more realistic variant of the environment, namely the official *text-rich* version, which provides additional information on interactive elements such as buttons. For all group-based RL methods, including GRPO and GiGPO, we reproduce their experimental results under this enhanced environment. Compared to the original setting, these methods consistently achieve noticeably better performance. Finally, the distance discount factor ω in GraphGPO is set to 0.20.

C.6. Sokoban

We evaluate GraphGPO in a vision-based interactive game environment using the Sokoban benchmark. All methods are configured with identical hyperparameters to ensure fair comparison. We use Qwen2.5-VL-3B-Instruct as the base vision language model. The maximum prompt length is set to 1024 tokens, and the maximum response length is 512 tokens. Each episode is limited to 15 environment steps.

The learning rate for the actor is set to 1×10^{-6} . We adopt a rule-based reward scheme, assigning a reward of 10 for successful task completion and 0 otherwise. Invalid actions generated by the agent are penalized with a reward of -0.1 . The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 64, and the KL-divergence loss coefficient is set to 0.01.

For group-based reinforcement learning methods, including GRPO and GiGPO, we use a group size of 8, corresponding to 8 parallel rollouts per state. A total of 128 environments are used during training. For GiGPO, the discount factor λ is set to 0.95. The distance discount factor ω in GraphGPO is set to 0.80.

Table 4. The test performance on AlfWorld with distance discount ω changes using the Qwen2.5-1.5B-Instruct. We report the average success rate (%). All results are averaged over 3 random seeds during testing. The best performances are highlighted in bold.

Task	$\omega = 0.1$	$\omega = 0.2$	$\omega = 0.4$	$\omega = 0.6$	$\omega = 0.8$
Pick	95.15 _(1.62)	96.34 _(0.06)	96.34 _(0.06)	92.68 _(0.12)	97.57 _(1.72)
Clean	100.0 _(0.00)	100.0 _(0.00)	100.0 _(0.00)	100.0 _(0.00)	100.0 _(0.00)
Cool	85.26 _(2.58)	83.98 _(1.32)	82.70 _(1.84)	83.98 _(1.32)	83.98 _(1.32)
Look	85.71 _(5.83)	85.71 _(0.00)	85.71 _(0.00)	80.95 _(3.36)	85.71 _(0.00)
Heat	96.30 _(2.61)	92.59 _(2.61)	100.0 _(0.00)	94.44 _(0.00)	94.44 _(0.00)
Pick2	93.65 _(2.24)	92.06 _(4.48)	93.65 _(2.24)	88.89 _(2.24)	90.48 _(3.89)
All	92.71 _(1.32)	91.93 _(0.97)	92.97 _(0.63)	90.36 _(0.36)	92.19 _(0.64)

Table 5. The test performance on WebShop with distance discount ω changes using the Qwen2.5-1.5B-Instruct. We report the average success rate (%). All results are averaged over 3 random seeds during testing. The best performances are highlighted in bold.

Metric	$\omega = 0.2$	$\omega = 0.4$	$\omega = 0.6$	$\omega = 0.8$	$\omega = 0.95$
Succ.	78.65 _(3.86)	75.78 _(3.76)	78.00 _(2.41)	75.00 _(2.23)	75.91 _(4.35)

D. Additional Results

D.1. Hyperparameter

GraphGPO introduces only one method-specific hyperparameter, the distance discount factor ω , while all other hyperparameters follow standard group-based RL settings. Table 4 reports the average success rate on ALFWORLD under different values of ω , using Qwen2.5-1.5B-Instruct as the base model. As shown in the table, although varying ω has some influence on performance, the overall sensitivity is relatively mild. Even in the worst-case setting, GraphGPO achieves performance comparable to the baseline, while in most cases it consistently outperforms the baseline.

A similar trend is observed on WebShop. Table 5 presents the average success rate on WebShop under different values of the distance discount factor ω , again using Qwen2.5-1.5B-Instruct as the base model. Overall, these results indicate that GraphGPO is robust to the choice of ω . In practice, we recommend using a relatively small value of ω , such as in the range of 0.1 to 0.4.

