StraTA: Incentivizing Agentic Reinforcement Learning with Strategic Trajectory Abstraction

Xiangyuan Xue[*1,2] **Yifan Zhou**[*3] **Zidong Wang**[1] **Shengji Tang**[1]
Philip Torr[4] **Wanli Ouyang**[1,2,5] **Lei Bai**[2] **Zhenfei Yin**[4]

Abstract

Large language models (LLMs) are increasingly used as interactive agents, but optimizing them for long-horizon decision making remains difficult because current methods are largely purely reactive, which weakens both exploration and credit assignment over extended trajectories. In this work, we present **Stra**tegic **T**rajectory **A**bstraction (StraTA), a simple framework that introduces an explicit trajectory-level strategy into agentic reinforcement learning (RL). StraTA samples a compact strategy from the initial task state, conditions subsequent actions on that strategy, and trains strategy generation and action execution jointly with a hierarchical GRPO-style rollout design, further enhanced by diverse strategy rollout and critical self-judgment. Experiments on ALFWorld, WebShop, and SciWorld show that StraTA consistently improves both sample efficiency and final performance over strong baselines. StraTA reaches success rates of 93.1% on ALFWorld and 84.2% on WebShop, surpassing the strongest baselines by 2.3% and 11.4% respectively. On SciWorld, StraTA attains a 63.5% overall score, outperforming frontier closed-source models by 6.1% and prior RL baselines by 6.5%. Our code is available at: https://github.com/xxyQwQ/StraTA.

1. Introduction

Large language models (LLMs) are increasingly deployed as interactive agents that perceive environments, reason in natural language, and act through external tools or APIs. This paradigm has enabled rapid progress on complex tasks such as web interaction, computer operation, and scientific problem solving (Xi et al., 2025a; Wang et al., 2024a). Nevertheless, strong general-purpose capabilities do not automatically translate into reliable performance on specialized long-horizon tasks. Agents must often recover from earlier mistakes, maintain coherent behavior over many turns, and make decisions whose consequences may only become apparent much later in the trajectory.

Agentic reinforcement learning (RL) provides a natural way to improve such behavior by optimizing agents from environment feedback (Zhang et al., 2025a). Compared with conventional language modeling, however, this setting is particularly challenging. Each decision is hierarchical: the policy must choose a step-level action while autoregressively generating the token sequence that realizes it. At the same time, rewards are often sparse and delayed until the episode terminates. These properties make long-horizon exploration difficult and complicate credit assignment.

Recent methods have made encouraging progress. PPO-style and GRPO-style training have shown that RL can improve multi-turn LLM agents (Xi et al., 2025c; Tan et al., 2025), while newer variants further refine the learning signal for long-horizon interaction (Zhou et al., 2024b; Feng et al., 2025b). However, most existing formulations remain purely reactive. As shown in Figure 1a, at each step the agent predicts the next action solely from the current state. The policy therefore has to decide both the immediate next move and the overall course of action at the same time. This coupling often yields short-sighted exploration, unnecessary backtracking, and internally inconsistent behavior.

We argue that a key missing abstraction is an explicit trajectory-level strategy (Erdogan et al., 2025). Human problem solving is rarely purely reactive: people typically form a high-level plan, execute local actions under that plan, and later reflect on whether the plan was appropriate. Motivated by this observation, we propose **Stra**tegic **T**rajectory **A**bstraction (StraTA), a framework that introduces explicit strategy guidance into agentic RL. Figure 1b illustrates how StraTA differs from reactive agents. At the beginning of

[1]The Chinese University of Hong Kong, Hong Kong, China [2]Shanghai Artificial Intelligence Laboratory, Shanghai, China [3]University of Georgia, Athens, GA, USA [4]University of Oxford, Oxford, UK [5]Shenzhen Loop Area Institute, Shenzhen, China. Correspondence to: Xiangyuan Xue <xxythecoder@gmail.com>, Lei Bai <baisanshi@gmail.com>, Zhenfei Yin <zhenfei.yin@eng.ox.ac.uk>.

ICML 2026 Workshop on Failure Modes in Agentic AI. *Copyright 2026 by the author(s).*

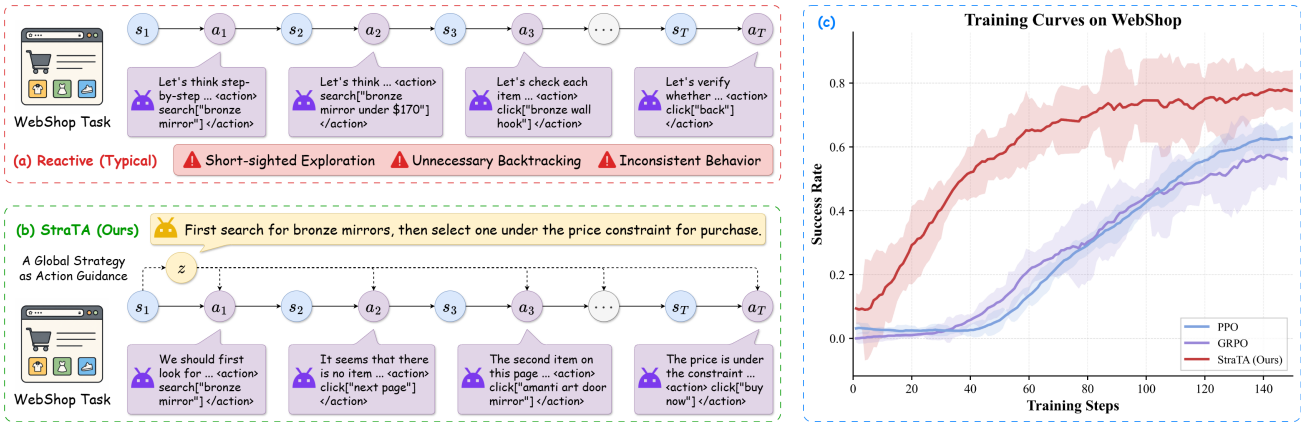

Figure 1. A comparison between reactive agent and our proposed StraTA framework. (a) Reactive agent generates actions from local states alone, which may lead to short-sighted exploration, unnecessary backtracking, and inconsistent behavior. (b) StraTA framework first generates a compact global strategy and conditions all subsequent actions on this, enabling more coherent and consistent execution. (c) On long-horizon agentic tasks, explicit strategy guidance significantly improves learning efficiency and final performance.

each episode, StraTA samples a strategy from the initial state, which is then fixed and prepended to every decision, so each action is generated from both the global strategy and the local state. In this way, long-horizon agentic decision making is decomposed into two objectives: generating a good strategy and executing actions that faithfully follow it.

To optimize these two levels jointly, StraTA builds a hierarchical rollout structure on top of GRPO-style training. For each task, the policy first samples multiple strategies and then performs multiple rollouts under each strategy. This design enables strategy-level comparisons across different plans and action-level comparisons across trajectories generated under the same plan. Beyond trajectory-level outcome rewards, StraTA further assigns each strategy a reward based on the top-performing fraction of its rollouts, estimating the strategy quality. StraTA also incorporates two simple techniques that further facilitate learning: diverse strategy rollout via farthest-point sampling, which broadens semantic exploration over candidate strategies, and a step-level auxiliary reward via critical self-judgment, which penalizes actions that fail to follow the strategy.

Experiments on multiple agentic benchmarks show that StraTA improves both sample efficiency and final task performance. On ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022), StraTA consistently outperforms both prompting-based agents and RL methods across model scales: with a 1.5B backbone, it reaches 90.7% success on ALFWorld and 82.5% on WebShop, surpassing GiGPO (Feng et al., 2025b) by up to 4.0% and 17.5%, respectively, and with a 7B backbone, it further improves to 93.1% and 84.2%. On the more challenging SciWorld (Wang et al., 2022) benchmark, StraTA attains a 63.5% overall score, outperforming both frontier closed-source models and prior RL baselines by 6.1% and 6.5%, respectively, while achieving a perfect 100.0% score on the Lifespan subset. In summary, our contributions are threefold: (1) we introduce StraTA, a strategy-guided framework that equips agentic RL with explicit trajectory-level abstraction; (2) we develop a hierarchical GRPO-style training with top-performing strategy rewards, diverse strategy rollout, and critical self-judgment to improve exploration and credit assignment; and (3) we validate its effectiveness through extensive experiments on long-horizon agentic tasks.