D.2. Training Dynamics

Figure 4 illustrates the evolution of the training episode success rate over training steps. As shown in Figure 4, GraphGPO consistently outperforms GRPO and GiGPO throughout the entire training process, with particularly pronounced advantages in the early and mid training stages. This behavior can be attributed to the more accurate and informative step-level credit assignment provided by GraphGPO, which enables the policy to receive meaningful learning signals even when overall trajectory success is sparse.

Figure 9 reports the evolution of the validation episode success rate over training steps on ALFWORLD and WebShop. Similar trends to the training curves can be clearly observed: GraphGPO maintains superior performance compared to GRPO and GiGPO across the whole training horizon. Moreover, GraphGPO exhibits faster convergence and higher training efficiency. On ALFWORLD, GraphGPO reaches comparable validation performance approximately 30 training steps earlier than GiGPO and about 110 steps earlier than GRPO. On WebShop, GraphGPO achieves similar performance roughly 80 steps earlier than GiGPO and 90 steps earlier than GRPO.

Overall, these results indicate that GraphGPO is able to more effectively exploit the information contained in rollout trajectories by leveraging global structural relationships across states. By providing faithful and low-variance step-level supervision beyond trajectory-level outcomes, GraphGPO accelerates policy improvement and leads to faster and more stable convergence in long-horizon multi-turn agentic tasks.

D.3. Noise Environments

In high-dimensional or long-text observations, minor variations in formatting or environment responses can indeed lead to a fragmented graph. Considering an extreme case where no states can be matched, GraphGPO relies only on the trajectory-level

Table 6. The test performance under different noise settings on WebShop. We compare raw-text matching and embedding-based matching, where p denotes the probability of inserting advertisements into a webpage, and q denotes the number of inserted advertisements. We report the average success rate (%). All results are averaged over 3 random seeds during testing. The best performances are highlighted in bold.

Noise setting	Raw-text matching	Embedding-based matching
$(q = 1, p = 0.3)$	77.85 _(3.15)	78.26 _(2.58)
$(q = 1, p = 0.6)$	75.27 _(2.15)	77.42 _(2.38)
$(q = 2, p = 0.3)$	76.85 _(3.60)	78.22 _(1.30)
$(q = 2, p = 0.6)$	74.22 _(2.60)	78.26 _(2.58)

Table 7. The test performance under different group sizes on WebShop. We compare GRPO, GiGPO, and GraphGPO. We report the average success rate (%). All results are averaged over 3 random seeds during testing. The best performances are highlighted in bold.

Group size	GRPO	GiGPO	GraphGPO
$n = 4$	65.23 _(1.99)	69.90 _(2.19)	74.06 _(2.88)
$n = 6$	66.41 _(2.78)	73.34 _(1.98)	74.58 _(3.00)
$n = 8$	71.35 _(2.05)	73.83 _(2.30)	78.65 _(3.86)

advantage and degenerates to GRPO. Therefore, GRPO can be viewed as a lower bound of GraphGPO in the extreme case.

To further examine the applicability of GraphGPO in such settings, we constructed a more challenging version of WebShop by inserting random advertisements generated by Claude. These injected advertisements act as random noise, making exact raw-text matching much more difficult and resulting in a highly sparse graph. To address this issue, we additionally used embedding-based state matching: two states are treated as identical when their embedding similarity exceeds 95%. Table 6 reports experimental results. With embedding-based matching, GraphGPO remains highly robust even under severe graph sparsity and substantial observation noise.

D.4. Group Sizes

In general, more exploration can better reflect the environment structure, leading to more accurate credit assignment and potentially better performance. However, GraphGPO does not require a complete state-transition graph in order to work effectively. Constructing a fully explored graph would indeed be computationally expensive, especially since rollout is the most time-consuming part of each training iteration, as shown in Figure 5. To examine this trade-off, we provide additional results under different numbers of rollouts in Table 7.

E. Case Study

Follow, we present a complete case study of an agent trained by GroupGPO on ALFWorld, including the prompt and the response at each step. For each prompt, the components used to define the state in GroupGPO are highlighted in yellow.

Prompt (ALFWorld, Step 1)

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

Your current observation is: -= Welcome to TextWorld, ALFRED! -=You are in the middle of a room. Looking quickly around you, you see a countertop 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, a towelholder 2, and a towelholder 1.

Your task is to: clean some cloth and put it in countertop. Location: middle of a room. Items in hand (status): nothing.

Your admissible actions of the current situation are: [go to countertop 1, go to drawer 1, go to drawer 2, go to drawer 3, go to drawer 4, go to drawer 5, go to drawer 6, go to garbagecan 1, go to handtowelholder 1, go to handtowelholder 2, go to sinkbasin 1, go to toilet 1, go to toiletpaperhanger 1, go to towelholder 1, go to towelholder 2, help, inventory, look].

Now it's your turn to take an action. You should first reason step-by-step about the current situation. 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.

Response (ALFWorld, Step 1)

<think> The task is to clean some cloth and put it on the countertop. We need to consider the actions we can take to achieve this.

1. First, we need to find a cloth.
2. Then, we can clean the cloth using the provided supplies.
3. Finally, we should put the cleaned cloth on the countertop. </think>

<action>go to countertop 1</action>

Prompt (ALFWorld, Step 2)

You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: clean some cloth and put it in countertop.
Prior to this step, you have already taken 1 step(s). Below are the most recent 1 observaitons and the corresponding actions you took: [Observation 1: '-= Welcome to TextWorld, ALFRED! -=You are in the middle of a room. Looking quickly around you, you see a countertop 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, a towelholder 2, and a towelholder 1.
Your task is to: clean some cloth and put it in countertop. Location: middle of a room. Items in hand (status): nothing.', Action 1: 'go to countertop 1']
You are now at step 2 and your current observation is: You arrive at countertop 1. On the countertop 1, you see a cloth 2, and a sink 1. Location: countertop 1. Items in hand (status): nothing.
Your admissible actions of the current situation are: [examine countertop 1, go to drawer 1, go to drawer 2, go to drawer 3, go to drawer 4, go to drawer 5, go to drawer 6, go to garbagecan 1, go to handtowelholder 1, go to handtowelholder 2, go to sinkbasin 1, go to toilet 1, go to toiletpaperhanger 1, go to towelholder 1, go to towelholder 2, help, inventory, look, take cloth 2 from countertop 1].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (ALFWorld, Step 2)

<think>The current situation is that you are at the countertop 1, where you see a cloth 2 and a sink 1. You have already decided to clean the cloth 2.</think>
<action>take cloth 2 from countertop 1</action>

Prompt (ALFWorld, Step 3)

You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: clean some cloth and put it in countertop.
Prior to this step, you have already taken 2 step(s). Below are the most recent 2 observaitons and the corresponding actions you took: [Observation 1: '-= Welcome to TextWorld, ALFRED! -=You are in the middle of a room. Looking quickly around you, you see a countertop 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a handtowelholder 2, a handtowelholder 1, a sinkbasin 1, a toilet 1, a toiletpaperhanger 1, a towelholder 2, and a towelholder 1.
Your task is to: clean some cloth and put it in countertop. Location: middle of a room. Items in hand (status): nothing.', Action 1: 'go to countertop 1']
[Observation 2: 'You arrive at countertop 1. On the countertop 1, you see a cloth 2, and a sink 1. Location: countertop 1. Items in hand (status): nothing.', Action 2: 'take cloth 2 from countertop 1']
You are now at step 3 and your current observation is: You pick up the cloth 2 from the countertop 1. Location: countertop 1. Items in hand (status): cloth 2(unprocessed).
Your admissible actions of the current situation are: [examine cloth 2, examine countertop 1, go to drawer 1, go to drawer 2, go to drawer 3, go to drawer 4, go to drawer 5, go to drawer 6, go to garbagecan 1, go to handtowelholder 1, go to handtowelholder 2, go to sinkbasin 1, go to toilet 1, go to toiletpaperhanger 1, go to towelholder 1, go to towelholder 2, help, inventory, look, move cloth 2 to countertop 1].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (ALFWorld, Step 3)

<think>The cloth is currently picked up from the countertop 1. The cloth currently is not yet cleaned. We need to clean the cloth before placing it in the countertop.</think>
<action>go to sinkbasin 1</action>

Prompt (ALFWorld, Step 4)