2. Related Work

LLM-based agents. LLM-based agents augment foundation models with external tools, such as search engines and code interpreters, enabling them to address complex real-world tasks (Xi et al., 2025a; Wang et al., 2024a). Early work primarily relied on prompting-based designs to build runnable agents for relatively simple and specialized tasks (Yao et al., 2023; Wang et al., 2024b; Schick et al., 2023; Shinn et al., 2023). More recent studies introduce training-based methods to improve performance on more challenging and general tasks (Zeng et al., 2024; Song et al., 2024; Xi et al., 2025b;c). A parallel line of work investigates multi-agent collaboration as a way to further enhance task performance (Hong et al., 2024; Wu et al., 2024; Du et al., 2024; Zhuge et al., 2024; Xue et al., 2025). These agents have now been applied to a wide range of domains, from repetitive short-horizon routines to more autonomous long-horizon settings, including computer operation (Yao et al., 2022; Zhou et al., 2024a; Xie et al., 2024), embodied manipulation (Shridhar et al., 2021; Wang et al., 2023; Chen et al., 2025b), and scientific discovery (Wang et al., 2022; Lu et al., 2024; Xu et al., 2025). In this work, we study how RL can further extend the capabilities of LLM-based agents on difficult long-horizon tasks.

Reinforcement fine-tuning. RL was first introduced to LLMs mainly for alignment with human preferences, a paradigm widely known as reinforcement learning from human feedback (RLHF) (Ziegler et al., 2019; Ouyang et al., 2022). Most early methods in this line are based upon proximal policy optimization (PPO) (Schulman et al., 2017), with later variants improving training stability and efficiency (Rafailov et al., 2023; Azar et al., 2024; Ethayarajh et al., 2024; Ahmadian et al., 2024; Hu et al., 2025). More recently, reinforcement learning with verifiable rewards (RLVR) has become a popular paradigm for improving reasoning, where group relative policy optimization (GRPO) (Shao et al., 2024) serves as a representative foundation and has demonstrated strong scalability on complex reasoning tasks (DeepSeek-AI et al., 2025). Many follow-up methods further improve its effectiveness (Liu et al., 2025; Lin et al., 2025; Yu et al., 2025; Zheng et al., 2025). Another direction explores RL with intrinsic reward signals in the absence of ground-truth labels (Yuan et al., 2024; Prabhudesai et al., 2025; Zhao et al., 2025; Zhang et al., 2025b; Zuo et al., 2025), which also shows promising results. In this work, we adopt a GRPO-style training framework, but focus on the more challenging setting of long-horizon agentic tasks rather than single-turn reasoning problems.

Agentic reinforcement learning. Agentic RL is a distinct emerging area in which agents learn long-horizon interactive behavior directly through trial and error in an environment (Zhang et al., 2025a). Compared with conventional LLM-based agents, it requires online exploration and adaptation rather than the execution of a fixed routine. Compared with reinforcement fine-tuning, it places greater emphasis on long-horizon decision-making instead of single-turn reasoning. Representative tasks span computer operation (Yao et al., 2022; Zhou et al., 2024a; Xie et al., 2024), interactive gameplay (Wang et al., 2023; Zhai et al., 2024; Xi et al., 2025b;c), and embodied manipulation (Shridhar et al., 2021; Wang et al., 2022). A growing body of work aims at addressing these challenges. Beyond efforts that establish training frameworks (Luo et al., 2025; Feng et al., 2025b; Tan et al., 2025), one research direction focuses on improving active exploration in multi-turn interaction (Putta et al., 2024; Wang et al., 2025; Feng et al., 2025a), while another studies more accurate and fine-grained credit assignment (Zhou et al., 2024b; Liu et al., 2026; Ji et al., 2026; Feng et al., 2025b; He et al., 2026; Peng et al., 2026). Our proposed StraTA framework supports both goals through a simple but effective design that introduces an explicit trajectory-level strategy to guide the entire episode.

3. Preliminaries

Long-horizon agentic tasks. We formulate a long-horizon agentic task as a finite-horizon Markov decision process (MDP) (Zhang et al., 2025a), $(\mathcal{S}, \mathcal{A}, P, R, H, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ are the state and action spaces, $P : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is the transition function, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, H is the horizon, and γ is the discount factor. In agentic RL, both states and actions are represented in natural language. At step t, the state s_t typically contains the task specification, interaction history, and additional action instructions. An episode produces a trajectory $\tau = \{(s_1, a_1, r_1), (s_2, a_2, r_2), \cdots, (s_T, a_T, r_T)\}$, where $T \leq H$ is the episode length. Because the environment usually provides only a terminal outcome reward, we set $\gamma = 1$ and denote the trajectory-level reward by $R(\tau)$.

LLM-driven agent policy. The agent policy π_θ, parameterized by θ, is instantiated by an LLM and can be described at two equivalent levels. At the action level, the policy maps the current state s_t to an action executed in the environment, i.e., $a_t \sim \pi_\theta(\cdot \mid s_t)$. At the token level, each action is generated auto-regressively as a token sequence $a_t = \{a_{t,k}\}_{k=1}^{K_t}$, where $a_{t,k} \sim \pi_\theta(\cdot \mid s_t, a_{t,<k})$. Rewards are defined at the action level and broadcast to all tokens in the action. Generally, the RL objective is:

$$J_{\text{RL}}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}\left[R(\tau)\right]. \tag{1}$$

We further assume a non-cumulative setting (Feng et al., 2025b), in which each step is generated from an individual prompt constructed from the current state rather than from the full history. This design helps stabilize training by keeping the effective context length under control.

Group-based reinforcement learning. Classical actor-critic methods, such as PPO (Schulman et al., 2017; Ouyang et al., 2022), jointly optimize an actor and a critic. Although effective, they introduce additional computation and memory cost. In contrast, group-based RL methods represented by GRPO (Shao et al., 2024) remove the explicit critic and derive the learning signal from relative reward comparisons within a sampled group. This formulation extends naturally to agentic tasks by assigning the trajectory-level reward to all actions in the episode. To be specific, for each task, GRPO samples a group of G trajectories $\{\tau^i\}_{i=1}^G$ from the old policy $\pi_{\theta_{\text{old}}}$. Each trajectory receives a scalar reward $r^i = R(\tau^i)$, which is then normalized within the group to produce the corresponding advantage:

$$A^i = \frac{r^i - \text{mean}(\{r^i\}_{i=1}^G)}{\text{std}(\{r^i\}_{i=1}^G)}. \tag{2}$$

For simplicity, we define the generic clipped surrogate as:

$$\mathcal{L}(x, A; \theta) = \frac{1}{|x|} \sum_{k=1}^{|x|} \min\Big(\rho_{x,k}(\theta) A, \; \text{clip}\left(\rho_{x,k}(\theta), 1-\epsilon, 1+\epsilon\right) A\Big), \tag{3}$$

where x is an arbitrary response, $|x|$ denotes the number of tokens in x, A denotes the corresponding advantage, and

$\rho_{x,k}(\theta) = \frac{\pi_\theta(x_k|s_t, x_{<k})}{\pi_{\theta_{\text{old}}}(x_k|s_t, x_{<k})}$ defines the importance sampling ratio for token x_k. Then GRPO optimizes the following objective built upon the generic clipped surrogate:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\{\tau^i\}_{i=1}^G \sim \pi_\theta}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\tau^i|}\sum_{t=1}^{|\tau^i|}\mathcal{L}(a_t^i, A^i; \theta) - \beta D_{\text{KL}}\left(\pi_\theta \| \pi_{\text{ref}}\right)\Bigg], \tag{4}$$

where $|\tau^i|$ is the number of steps in trajectory τ^i. By forgoing the explicit critic, group-based RL achieves significant gains in computational and memory efficiency while maintaining scalability, making it particularly well-suited for training LLM-based agents.

4. Method

Motivated by how humans solve complex interactive tasks in real-world scenarios, we propose **Stra**tegic **T**rajectory **A**bstraction (StraTA), a framework that introduces explicit strategy guidance into agentic RL. Instead of generating actions purely reactively from the current state, StraTA first produces a compact natural-language strategy at the beginning of each episode and conditions all subsequent actions on it. This design decomposes learning into two simpler objectives: (1) generating feasible strategies, and (2) executing actions that consistently follow them.

In this section, we will describe StraTA in four parts as shown in Figure 2. First, we will present the strategy-guided task execution process. Then, we will explain how StraTA samples trajectories and constructs hierarchical groups for GRPO-style training. Finally, we will introduce two techniques that respectively support the two learning objectives mentioned above: (1) diverse strategy rollout via farthest point sampling, and (2) step-level auxiliary reward via critical self-judgment.

4.1. Strategy-guided Task Execution

In typical agentic RL, the agent acts reactively: at each step t, it generates the current action a_t directly from the current state s_t, i.e., $a_t \sim \pi_\theta(\cdot \mid s_t)$. Under this formulation, the trajectory is determined entirely by step-by-step decisions, without explicit trajectory-level guidance. The policy therefore has to infer both the long-term course of action and the immediate next action at the same time, which often leads to short-sighted behavior, unnecessary backtracking, and inconsistent decisions.