You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: clean some cloth and put it in countertop.
Prior to this step, you have already taken 3 step(s). Below are the most recent 2 observaitons and the corresponding actions you took: [Observation 2: 'You arrive at countertop 1. On the countertop 1, you see a cloth 2, and a sink 1. Location: countertop 1. Items in hand (status): nothing.', Action 2: 'take cloth 2 from countertop 1']
[Observation 3: 'You pick up the cloth 2 from the countertop 1. Location: countertop 1. Items in hand (status): cloth 2(unprocessed).', Action 3: 'go to sinkbasin 1']
You are now at step 4 and your current observation is: You arrive at sinkbasin 1. On the sinkbasin 1, you see a sink 1, and a soapbar 2. Location: sinkbasin 1. Items in hand (status): cloth 2(unprocessed).
Your admissible actions of the current situation are: [clean cloth 2 with sinkbasin 1, examine cloth 2, examine sinkbasin 1, go to countertop 1, go to drawer 1, go to drawer 2, go to drawer 3, go to drawer 4, go to drawer 5, go to drawer 6, go to garbagecan 1, go to handtowelholder 1, go to handtowelholder 2, go to toilet 1, go to toiletpaperhanger 1, go to towelholder 1, go to towelholder 2, help, inventory, look, move cloth 2 to sinkbasin 1].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (ALFWorld, Step 4)

<think>The user is at the sinkbasin 1 and has a cloth 2 in their hand. They need to clean the cloth and put it back on the countertop 1.</think>
<action>clean cloth 2 with sinkbasin 1</action>

Prompt (ALFWorld, Step 5)

You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: clean some cloth and put it in countertop.
Prior to this step, you have already taken 4 step(s). Below are the most recent 2 observaitons and the corresponding actions you took: [Observation 3: 'You pick up the cloth 2 from the countertop 1. Location: countertop 1. Items in hand (status): cloth 2(unprocessed).', Action 3: 'go to sinkbasin 1']
[Observation 4: 'You arrive at sinkbasin 1. On the sinkbasin 1, you see a sink 1, and a soapbar 2. Location: sinkbasin 1. Items in hand (status): cloth 2(unprocessed).', Action 4: 'clean cloth 2 with sinkbasin 1']
You are now at step 5 and your current observation is: You clean the cloth 2 using the sinkbasin 1. Location: sinkbasin 1. Items in hand (status): cloth 2(cleaned).
Your admissible actions of the current situation are: [clean cloth 2 with sinkbasin 1, examine cloth 2, examine sinkbasin 1, go to countertop 1, go to drawer 1, go to drawer 2, go to drawer 3, go to drawer 4, go to drawer 5, go to drawer 6, go to garbagecan 1, go to handtowelholder 1, go to handtowelholder 2, go to toilet 1, go to toiletpaperhanger 1, go to towelholder 1, go to towelholder 2, help, inventory, look, move cloth 2 to sinkbasin 1].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (ALFWorld, Step 5)

<think>After completing the previous step and cleaning the cloth using the sinkbasin 1, the cloth is now clean. Next, one should put the clean cloth in the countertop, ensuring everything is organized.</think>
<action>go to countertop 1</action>

Prompt (ALFWorld, Step 6)

You are an expert agent operating in the ALFRED Embodied Environment.
Your task is to: clean some cloth and put it in countertop.
Prior to this step, you have already taken 5 step(s). Below are the most recent 2 observaitons and the corresponding actions you took: [Observation 4: 'You arrive at sinkbasin 1. On the sinkbasin 1, you see a sink 1, and a soapbar 2. Location: sinkbasin 1. Items in hand (status): cloth 2(unprocessed).', Action 4: 'clean cloth 2 with sinkbasin 1']
[Observation 5: ' You clean the cloth 2 using the sinkbasin 1. Location: sinkbasin 1. Items in hand (status): cloth 2(cleaned).', Action 5: 'go to countertop 1']
You are now at step 6 and your current observation is: You arrive at countertop 1. On the countertop 1, you see a sink 1. Location: countertop 1. Items in hand (status): cloth 2(cleaned).
Your admissible actions of the current situation are: [examine cloth 2, examine countertop 1, go to drawer 1, go to drawer 2, go to drawer 3, go to drawer 4, go to drawer 5, go to drawer 6, go to garbagecan 1, go to handtowelholder 1, go to handtowelholder 2, go to sinkbasin 1, go to toilet 1, go to toiletpaperhanger 1, go to towelholder 1, go to towelholder 2, help, inventory, look, move cloth 2 to countertop 1].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (ALFWorld, Step 6)