StraTA augments this process by inserting a strategy generation step before interaction begins. Given the initial state s_1, which contains the task description, the agent first samples a global strategy:

$$z \sim \pi_\theta(\cdot \mid s_1), \tag{5}$$

where z is a compact natural-language plan that summarizes the anticipated course of action for the entire episode. The strategy is generated once at the beginning of the episode and kept fixed thereafter. Conditioned on z, action generation is no longer purely reactive. At each step t, the agent generates its action a_t based on both the global strategy z and the current state s_t:

$$a_t \sim \pi_\theta(\cdot \mid z, s_t), \tag{6}$$

where z serves as a trajectory-level signal that is prepended to the state at every step, so each action is generated from both the global strategy and the local context. The resulting trajectory becomes

$$\tau = \{z, (s_1, a_1, r_1), \cdots, (s_T, a_T, r_T)\}. \tag{7}$$

This reformulation decomposes task solving into two subgoals: (1) generating high-quality strategies from the initial task specification, and (2) executing actions that consistently follow the generated strategy throughout the episode. By separating these two roles, StraTA turns the original fully reactive decision process into a more structured one. Long-horizon behavior can now be coordinated through an explicit strategy instead of being inferred implicitly from local action choices alone. We will next describe how to train this strategy-guided policy with a GRPO-style objective.

4.2. Hierarchical Group Construction

To jointly optimize strategy generation and action taking, StraTA organizes rollouts into a two-level hierarchy on top of GRPO-style training. Figure 2a intuitively illustrates the sampling process. For each task, the agent first samples N strategies $\{z^i\}_{i=1}^N$ and then performs M independent rollouts $\{\tau^{i,j}\}_{j=1}^M$ under each strategy z^i, which yields $N \times M$ trajectories in total. This construction induces one strategy-level group and N action-level groups:

$$\begin{aligned} \mathcal{G}_{\text{strategy}} &= \{z^i \mid 1 \le i \le N\}, \\ \mathcal{G}^i_{\text{action}} &= \{a_t^{i,j} \mid 1 \le j \le M,\, 1 \le t \le |\tau^{i,j}|\}. \end{aligned} \tag{8}$$

The strategy-level group compares multiple strategies, identifying which high-level plans are more promising. Each action-level group compares multiple rollouts under the same strategy, revealing which trajectories are more likely to solve the task while adhering to that strategy.

At the action level, each trajectory $\tau^{i,j}$ receives an outcome reward $R(\tau^{i,j})$ at the end of the episode. Because we adopt a non-cumulative setting, this reward will be broadcast to all action steps $\{a_t^{i,j}\}_{t=1}^{|\tau^{i,j}|}$ within the episode:

$$R(a_t^{i,j}) = R(\tau^{i,j}), \tag{9}$$

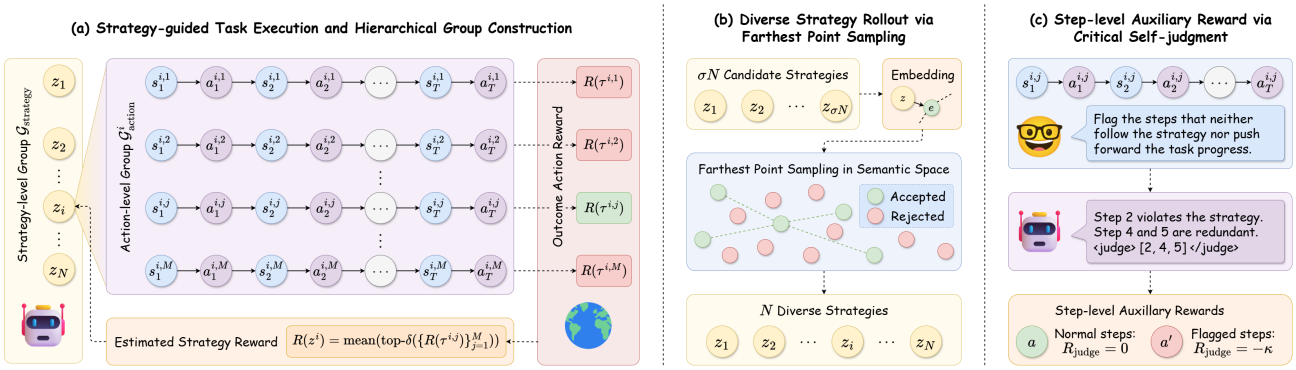

Figure 2. An overview of the StraTA framework. (a) For a given task, StraTA generates multiple strategies and performs multiple rollouts for each strategy, forming hierarchical groups for strategy-level and action-level optimization. (b) Diverse strategy rollout uses farthest point sampling to select semantically distinct strategies. (c) Critical self-judgment identifies action steps that neither follow the strategy nor advance task progress, and assigns a step-level auxiliary penalty to achieve more fine-grained credit assignment.

so that every action step can receive a learning signal even though the reward is returned only at episode termination.

At the strategy level, each strategy z^i is assigned a reward based on the rollouts $\{\tau^{i,j}\}_{j=1}^M$ generated under it. Instead of averaging over all rollouts equally, StraTA uses the mean of the top-δ fraction of outcome rewards, where δ controls the fraction of rewards to be considered:

$$R(z^i) = \text{mean}\left(\text{top-}\delta\left(\{R(\tau^{i,j})\}_{j=1}^M\right)\right), \tag{10}$$

which emphasizes the best performance achievable under a strategy and reduces the influence of low-quality rollouts caused by early-stage action noise. As a result, the reward more faithfully reflects strategy quality rather than imperfections in action execution.

Both the action and strategy rewards are augmented with two auxiliary terms. GRPO-style training often encourages long responses. To keep generations within the context budget, we follow prior work (Yeo et al., 2025; Liu et al., 2025; Yu et al., 2025) and introduce a soft length penalty:

$$R_{\text{length}}(x) = \begin{cases} 0, & |x| \le \lambda L_{\text{total}} \\ -\frac{1}{1-\lambda}\left(\frac{|x|}{L_{\text{total}}} - \lambda\right), & \lambda L_{\text{total}} < |x| \le L_{\text{total}} \\ -1, & L_{\text{total}} < |x| \end{cases}, \tag{11}$$

where x can be either a strategy or an action, $|x|$ is the number of tokens in x, L_{total} is the response length limit, and λ controls when the penalty begins. This term softly discourages overlong generations and helps stabilize training. In addition, to encourage responses that can be executed by the environment, we introduce a hard format penalty:

$$R_{\text{format}}(x) = \begin{cases} 0, & x \text{ is correct in format} \\ -1, & x \text{ is incorrect in format} \end{cases}, \tag{12}$$

which penalizes only unrecognized responses and further stabilizes training. The final reward is

$$\hat{R}(x) = \text{clip}\left(R(x) + R_{\text{length}}(x) + R_{\text{format}}(x), -1, 1\right), \tag{13}$$

which clips the total reward to $[-1, 1]$ and applies to both strategy and action steps.

All resulting groups are compatible with the GRPO-style training introduced in Section 3. Accordingly, advantage computation in Equation 2 is applied unchanged to both the strategy-level group $\mathcal{G}_{\text{strategy}}$ and each action-level group $\mathcal{G}^i_{\text{action}}$. The overall StraTA training objective is:

$$\mathcal{J}_{\text{StraTA}}(\theta) = \mathbb{E}_{\mathcal{G}_{\text{strategy}}, \{\mathcal{G}^i_{\text{action}}\}_{i=1}^N \sim \pi_\theta}\Bigg[\sum_{z^i \in \mathcal{G}_{\text{strategy}}} \mathcal{L}(z^i, A(z^i); \theta) + \sum_{i=1}^N \sum_{a_t^{i,j} \in \mathcal{G}^i_{\text{action}}} \mathcal{L}(a_t^{i,j}, A(a_t^{i,j}); \theta) - \beta D_{\text{KL}}(\pi_\theta \,\|\, \pi_{\text{ref}})\Bigg], \tag{14}$$

where $A(z^i)$ and $A(a_t^{i,j})$ denote the computed advantages for z^i and $a_t^{i,j}$, respectively.

4.3. Diverse Strategy Rollout via Farthest Point Sampling

Prior work (Cui et al., 2025; Chen et al., 2025a; Yu et al., 2025) has shown that rollout diversity is crucial for agentic RL. However, even with elevated sampling temperature, the model produces highly similar strategies within the same group, which limits exploration of the strategy space.

As shown in Figure 2b, to encourage diverse strategies, StraTA first oversamples $\sigma \times N$ candidate strategies and then applies farthest point sampling to select N strategies that are distinct from one another, where σ controls the oversampling ratio. To quantify semantic diversity, each candidate strategy z^i is encoded as a normalized vector $e^i = \phi(z^i)$ using a pre-trained embedding model $\phi(\cdot)$ (Reimers

& Gurevych, 2019), so ${e^i}^\top e^j$ is the cosine similarity.

The selection proceeds greedily. We initialize the selected set $\mathcal{Z}$ with the candidate z_{center} that is closest to the centroid, yielding a representative starting point. At each subsequent round, we add the candidate with the smallest maximum similarity to the already selected set $\mathcal{Z}$:

$$z_{\text{select}} = \arg\min_{z^i \notin \mathcal{Z}} \max_{z^j \in \mathcal{Z}} {e^i}^\top e^j. \tag{15}$$

This process repeats until N strategies have been selected, producing a diverse set of strategy rollouts. As a result, strategy-level comparison in GRPO becomes more informative because the sampled strategies provide meaningful contrast rather than minor variations. It also exposes the agent to a broader range of strategic behaviors, encouraging more effective exploration.

4.4. Step-level Auxiliary Reward via Critical Self-judgment

Another key challenge is credit assignment. Sparse outcome rewards make it difficult to identify which actions truly contribute to task success. Moreover, optimizing only for final success may favor opportunistic actions that work in hindsight but do not faithfully follow the intended strategy.

As shown in Figure 2c, to address this issue, StraTA introduces a self-judgment mechanism inspired by LLM-as-a-judge techniques (Zheng et al., 2023; Gu et al., 2024; Li et al., 2025) to achieve fine-grained credit assignment. After a rollout is completed, the agent retrospectively evaluates the trajectory against its guiding strategy. Concretely, given an episode $\tau^{i,j}$ under strategy z^i, the agent is instructed to flag all action steps that neither contribute to the final success nor follow the global strategy:

$$\mathcal{I}^{i,j} \sim \pi_\theta(\cdot \mid \mathcal{P}, \tau^{i,j}), \tag{16}$$

where $\mathcal{P}$ is the self-judgment prompt and $\mathcal{I}^{i,j}$ is the set of flagged action-step indices. The self-judgment is performed by the agent itself rather than by a separate judge model, and these judging steps do not receive reward. The resulting step-level auxiliary reward for each action $a_t^{i,j}$ is

$$R_{\text{judge}}(a_t^{i,j}) = -\kappa \cdot \mathbb{1}[t \in \mathcal{I}^{i,j}], \tag{17}$$

where κ controls the weight of the self-judgment penalty. This term is then additioanally added to the action reward, replacing Equation 13 for action steps:

$$\begin{aligned}\hat{R}(a_t^{i,j}) = \text{clip}\Big(&R(a_t^{i,j}) + R_{\text{length}}(a_t^{i,j}) \\ &+ R_{\text{format}}(a_t^{i,j}) + R_{\text{judge}}(a_t^{i,j}), -1, 1\Big).\end{aligned} \tag{18}$$

Therefore, the final action reward combines coarse outcome supervision with fine-grained self-judgment. This design encourages the policy to optimize not only for successful rollouts, but also for intermediate actions that are genuinely aligned with the intended strategy.

5. Experiments

5.1. Experimental Setup

Benchmarks. We evaluate StraTA on three representative benchmarks: ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and SciWorld (Wang et al., 2022), covering scenarios including embodied manipulation, web navigation, and scientific experimentation. To standardize evaluation, we serve all benchmarks through AgentGym (Xi et al., 2025b), a unified agentic RL codebase. Following GiGPO (Feng et al., 2025b) and AgentGym-RL (Xi et al., 2025c), we report success rate on ALFWorld with a detailed per-subtask breakdown, both task score and success rate on WebShop, and task score on SciWorld.

Baselines. We compare StraTA against diverse baselines. For prompting-based methods, we include the frontier closed-source models GPT-5.1 (OpenAI, 2025), Claude-4-Sonnet (Anthropic, 2025), and Gemini-2.5-Flash (Comanici et al., 2025), together with the open-source Qwen2.5 series. In addition to vanilla prompting, which generates actions directly without explicit reasoning, we include ReAct (Yao et al., 2023) as a stronger baseline that interleaves reasoning and action generation at each step. For training-based approaches, we consider standard RL methods: PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), and GiGPO (Feng et al., 2025b). PPO optimizes the policy using a critic network, whereas RLOO replaces the critic with leave-one-out reward estimation. GRPO derives learning signals from relative reward comparisons within groups. GiGPO further improves long-horizon performance by introducing anchor states for credit assignment. For SciWorld, we additionally include AgentGym-RL-7B and ScalingInter-7B (Xi et al., 2025c) as strong RL-trained baselines.

Training details. Our training procedure primarily follows GiGPO (Feng et al., 2025b), with additional specifics aligned with AgentGym-RL (Xi et al., 2025c). For our backbone models, we utilize Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct as representative architectures of different scales (Yang et al., 2024). Training is conducted for 150 steps with a batch size of 16. In StraTA, each task samples 4 strategies, and for each strategy, 8 rollouts are performed. The oversampling ratio σ is set to 8, the aggregation ratio δ to 0.5, the length penalty threshold λ to 0.5, and the self-judgment reward weight κ to 0.1. All experiments are run on a single node equipped with 8 NVIDIA H100 GPUs. Further implementation details and complete parameter specifications are available in Appendix B.3.

Table 1. Performance on ALFWorld and WebShop. Results from open-source models are averaged over 3 random seeds. For ALFWorld, we report the success rate of each subtask together with the overall success rate. For WebShop, we report the overall task score and overall success rate. The best and second best results are highlighted. †The results are from the original paper (Feng et al., 2025b).

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ
Closed-Source Model									
GPT-5.1	80.0	53.8	22.2	62.5	44.0	66.7	55.7	31.2	22.2
Claude-4-Sonnet	80.0	76.9	63.0	75.0	72.0	70.8	72.9	23.7	18.4
Gemini-2.5-Flash	71.4	30.8	40.7	18.8	20.0	25.0	38.6	4.7	4.4
Qwen2.5-1.5B-Instruct									
Vanilla	$5.7_{\pm0.0}$	$25.6_{\pm4.4}$	$1.2_{\pm2.1}$	$0.0_{\pm0.0}$	$0.0_{\pm0.0}$	$2.8_{\pm2.4}$	$4.5_{\pm0.4}$	$20.2_{\pm0.7}$	$2.9_{\pm0.8}$
ReAct	$30.5_{\pm1.3}$	$12.8_{\pm9.6}$	$7.4_{\pm3.0}$	$18.8_{\pm8.8}$	$17.3_{\pm1.9}$	$12.5_{\pm6.8}$	$17.6_{\pm2.0}$	$16.2_{\pm0.3}$	$9.1_{\pm0.4}$
PPO	$86.7_{\pm4.4}$	$61.5_{\pm7.7}$	$93.8_{\pm2.1}$	$97.9_{\pm3.6}$	$84.0_{\pm0.0}$	$80.6_{\pm2.4}$	$85.5_{\pm2.2}$	$78.2_{\pm0.4}$	$52.8_{\pm0.2}$
RLOO†	$88.3_{\pm3.0}$	$52.8_{\pm8.6}$	$71.0_{\pm5.9}$	$62.8_{\pm8.7}$	$66.4_{\pm5.5}$	$56.9_{\pm4.7}$	$69.7_{\pm2.5}$	$73.9_{\pm5.6}$	$52.1_{\pm6.7}$
GRPO	$80.0_{\pm0.0}$	$61.5_{\pm0.0}$	$81.5_{\pm3.7}$	$95.8_{\pm7.2}$	$62.7_{\pm4.6}$	$69.4_{\pm4.8}$	$75.5_{\pm2.9}$	$80.0_{\pm0.5}$	$62.2_{\pm1.2}$
GiGPO†	$94.4_{\pm5.9}$	$67.5_{\pm4.6}$	$94.8_{\pm3.8}$	$94.4_{\pm7.8}$	$79.8_{\pm4.7}$	$76.4_{\pm5.4}$	$86.7_{\pm1.7}$	$83.1_{\pm1.6}$	$65.0_{\pm3.2}$
StraTA (Ours)	$89.5_{\pm1.6}$	$87.2_{\pm4.4}$	$92.6_{\pm0.0}$	$93.8_{\pm0.0}$	$90.7_{\pm4.6}$	$90.3_{\pm2.4}$	$90.7_{\pm1.9}$	$91.1_{\pm0.3}$	$82.5_{\pm0.6}$
Qwen2.5-7B-Instruct									
Vanilla	$50.5_{\pm9.1}$	$43.6_{\pm4.4}$	$27.2_{\pm5.7}$	$14.6_{\pm7.2}$	$16.0_{\pm4.0}$	$5.6_{\pm4.8}$	$27.4_{\pm1.5}$	$31.2_{\pm1.0}$	$5.3_{\pm1.5}$
ReAct	$56.2_{\pm3.6}$	$43.6_{\pm3.6}$	$49.4_{\pm4.6}$	$43.8_{\pm5.1}$	$37.3_{\pm6.8}$	$18.1_{\pm5.2}$	$42.4_{\pm1.7}$	$35.7_{\pm0.2}$	$27.3_{\pm0.2}$
PPO	$96.2_{\pm1.7}$	$71.8_{\pm8.9}$	$96.3_{\pm0.0}$	$91.7_{\pm3.6}$	$70.7_{\pm6.1}$	$87.5_{\pm0.0}$	$87.4_{\pm1.1}$	$82.2_{\pm0.2}$	$68.9_{\pm1.0}$
RLOO†	$87.6_{\pm4.3}$	$78.2_{\pm8.3}$	$87.3_{\pm5.8}$	$81.3_{\pm7.6}$	$71.9_{\pm5.2}$	$48.9_{\pm8.4}$	$75.5_{\pm4.6}$	$80.3_{\pm3.2}$	$65.7_{\pm4.0}$
GRPO	$88.6_{\pm0.0}$	$59.0_{\pm4.4}$	$87.7_{\pm4.3}$	$79.2_{\pm7.2}$	$54.7_{\pm2.3}$	$80.6_{\pm8.7}$	$77.1_{\pm0.7}$	$79.8_{\pm0.9}$	$68.2_{\pm1.4}$
GiGPO†	$97.7_{\pm1.6}$	$82.7_{\pm7.9}$	$98.8_{\pm1.6}$	$83.7_{\pm7.2}$	$89.3_{\pm8.2}$	$79.2_{\pm6.6}$	$90.8_{\pm1.3}$	$84.4_{\pm2.9}$	$72.8_{\pm3.2}$
StraTA (Ours)	$94.3_{\pm0.0}$	$92.3_{\pm0.0}$	$97.5_{\pm2.1}$	$100.0_{\pm0.0}$	$93.3_{\pm4.6}$	$81.9_{\pm2.4}$	$93.1_{\pm0.8}$	$91.2_{\pm0.3}$	$84.2_{\pm0.3}$