<think>The user is currently at countertop 1 with the cloth 2 (status: cleaned).
The user needs to put the cleaned cloth 2 in the countertop 1.</think>
<action>move cloth 2 to countertop 1</action>

Follow, we present a complete case study of an agent trained by GroupGPO on WebShop, including the prompt and the response at each step. For each prompt, the components used to define the state in GroupGPO are highlighted in yellow.

Prompt (WebShop, Step 1)

You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.
Your current observation is: WebShop\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Search [button.]\n.
Your admissible actions of the current situation are: ['search[<your query>]', 'click[search]', \n.].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (WebShop, Step 1)

<think>To find the machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars, I should first search for these specific attributes on the WebShop.</think>
<action>search[men's dress shirts with polyester heathers navy x-large]</action>

Prompt (WebShop, Step 2)

You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.
Prior to this step, you have already taken 1 step(s). Below are the most recent 1 observaitons and the corresponding actions you took: [Observation 1: 'WebShop\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Search [button.]\n.', Action 1: 'search[men's dress shirts with polyester heathers navy x-large]']
You are now at step 2 and your current observation is: Instruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button.]\nPage 1 (Total results: 50)\n[button] Next >[button.]\n\n[button] B09RVF4JP1 [button.]\nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\n\$19.99\n\n[button] B09P39QN2W [button.]\nI Tackled 100 Day Of School Football Boy 100th Day School T-Shirt\n\$16.99\n\n[button] B09P7H5YK7 [button.]\nIt's In My DNA Palestinian Shirt Arabic Gifts Palestine Flag Tank Top\n\$19.99\n\n[button] B07JVVJDJ6L [button.]\nCanadian Tuxedo T Shirt - Novelty Denim Tux T-Shirt\n\$19.99\n\n[button] B07DKGJR74 [button.]\nAmazon Brand - Daily Ritual Women's 100% Cotton Oversized Fit V-Neck Pullover Sweater\n\$20.66 to \$29.2\n\n[button] B07T3VMGDG [button.]\nTuxedo Cat 4th of July Hat Patriotic Gift Adults Kids Raglan Baseball Tee\n\$23.99\n\n[button] B08HF132PN [button.]\nFlamingo Pink waterbird costume Gift Premium T-Shirt\n\$19.99\n\n[button] B07XPR3R7N [button.]\nMarvel Avengers: Endgame Captain America America's Language T-Shirt\n\$22.99\n\n[button] B07HRFSNL4 [button.]\nNautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt\n\$16.05 to \$40.98\n\n[button] B08226NDZW [button.]\nDisney Winnie The Pooh Tigger Upside Down Portrait Tank Top\n\$25.99\n.
Your admissible actions of the current situation are: ['click[back to search]', 'click[next >]', 'click[b09rvf4jp1]', 'click[b09p39qn2w]', 'click[b09p7h5yk7]', 'click[b07jvvdj6l]', 'click[b07dkgr74]', 'click[b07t3vmgdg]', 'click[b08hf132pn]', 'click[b07xpr3r7n]', 'click[b07hrfsl4]', 'click[b08226ndzw]', \n.].
Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (WebShop, Step 2)

<think>To find a machine washable men's dress shirt that matches the given criteria, I need to search the available options for matching criteria. The last search resulted in a few options that seemed acceptable, such as a Queen Born in 1982 T-Shirt. However, I want to look for higher quality and matching products.</think>
<action>click[b09rvf4jp1]</action>

Prompt (WebShop, Step 3)

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

Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.

Prior to this step, you have already taken 2 step(s). Below are the most recent 2 observations and the corresponding actions you took: [Observation 1: 'WebShop\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Search [button.]\n.', Action 1: 'search[men's dress shirts with polyester heathers navy x-large]']

[Observation 2: 'Instruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button.]\nPage 1 (Total results: 50)\n[button] Next >[button.]\n[button] B09RVF4JP1 [button.] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\n\$19.99\n[button] B09P39QN2W [button.] \nI Tackled 100 Day Of School Football Boy 100th Day School T-Shirt\n\$16.99\n[button] B09P7H5YK7 [button.] \nIt's In My DNA Palestinian Shirt Arabic Gifts Palestine Flag Tank Top\n\$19.99\n[button] B07JVVDJ6L [button.] \nCanadian Tuxedo T Shirt - Novelty Denim Tux T-Shirt\n\$19.99\n[button] B07DKGJR74 [button.] \nAmazon Brand - Daily Ritual Women's 100% Cotton Oversized Fit V-Neck Pullover Sweater\n\$20.66 to \$29.2\n[button] B07T3VMGDG [button.] \nTuxedo Cat 4th of July Hat Patriotic Gift Adults Kids Raglan Baseball Tee\n\$23.99\n[button] B08HF132PN [button.] \nFlamingo Pink waterbird costume Gift Premium T-Shirt\n\$19.99\n[button] B07XPR3R7N [button.] \nMarvel Avengers: Endgame Captain America America's Language T-Shirt\n\$22.99\n[button] B07HRFSNL4 [button.] \nNautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt\n\$16.05 to \$40.98\n[button] B08226NDZW [button.] \nDisney Winnie The Pooh Tigger Upside Down Portrait Tank Top\n\$25.99\n.', Action 2: 'click[b09rvf4jp1]']

You are now at step 3 and your current observation is: Instruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button.] \n[button] <Prev [button.] \nfit type\n[button] men [button.] \n[button] women [button.] \nncolor\n[button] black [button.] \n[button] navy [button.] \n[button] dark heather [button.] \n[button] neon pink [button.] \n[button] royal blue [button.] \nsize\n[button] small [button.] \n[button] medium [button.] \n[button] large [button.] \n[button] x-large [button.] \n[button] xx-large [button.] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A.\n[button] Description [button.] \n[button] Features [button.] \n[button] Reviews [button.] \n[button] Buy Now [button.] \n.

Your admissible actions of the current situation are: ['click[back to search]', 'click[<prev]', 'click[description]', 'click[features]', 'click[reviews]', 'click[buy now]', 'click[men]', 'click[women]', 'click[black]', 'click[navy]', 'click[dark heather]', 'click[neon pink]', 'click[royal blue]', 'click[small]', 'click[medium]', 'click[large]', 'click[x-large]', 'click[xx-large]'],\n

Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (WebShop, Step 3)

<think> Step 1: I will ensure a thorough search within the current webshop results for men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.

Step 2: I will check if the current observation contains relevant items matching the criteria mentioned.

Step 3: I will consider the requirements, explore available options, and price point to determine item quality.</think>

<action>click[men]</action>

Prompt (WebShop, Step 4)

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

Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.

Prior to this step, you have already taken 3 step(s). Below are the most recent 2 observations and the corresponding actions you took: [Observation 2: 'Instruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button.]\nPage 1 (Total results: 50)\n[button] Next >[button.]\n\n[button] B09RVF4JP1 [button.]\nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\n\$19.99\n[button] B09P39QN2W [button.]\nI Tackled 100 Day Of School Football Boy 100th Day School T-Shirt\n\$16.99\n[button] B09P7H5YK7 [button.]\nIt's In My DNA Palestinian Shirt Arabic Gifts Palestine Flag Tank Top\n\$19.99\n[button] B07JVVDJ6L [button.]\nCanadian Tuxedo T Shirt - Novelty Denim Tux T-Shirt\n\$19.99\n[button] B07DKGJR74 [button.]\nAmazon Brand - Daily Ritual Women's 100% Cotton Oversized Fit V-Neck Pullover Sweater\n\$20.66 to \$29.2\n[button] B07T3VMGDG [button.]\nTuxedo Cat 4th of July Hat Patriotic Gift Adults Kids Raglan Baseball Tee\n\$23.99\n[button] B08HF132PN [button.]\nFlamingo Pink waterbird costume Gift Premium T-Shirt\n\$19.99\n[button] B07XPR3R7N [button.]\nMarvel Avengers: Endgame Captain America America's Language T-Shirt\n\$22.99\n[button] B07HRFSNL4 [button.]\nNautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt\n\$16.05 to \$40.98\n[button] B08226NDZW [button.]\nDisney Winnie The Pooh Tigger Upside Down Portrait Tank Top\n\$25.99\n.', Action 2: 'click[b09rvf4jp1]']