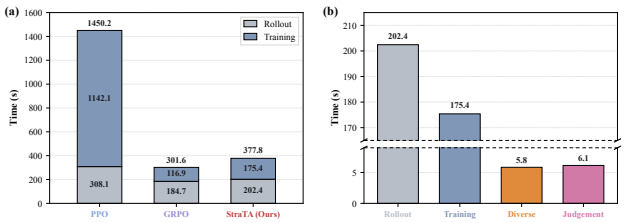

Figure 3. Computational efficiency analysis on ALFWorld. (a) Per-step wall-clock time for PPO, GRPO, and StraTA, broken down into rollout and training phases. (b) Detailed per-step breakdown for StraTA on rollout, training, and the two core techniques.

5.2. Main Results

Performance on ALFWorld and WebShop. Table 1 reports the results on ALFWorld and WebShop. We highlight three key observations. (1) Models without RL training fail to reliably solve these tasks, suggesting that specialized training is necessary for long-horizon agentic problems. Frontier closed-source models only achieve the best success rates reaching 72.9% on ALFWorld and 22.2% on WebShop. (2) RL-based methods consistently improve performance and clearly outperform frontier closed-source models. PPO achieves success rates of 87.4% and 68.9% on ALFWorld and WebShop, respectively, while GRPO and GiGPO perform even better in some settings. (3) StraTA delivers the strongest overall performance. At the 1.5B scale, StraTA reaches success rates of 90.7% on ALFWorld and 82.5% on WebShop. At the 7B scale, StraTA further improves to 93.1% and 84.2%. The training curves on WebShop in Figure 1c intuitively show the efficiency of StraTA.

Performance on SciWorld. Table 2 reports the results on SciWorld. We emphasize three observations. (1) Frontier closed-source models substantially outperform lightweight open-source models, which reflects the difficulty of SciWorld. (2) RL training can raise open-source models to the level of frontier models: ScalingInter achieves an overall score of 57.0%, and PPO and GRPO approximately reproduce this result. (3) StraTA again achieves state-of-the-art performance, reaching an overall score of 63.5% and even attaining a perfect 100.0% score on the Lifespan subset.

5.3. Ablation Studies

Effectiveness of core components. To assess the contribution of StraTA's key components, we conduct an ablation study using Qwen2.5-3B-Instruct (Yang et al., 2024) on ALFWorld and WebShop. To amplify the differences, we reduce the batch size to 4 while keeping 150 training steps. We compare StraTA with three variants: (1) *Vanilla*, the base version in Section 4.2; (2) *Diverse*, which applies only diverse strategy rollout via farthest-point sampling in Section 4.3; and (3) *Judgment*, which applies only the step-level auxiliary reward via critical self-judgment in Section 4.4. Table 3 reports the results. The *Vanilla* variant already performs strongly. The *Diverse* variant substantially improves

Table 2. Performance on SciWorld. Results from open-source models are averaged over 3 random seeds. We report the score on each subtask and the overall score. The best and second best results are highlighted. †The results are from the original paper (Xi et al., 2025c).

Method	Category	SciWorld					
		Measure	Test-Cond	Find	Chem-Mix	Lifespan	All
Closed-Source Model							
GPT-5.1	Prompting-Based	32.1	46.5	41.8	69.6	38.3	43.0
Claude-4-Sonnet	Prompting-Based	45.4	54.7	60.3	45.6	90.0	57.4
Gemini-2.5-Flash	Prompting-Based	40.7	51.3	53.7	33.0	50.0	49.6
Qwen2.5-7B-Instruct							
AgentGym-RL†	Training-Based	24.5	59.0	65.9	0.0	66.7	50.5
ScalingInter†	Training-Based	34.0	55.4	88.6	0.0	73.3	57.0
Vanilla	Prompting-Based	$2.1_{\pm 1.5}$	$12.3_{\pm 1.3}$	$21.2_{\pm 1.6}$	$9.6_{\pm 3.3}$	$25.0_{\pm 2.4}$	$13.9_{\pm 0.8}$
ReAct	Prompting-Based	$12.2_{\pm 2.1}$	$14.4_{\pm 1.2}$	$33.6_{\pm 2.7}$	$9.6_{\pm 2.8}$	$28.9_{\pm 3.4}$	$20.0_{\pm 0.3}$
PPO	Training-Based	$1.8_{\pm 1.3}$	$57.5_{\pm 0.2}$	$60.9_{\pm 0.8}$	$6.9_{\pm 1.9}$	$97.8_{\pm 3.1}$	$51.4_{\pm 0.3}$
GRPO	Training-Based	$31.2_{\pm 4.8}$	$49.8_{\pm 2.9}$	$32.7_{\pm 3.6}$	$14.7_{\pm 5.4}$	$53.3_{\pm 5.0}$	$41.8_{\pm 2.2}$
StraTA (Ours)	Training-Based	$55.7_{\pm 3.2}$	$54.4_{\pm 0.7}$	$78.6_{\pm 1.1}$	$16.3_{\pm 1.2}$	$100.0_{\pm 0.0}$	$63.5_{\pm 0.7}$

Table 3. Performance comparison across different variants. *Vanilla* means no trick is applied. *Diverse* represents the version with diverse strategy rollout via farthest point sampling. *Judgment* represents the version with step-level auxillary reward via critical self-judgment.

Variant	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ
Qwen2.5-3B-Instruct									
Vanilla	$87.6_{\pm 2.7}$	$69.2_{\pm 6.3}$	$88.9_{\pm 0.0}$	$66.7_{\pm 5.9}$	$70.7_{\pm 1.9}$	$77.8_{\pm 2.0}$	$79.0_{\pm 2.2}$	$76.0_{\pm 0.2}$	$64.0_{\pm 0.4}$
Diverse	$92.4_{\pm 1.3}$	$84.6_{\pm 0.0}$	$96.3_{\pm 0.0}$	$85.4_{\pm 2.9}$	$78.7_{\pm 1.9}$	$84.7_{\pm 2.0}$	$87.9_{\pm 0.6}$	$75.9_{\pm 0.6}$	$64.6_{\pm 0.2}$
Judgment	$88.6_{\pm 4.7}$	$64.1_{\pm 7.3}$	$92.6_{\pm 5.2}$	$81.3_{\pm 0.0}$	$68.0_{\pm 3.3}$	$84.7_{\pm 3.9}$	$81.9_{\pm 3.2}$	$79.3_{\pm 0.2}$	$66.7_{\pm 0.5}$
StraTA (Ours)	$98.1_{\pm 1.7}$	$71.8_{\pm 8.9}$	$93.8_{\pm 2.1}$	$97.9_{\pm 3.6}$	$72.0_{\pm 0.0}$	$88.9_{\pm 2.4}$	$88.6_{\pm 1.9}$	$85.9_{\pm 0.5}$	$73.4_{\pm 1.0}$

ALFWorld performance to 87.9%, while the *Judgment* variant improves WebShop performance to 66.7%. Figure 4 intuitively demonstrates the convergence speed of different variants. These results indicate that the two components accelerate learning and improve performance in a largely orthogonal manner, jointly enabling StraTA's strong results on long-horizon agentic tasks.

Analysis on computational efficiency. We analyze the computational overhead of StraTA from both theoretical and empirical perspectives. For a given task, StraTA samples N strategies and performs M rollouts for each strategy over a horizon of H, resulting in $N \times M \times H$ action rollouts and $N \times M$ judgment rollouts. When the latency of each rollout is comparable and H is sufficiently large, the total computation is comparable to that of GRPO with a group size of $N \times M$. In addition, the overhead introduced by diverse strategy rollout and critical self-judgment accounts for only $1/H$ of the main action rollouts. Empirically, StraTA is only marginally slower than GRPO during both rollout and training. Detailed wall-clock measurements and a stage-level cost breakdown are provided in Figure 3.

6. Conclusion

Long-horizon agentic RL remains difficult, which is largely due to the fact that purely reactive decision making weakens both exploration and credit assignment. To address this challenge, we introduce StraTA, a simple framework that equips agentic RL with explicit trajectory-level strategy. By generating a compact strategy from the initial task state and conditioning subsequent actions on it, StraTA separates high-level planning from low-level execution. Building on this formulation, hierarchical GRPO-style rollouts, diverse strategy rollout, and critical self-judgment jointly improve exploration and learning. Experiments on ALFWorld, WebShop, and SciWorld show that StraTA consistently improves both sample efficiency and final performance over strong prompting-based and RL baselines, underscoring the value of trajectory-level abstraction for long-horizon agents. StraTA nevertheless has limitations: its effectiveness depends on the quality of the generated strategy, and a fixed strategy may become restrictive when the environment changes substantially during an episode. Promising directions for future work include adaptive strategy revision, richer strategy representations, and extending the framework to broader agentic domains.

Impact Statement

This work aims to improve the training of LLM-based agents for long-horizon interactive tasks. The proposed method introduces explicit trajectory-level strategies and finer-grained credit assignment, which may improve the reliability and

sample efficiency of agentic systems. These advances may benefit applications that require robust decision making in simulated or controlled environments.

Our experiments are conducted entirely in established simulated benchmarks (ALFWorld, WebShop, and SciWorld), and do not involve real-world deployment. We encourage future applications to incorporate human oversight, task-appropriate constraints, and thorough evaluations of reliability and robustness in the target setting.

References

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Anthropic. Introducing claude 4, 2025. URL https://www.anthropic.com/news/claude-4.

Azar, M. G., Guo, Z. D., Piot, B., Munos, R., Rowland, M., Valko, M., and Calandriello, D. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*, 2024.

Chen, A., Li, A., Gong, B., Jiang, B., Fei, B., Yang, B., Shan, B., Yu, C., Wang, C., Zhu, C., et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025a.

Chen, T., Chen, Z., Chen, B., Cai, Z., Liu, Y., Li, Z., Liang, Q., Lin, X., Ge, Y., Gu, Z., et al. Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. *arXiv preprint arXiv:2506.18088*, 2025b.

Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

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.

DeepSeek-AI, 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.

Du, Y., Li, S., Torralba, A., Tenenbaum, J. B., and Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. In *Forty-first international conference on machine learning*, 2024.

Erdogan, L. E., Lee, N., Kim, S., Moon, S., Furuta, H., Anumanchipalli, G., Keutzer, K., and Gholami, A. Plan-and-act: Improving planning of agents for long-horizon tasks. *arXiv preprint arXiv:2503.09572*, 2025.

Ethayarajh, K., Xu, W., Muennighoff, N., Jurafsky, D., and Kiela, D. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

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. *arXiv preprint arXiv:2505.03792*, 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.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., et al. A survey on llm-as-a-judge. *The Innovation*, 2024.

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.

Hong, S., Zhuge, M., Chen, J., Zheng, X., Cheng, Y., Wang, J., Zhang, C., Wang, Z., Yau, S. K. S., Lin, Z., et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*, 2024.

Hu, J., Liu, J. K., Xu, H., and Shen, W. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 2025.

Ji, Y., Ma, Z., Wang, Y., Chen, G., Chu, X., and Wu, L. Tree search for llm agent reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Li, D., Jiang, B., Huang, L., Beigi, A., Zhao, C., Tan, Z., Bhattacharjee, A., Jiang, Y., Chen, C., Wu, T., et al. From generation to judgment: Opportunities and challenges of llm-as-a-judge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 2757–2791, 2025.

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, X., Wang, K., Wu, Y., Huang, F., Li, Y., Zhang, J., and Jiao, J. Agentic reinforcement learning with implicit step rewards. In *The Fourteenth International Conference on Learning Representations*, 2026.

Liu, Z., Chen, C., Li, W., Qi, P., Pang, T., Du, C., Lee, W. S., and Lin, M. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The ai scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024.

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*, 2025.

OpenAI. Gpt-5.1: A smarter, more conversational chatgpt, 2025. URL https://openai.com/index/gpt-5-1.

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 *Advances in Neural Information Processing Systems*, 2022.

Peng, J., Liu, Y., Zhou, R., Fleming, C., Wang, Z., Garcia, A., and Hong, M. Hiper: Hierarchical reinforcement learning with explicit credit assignment for large language model agents. *arXiv preprint arXiv:2602.16165*, 2026.

Prabhudesai, M., Chen, L., Ippoliti, A., Fragkiadaki, K., Liu, H., and Pathak, D. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

Putta, P., Mills, E., Garg, N., Motwani, S., Finn, C., Garg, D., and Rafailov, R. Agent q: Advanced reasoning and learning for autonomous ai agents. *arXiv preprint arXiv:2408.07199*, 2024.

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 *Advances in Neural Information Processing Systems*, 2023.

Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, 2019.

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 *Advances in Neural Information Processing Systems*, 2023.

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.

Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., and Yao, S. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 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 *International Conference on Learning Representations*, 2021.

Song, Y., Xiong, W., Zhao, X., Zhu, D., Wu, W., Wang, K., Li, C., Peng, W., and Li, S. Agentbank: Towards generalized llm agents via fine-tuning on 50000+ interaction trajectories. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024.

Tan, S., Luo, M., Cai, C., Venkat, T., Montgomery, K., Hao, A., Wu, T., Balyan, A., Roongta, M., Wang, C., Li, L. E., Popa, R. A., and Stoica, I. rllm: A framework for post-training language agents, 2025. URL https://pretty-radio-b75.notion.site/rLLM-A-Framework-for-Post-Training-Language-Agents-21b81902c146819db63cd98a54ba5f31.

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, 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*, 2024a.

Wang, R., Jansen, P., Côté, M.-A., and Ammanabrolu, P. Scienceworld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022.

Wang, X., Chen, Y., Yuan, L., Zhang, Y., Li, Y., Peng, H., and Ji, H. Executable code actions elicit better llm agents. In *Forty-first International Conference on Machine Learning*, 2024b.

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*, 2025.

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., Jiang, L., Zhang, X., Zhang, S., Liu, J., et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

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*, 2025a.

Xi, Z., Ding, Y., Chen, W., Hong, B., Guo, H., Wang, J., Guo, X., Yang, D., Liao, C., He, W., et al. Agentgym: Evaluating and training large language model-based agents across diverse environments. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025b.

Xi, Z., Huang, J., Liao, C., Huang, B., Guo, H., Liu, J., Zheng, R., Ye, J., Zhang, J., Chen, W., et al. Agentgym-rl: Training llm agents for long-horizon decision making through multi-turn reinforcement learning. *arXiv preprint arXiv:2509.08755*, 2025c.

Xie, T., Zhang, D., Chen, J., Li, X., Zhao, S., Cao, R., Hua, T. J., Cheng, Z., Shi, D., Liu, F., Yu, Z., Ma, B., Li, G., Uddin, A., Yu, H., Zhang, R., Chen, X., Zhou, V., Shi, S., Zhu, M., Fan, L., Li, J., Savarese, S., Xiong, C., Yu, T., and Yang, D. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. In *Advances in Neural Information Processing Systems*, 2024.

Xu, W., Zhou, Y., Zhou, Y., Cao, Q., Li, S., Bu, J., Liu, B., Chen, Y., He, X., Zhao, X., et al. Probing scientific general intelligence of llms with scientist-aligned workflows. *arXiv preprint arXiv:2512.16969*, 2025.

Xue, X., Zhou, Y., Zhang, G., Zhang, Z., Li, Y., Zhang, C., Yin, Z., Torr, P., Ouyang, W., and Bai, L. Comas: Co-evolving multi-agent systems via interaction rewards. *arXiv preprint arXiv:2510.08529*, 2025.

Yang, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Li, C., Liu, D., Huang, F., Wei, H., et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Yao, S., Chen, H., Yang, J., and Narasimhan, K. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, 2022.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Yeo, E., Tong, Y., Niu, M., Neubig, G., and Yue, X. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.

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*, 2025.

Yuan, W., Pang, R. Y., Cho, K., Li, X., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024.

Zeng, A., Liu, M., Lu, R., Wang, B., Liu, X., Dong, Y., and Tang, J. Agenttuning: Enabling generalized agent abilities for llms. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024.

Zhai, Y., Bai, H., Lin, Z., Pan, J., Tong, S., 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. *Advances in neural information processing systems*, 2024.

Zhang, G., Geng, H., Yu, X., Yin, Z., Zhang, Z., Tan, Z., Zhou, H., Li, Z., Xue, X., Li, Y., et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025a.

Zhang, Q., Wu, H., Zhang, C., Zhao, P., and Bian, Y. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *arXiv preprint arXiv:2504.05812*, 2025b.