[Observation 3: 'Instruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button.]\n[button] <Prev [button.]\nfit type\n[button] men [button.]\n[button] women [button.]\nncolor\n[button] black [button.]\n[button] navy [button.]\n[button] dark heather [button.]\n[button] neon pink [button.]\n[button] royal blue [button.]\nsize\n[button] small [button.]\n[button] medium [button.]\n[button] large [button.]\n[button] x-large [button.]\n[button] xx-large [button.] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A.\n[button] Description [button.]\n[button] Features [button.]\n[button] Reviews [button.] \n[button] Buy Now [button.] \n.', Action 3: 'click[men]']

You are now at step 4 and your current observation is: You have clicked men.\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button.] \n[button] ; Prev [button.] \nfit type\n[button] men [button] women [button.] \nncolor\n[button] black [button.] \n[button] navy [button.] \n[button] dark heather [button.] \n[button] neon pink [button.] \n[button] royal blue [button.] \nsize\n[button] small [button.] \n[button] medium [button.] \n[button] large [button.] \n[button] x-large [button.] \n[button] xx-large [button.] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A.\n[button] Description [button.] \n[button] Features [button.] \n[button] Reviews [button.] \n[button] Buy Now [button.] \n.

Your admissible actions of the current situation are: ['click[back to search]', 'click[<prev]', 'click[description]', 'click[features]', 'click[reviews]', 'click[buy now]', 'click[men]', 'click[women]', 'click[black]', 'click[navy]', 'click[dark heather]', 'click[neon pink]', 'click[royal blue]', 'click[small]', 'click[medium]', 'click[large]', 'click[x-large]', 'click[xx-large]', 'n']

Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (WebShop, Step 4)

<think>The user has just clicked "men" which suggests the search is further targeting items of that size. Now, we need to filter the results based on color, size, price, and fit in an attempt to find machine wash men's dress shirts with polyester heathers that match all the specified criteria.</think>

<action>click[navy]</action>

Prompt (WebShop, Step 5)

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

Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.

Prior to this step, you have already taken 4 step(s). Below are the most recent 2 observations and the corresponding actions you took: [Observation 3: 'Instruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button_] \n[button] <Prev [button_] \nfit type\n[button] men [button_] \n[button] women [button_] \ncolor\n[button] black [button_] \n[button] navy [button_] \n[button] dark heather [button_] \n[button] neon pink [button_] \n[button] royal blue [button_] \nsize\n[button] small [button_] \n[button] medium [button_] \n[button] large [button_] \n[button] x-large [button_] \n[button] xx-large [button_] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A. \n[button] Description [button_] \n[button] Features [button_] \n[button] Reviews [button_] \n[button] Buy Now [button_] \n.', Action 3: 'click[men]']

[Observation 4: 'You have clicked men.\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button_] \n[button] ; Prev [button_] \nfit type\n[button] [clicked button] men [clicked button_] \n[button] women [button_] \ncolor\n[button] black [button_] \n[button] navy [button_] \n[button] dark heather [button_] \n[button] neon pink [button_] \n[button] royal blue [button_] \nsize\n[button] small [button_] \n[button] medium [button_] \n[button] large [button_] \n[button] x-large [button_] \n[button] xx-large [button_] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A. \n[button] Description [button_] \n[button] Features [button_] \n[button] Reviews [button_] \n[button] Buy Now [button_] \n.', Action 4: 'click[navy]']

You are now at step 5 and your current observation is: You have clicked navy.\nYou have clicked men.\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button_] \n[button] ; Prev [button_] \nfit type\n[button] [clicked button] men [clicked button_] \n[button] women [button_] \ncolor\n[button] black [button_] \n[button] navy [button_] \n[button] dark heather [button_] \n[button] neon pink [button_] \n[button] royal blue [button_] \nsize\n[button] small [button_] \n[button] medium [button_] \n[button] large [button_] \n[button] x-large [button_] \n[button] xx-large [button_] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A. \n[button] Description [button_] \n[button] Features [button_] \n[button] Reviews [button_] \n[button] Buy Now [button_] \n.