Zhao, X., Kang, Z., Feng, A., Levine, S., and Song, D. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

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*, 2025.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, 2023.

Zhou, S., Xu, F. F., Zhu, H., Zhou, X., Lo, R., Sridhar, A., Cheng, X., Bisk, Y., Fried, D., Alon, U., et al. WebArena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*, 2024a.

Zhou, Y., Zanette, A., Pan, J., Levine, S., and Kumar, A. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024b.

Zhuge, M., Wang, W., Kirsch, L., Faccio, F., Khizbullin, D., and Schmidhuber, J. Gptswarm: Language agents as optimizable graphs. In *Forty-first International Conference on Machine Learning*, 2024.

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.

Zuo, Y., Zhang, K., Sheng, L., Qu, S., Cui, G., Zhu, X., Li, H., Zhang, Y., Long, X., Hua, E., et al. Ttrl: Test-time reinforcement learning. *arXiv preprint arXiv:2504.16084*, 2025.

A. Additional Experimental Results

A.1. Effectiveness of Core Components

Section 5.3 examines the effectiveness of StraTA's core components, and Table 3 reports the numerical results for different variants. In this section, we present their training curves on ALFWorld (Shridhar et al., 2021) in Figure 4. The *Vanilla* variant already substantially outperforms the GRPO (Shao et al., 2024) baseline, and adding the *Diverse* and *Judgment* techniques further accelerates convergence. Overall, StraTA with both techniques achieves the most stable and efficient training.

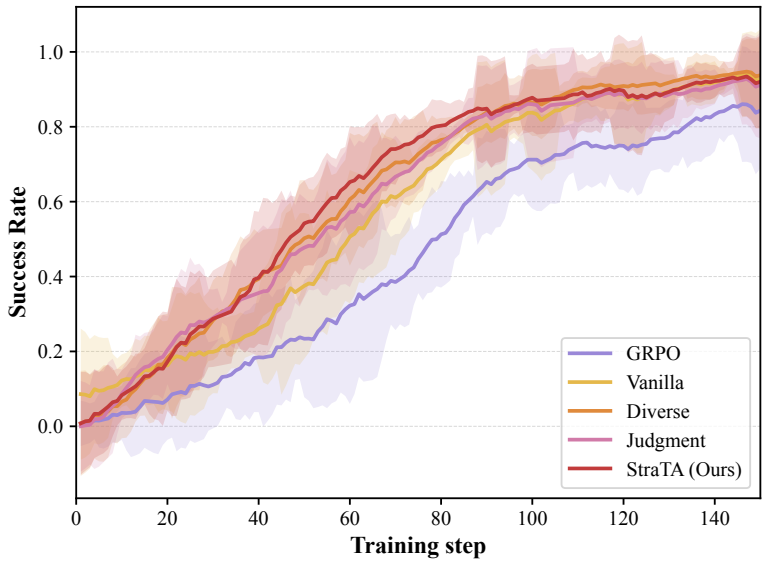

Figure 4. Training curves on ALFWorld for the ablation study on the effectiveness of core components. We compare StraTA with its ablated variants to evaluate the contribution of each component.

A.2. Impact of Training Parameters

In this section, we further examine the impact of key hyperparameters, including the oversampling ratio σ, aggregation ratio δ, and self-judgment reward weight κ. Following Section 5.3, we conduct ablation studies using Qwen2.5-3B-Instruct (Yang et al., 2024) on ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022). The model is trained for 150 steps with a batch size of 4, while all other settings remain consistent with the main experiments.

Table 4 reports the performance of StraTA with σ set to 1, 2, 4, and 8. As σ increases, performance improves steadily. This trend suggests that diverse strategy rollout effectively promotes exploration in the strategy space and yields more informative and discriminative reward signals.

Table 5 summarizes the results for δ values of 0.1, 0.5, and 1.0. When δ is set to 0.1, too few trajectories are used to reliably estimate strategy quality, resulting in poor performance. In contrast, setting δ to 1.0 ignores potential execution errors during action selection, leading to overly pessimistic evaluations. Setting δ to 0.5, as used in the main experiments, achieves the best performance.

Table 6 shows the performance for κ values of 0.01, 0.1, and 1.0. A small weight of 0.01 underutilizes the benefits of critical self-judgment, leading to degraded performance. Conversely, a large weight of 1.0 places excessive reliance on self-judgment quality, which further harms performance. In general, setting κ to 0.1, adopted in the main experiments, provides the most effective trade-off.

Table 4. Performance of StraTA on ALFWorld and WebShop across varying oversampling ratios σ. We sweep the values of 1, 2, 4, and 8 to evaluate its impact on performance.

Parameter	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ
Qwen2.5-3B-Instruct									
$\sigma = 1$	$88.6_{\pm 4.7}$	$64.1_{\pm 7.3}$	$92.6_{\pm 5.2}$	$81.3_{\pm 0.0}$	$68.0_{\pm 3.3}$	$84.7_{\pm 3.9}$	$81.9_{\pm 3.2}$	$79.3_{\pm 0.2}$	$66.7_{\pm 0.5}$
$\sigma = 2$	$93.3_{\pm 1.4}$	$82.1_{\pm 3.6}$	$87.7_{\pm 3.5}$	$87.5_{\pm 0.0}$	$70.7_{\pm 3.8}$	$73.6_{\pm 2.0}$	$83.1_{\pm 0.3}$	$73.9_{\pm 0.6}$	$56.6_{\pm 0.8}$
$\sigma = 4$	$92.4_{\pm 1.3}$	$61.5_{\pm 0.0}$	$96.3_{\pm 3.0}$	$89.6_{\pm 5.9}$	$66.7_{\pm 5.0}$	$79.2_{\pm 3.4}$	$83.1_{\pm 0.9}$	$77.9_{\pm 0.9}$	$69.1_{\pm 0.8}$
$\sigma = 8$	$98.1_{\pm 1.7}$	$71.8_{\pm 8.9}$	$93.8_{\pm 2.1}$	$97.9_{\pm 3.6}$	$72.0_{\pm 0.0}$	$88.9_{\pm 2.4}$	$88.6_{\pm 1.9}$	$85.9_{\pm 0.5}$	$73.4_{\pm 1.0}$

Table 5. Performance of StraTA on ALFWorld and WebShop across varying aggregation ratios δ. We sweep the values of 0.1, 0.5, and 1.0 to evaluate its impact on performance.

Parameter	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ
Qwen2.5-3B-Instruct									
$\delta = 0.1$	$89.5_{\pm 2.7}$	$71.8_{\pm 3.6}$	$82.7_{\pm 3.5}$	$50.0_{\pm 17.7}$	$34.7_{\pm 5.0}$	$81.9_{\pm 2.0}$	$71.0_{\pm 0.7}$	$79.3_{\pm 0.3}$	$66.1_{\pm 0.3}$
$\delta = 0.5$	$98.1_{\pm 1.7}$	$71.8_{\pm 8.9}$	$93.8_{\pm 2.1}$	$97.9_{\pm 3.6}$	$72.0_{\pm 0.0}$	$88.9_{\pm 2.4}$	$88.6_{\pm 1.9}$	$85.9_{\pm 0.5}$	$73.4_{\pm 1.0}$
$\delta = 1.0$	$92.4_{\pm 1.4}$	$69.2_{\pm 6.3}$	$92.6_{\pm 3.0}$	$85.4_{\pm 5.9}$	$82.7_{\pm 1.9}$	$76.4_{\pm 2.0}$	$85.0_{\pm 0.6}$	$85.2_{\pm 0.7}$	$71.3_{\pm 1.5}$

B. Additional Implementation Details

B.1. Environmental Setup

All benchmark environments are accessed through AgentGym (Xi et al., 2025b), which provides a unified interface for both training and evaluation. This setup standardizes the observation format, action interface, interaction loop, and reward computation across ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and SciWorld (Wang et al., 2022).

ALFWorld. ALFWorld (Shridhar et al., 2021) is a text-based embodied household environment in which an agent interacts with simulated home scenes through natural-language actions. Each task requires the agent to complete a goal-oriented household instruction, such as picking up an object, examining an object under a specified condition, cleaning an object, heating or cooling an object, or placing two objects in the same location. We report the overall success rate as well as the success rates for six task types: Pick, Look, Clean, Heat, Cool, and Pick2. Each episode allows up to 50 environment steps. During training, the environment provides a rule-based, step-decayed reward, with a maximum value of 1 for successful completion and 0 for failure. This design preserves the binary success signal while assigning larger rewards to trajectories that complete the task in fewer steps. We also apply a format penalty of -1 to malformed responses that cannot be parsed into the required action format.

WebShop. WebShop (Yao et al., 2022) is a web-based shopping environment that evaluates an agent's ability to complete realistic purchasing tasks through multi-step web interaction. Given a user instruction, the agent must search for relevant products, inspect product pages, compare attributes such as price and specifications, and select an item that satisfies the task requirements. Each episode is limited to 15 environment steps. Following the standard evaluation protocol, we report both the task score and the success rate. The task score measures partial progress toward the target product, whereas the success rate indicates whether the final selected product fully satisfies the user instruction. During training, the environment provides a rule-based, step-decayed reward, with a maximum value of 1 for success and 0 for failure. This design preserves the binary success signal while assigning larger rewards to trajectories that complete the task in fewer steps. A format penalty of -1 is applied to malformed responses that cannot be parsed into the required action format.