Your admissible actions of the current situation are: ['click[back to search]', 'click[; prev]', 'click[description]', 'click[features]', 'click[reviews]', 'click[buy now]', 'click[men]', 'click[women]', 'click[black]', 'click[navy]', 'click[dark heather]', 'click[neon pink]', 'click[royal blue]', 'click[small]', 'click[medium]', 'click[large]', 'click[x-large]', 'click[xx-large]', '\n']

Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (WebShop, Step 5)

<think>Current Observation: I have selected the color navy and size x-large. Next Step: I need to find machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit, color: navy, and size: x-large, and price lower than \$50. Frequency: - cache miss</think>

<action>click[x-large]</action>

Prompt (WebShop, Step 6)

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

Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars.

Prior to this step, you have already taken 5 step(s). Below are the most recent 2 observations and the corresponding actions you took: [Observation 4: 'You have clicked men.\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button_] \n [button] ; Prev [button_] \nfit type\n [clicked button] men [clicked button_] \n [button] women [button_] \nncolor\n [button] black [button_] \n [button] navy [button_] \n [button] dark heather [button_] \n [button] neon pink [button_] \n [button] royal blue [button_] \nsize\n [button] small [button_] \n [button] medium [button_] \n [button] large [button_] \n [button] x-large [button_] \n [button] xx-large [button_] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A.\n[button] Description [button_] \n [button] Features [button_] \n [button] Reviews [button_] \n [button] Buy Now [button_] \n.', Action 4: 'click[navy]']

[Observation 5: 'You have clicked navy.\nYou have clicked men.\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button_] \n [button] ; Prev [button_] \nfit type\n [clicked button] men [clicked button_] \n [button] women [button_] \nncolor\n [button] black [button_] \n [button] navy [clicked button_] \n [button] dark heather [button_] \n [button] neon pink [button_] \n [button] royal blue [button_] \nsize\n [button] small [button_] \n [button] medium [button_] \n [button] large [button_] \n [button] x-large [button_] \n [button] xx-large [button_] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A.\n[button] Description [button_] \n [button] Features [button_] \n [button] Reviews [button_] \n [button] Buy Now [button_] \n.', Action 5: 'click[x-large]']

You are now at step 6 and your current observation is: You have clicked x-large.\nYou have clicked navy.\nYou have clicked men.\nInstruction:\nFind me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: navy, and fit type: men, and size: x-large, and price lower than 50.00 dollars\n[button] Back to Search [button_] \n [button] ; Prev [button_] \nfit type\n [clicked button] men [clicked button_] \n [button] women [button_] \nncolor\n [button] black [button_] \n [button] navy [clicked button_] \n [button] dark heather [button_] \n [button] neon pink [button_] \n [button] royal blue [button_] \nsize\n [button] small [button_] \n [button] medium [button_] \n [button] large [button_] \n [button] x-large [clicked button_] \n [button] xx-large [button_] \nOnce Upon A Time Queen Born In 1982 T Shirts It Was Me Tank Top\nPrice: \$19.99\nRating: N.A.\n[button] Description [button_] \n [button] Features [button_] \n [button] Reviews [button_] \n [button] Buy Now [button_] \n.

Your admissible actions of the current situation are: ['click[back to search]', 'click[; prev]', 'click[description]', 'click[features]', 'click[reviews]', 'click[buy now]', 'click[men]', 'click[women]', 'click[black]', 'click[navy]', 'click[dark heather]', 'click[neon pink]', 'click[royal blue]', 'click[small]', 'click[medium]', 'click[large]', 'click[x-large]', 'click[xx-large]', '\n']

Now it's your turn to take an action.\nYou should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think> </think> tags. \n Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Response (WebShop, Step 6)

<think>Found the machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit, navy color with fit type: x-large, and price lower than 50.00 dollars.</think>

<action>click[buy now]</action>