SciWorld. SciWorld (Wang et al., 2022) is a text-based scientific experimentation environment that requires the agent to perform procedural reasoning and execute actions in simulated science tasks. The benchmark covers multiple task categories, including but not limited to measurement, conditional testing, object finding, chemical mixing, and lifespan reasoning. Following the standard benchmark setting, we report the normalized task score as the primary evaluation metric. Each episode allows up to 20 environment steps. During training, the reward is computed from the final task score, normalized to the range $[0, 1]$, and further adjusted by a step-decay factor so that more efficient task completion receives a higher reward. As in the other environments, malformed responses that cannot be parsed into the required action format receive a format penalty of -1.

Table 6. Performance of StraTA on ALFWorld and WebShop across varying self-judgment reward weights κ. We sweep the values of 0.01, 0.1, and 1.0 to evaluate its impact on performance.

Parameter	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ
Qwen2.5-3B-Instruct									
$\kappa = 0.01$	$94.3_{\pm 0.0}$	$71.8_{\pm 3.6}$	$88.9_{\pm 0.0}$	$87.5_{\pm 0.0}$	$72.0_{\pm 0.0}$	$83.3_{\pm 0.0}$	$84.5_{\pm 0.3}$	$81.0_{\pm 0.4}$	$73.3_{\pm 0.8}$
$\kappa = 0.1$	$98.1_{\pm 1.7}$	$71.8_{\pm 8.9}$	$93.8_{\pm 2.1}$	$97.9_{\pm 3.6}$	$72.0_{\pm 0.0}$	$88.9_{\pm 2.4}$	$88.6_{\pm 1.9}$	$85.9_{\pm 0.5}$	$73.4_{\pm 1.0}$
$\kappa = 1.0$	$94.3_{\pm 0.0}$	$61.5_{\pm 0.0}$	$96.3_{\pm 0.0}$	$93.8_{\pm 5.1}$	$76.0_{\pm 0.0}$	$87.5_{\pm 0.0}$	$87.1_{\pm 0.6}$	$78.9_{\pm 0.8}$	$61.8_{\pm 1.0}$

B.2. Prompt Templates

Section 4 introduces the StraTA pipeline, and Figure 2 illustrates the sampling process. For reproducibility, we provide the prompt templates used for global strategy generation, strategy-guided action selection, and critical self-judgment in this section.

A Template Prompt for Global Strategy Generation

```
{description}

{context}

Before you start to work on the task, you should propose a global strategy. The requirements are presented as follows:
1. The strategy is a constant text snippet, which guides your future actions throughout the whole episode to achieve the goal.
2. The strategy should be concrete enough, so that future actions can be taken by strictly following it.
3. The strategy should be practical enough, so that future actions can be taken based on existing information.

First think step by step to come up with an overall planning. Then propose your strategy enclosed within <strategy>...</strategy>
↪ tags. For example:
According to the rules, any non-random strategy will lead to sub-optimal outcomes.
<strategy>Select one of the actions uniformly at random at each step.</strategy>
```

A Template Prompt for Strategy-guided Action Taking

```
{description}

You should strictly observe the following strategy to take actions:
{strategy}

{context}

{instruction}
```

A Template Prompt for Critical Self-judgment

```
{description}

The proposed global strategy for the task is presented as follows:
{strategy}

The complete history of actions taken is presented as follows:
{history}

You should point out all the problematic steps. A step is said to be problematic if it neither follows the global strategy nor pushes
↪ forward the task progress.

First think step by step to analyze the history. Then provide your judgment enclosed within <judgment>...</judgment> tags. The
↪ content should be a list of step numbers corresponding to the given history. If all the steps are reasonable, yield an empty
↪ list. For example:
```

Step 1 seems to deviate from the strategy, but it effectively facilitates future exploration. Step 2 and 5 fail to follow the strategy
↪ and make no contribution to the final success.
<judgment>[2, 5]</judgment>

B.3. Training Parameters

In this section, we provide additional training details that complement the setup described in Section 5.1. Our implementation of StraTA is mainly built on the rLLM framework (Tan et al., 2025). The complete parameter settings are summarized in Table 7.

Table 7. The detailed parameter settings when implementing our StraTA framework.

Parameter	Setting
Base model	Qwen2.5-7B-Instruct
Number of strategies per task N	4
Number of rollouts per strategy M	8
Oversampling ratio σ	8
Aggregation ratio δ	0.5
Length penalty threshold λ	0.5
Self-judgment reward weight κ	0.1
Embedding model	MiniLM-L6
Token limit for prompts	7168
Token limit for responses	1024
Training temperature	1.0
Evaluation temperature	0.7
Evaluation top-p	0.8
Evaluation top-k	20
Clipping ϵ_{low}	0.2
Clipping ϵ_{high}	0.28
Optimizer	AdamW
Learning rate	1e-6
Weight decay	0.01
Mini batch size	1024
Number of training steps	150

B.4. Diverse Strategy Rollout via Farthest Point Sampling

Section 4.3 introduces diverse strategy rollout, in which farthest point sampling in semantic space is used to select diverse strategies. Here, we provide additional details of this technique. Algorithm 1 summarizes the procedure. The overall complexity is $O(\sigma N^2 d)$, where d denotes the dimensionality of the semantic embedding. In practice, selection takes only a few milliseconds per task, which is negligible relative to the cost of LLM rollouts, as further confirmed in Section 5.3.

C. Case Studies

To further illustrate the behavior of StraTA, we provide case studies in this section, including one example each for ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and SciWorld (Wang et al., 2022). All the examples are randomly selected from the evaluation history of the models trained in the main experiments.

Algorithm 1 Diverse Strategy Selection via Farthest Point Sampling

Require: Oversampled candidates $\mathcal{C} = \{z^i\}_{i=1}^{\sigma N}$ and semantic encoder $\phi(\cdot)$
Ensure: Diverse strategies $\mathcal{Z} = \{z^i\}_{i=1}^{N}$

```
for each candidate strategy z^i ∈ C do
    e^i ← φ(z^i)
    e^i ← e^i / ||e^i||_2
end for
ē ← (1/|C|) Σ_{z^i ∈ C} e^i
z* ← arg max_{z^i ∈ C} (e^i)^⊤ ē
Z ← {z*}
while |Z| < N do
    z_next ← arg min_{z^i ∈ C \ Z} max_{z^j ∈ Z} (e^i)^⊤ e^j
    Z ← Z ∪ {z_next}
end while
return Z
```

C.1. ALFWorld

Environment (Initial Observation)

Task: put a tissuebox in sidetable.
You are in the middle of a room. Looking quickly around you, you see an armchair 1, a coffeetable 1, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a sidetable 3, a sidetable 2, a sidetable 1, and a sofa 1.

Agent (Strategy Generation)

<strategy>Explore the room to find where the tissuebox might be and how to get it to the sidetable.</strategy>

Environment (Step 1)

You are in the middle of a room. Looking quickly around you, you see an armchair 1, a coffeetable 1, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a dresser 1, a garbagecan 1, a sidetable 3, a sidetable 2, a sidetable 1, and a sofa 1.

Agent (Step 1)

<action>go to coffeetable 1</action>

Environment (Step 2)

On the coffeetable 1, you see a box 1, a cellphone 3, a pencil 1, a statue 1, a tissuebox 2, and a tissuebox 1.

Agent (Step 2)

<action>take tissuebox 1 from coffeetable 1</action>

Environment (Step 3)

You pick up the tissuebox 1 from the coffeetable 1.

Agent (Step 3)

<action>go to sidetable 1</action>

Environment (Step 4)

On the sidetable 1, you see a cellphone 2 and a houseplant 1.

Agent (Step 4)

<action>put tissuebox 1 in/on sidetable 1</action>

Environment (Final Feedback)

You put the tissuebox 1 in/on the sidetable 1. The task is completed successfully.

C.2. WebShop

Environment (Initial Observation)

Task: Find me high density, ready use, twin size, long lasting, assembly required mattresses with box spring, and price lower than 550.00 dollars.

Agent (Strategy Generation)

<strategy>Start with a generic search query to identify available products fitting the criteria. Then, refine the search based on user feedback to find the best option that meets the specifications.</strategy>

Environment (Step 1)

Search

Agent (Step 1)

<action>search[High density, ready use, twin size, long lasting, assembly required mattresses with box spring, price lower than 550]</action>

Environment (Step 2)

Back to Search | Page 1 (Total results: 50) | Next >
B078PKLZFB | Mattress Solution, 13-Inch Soft Foam Encased Hybrid Eurotop Pillowtop Innerspring Mattress And Split Wood Traditional Box Spring/Foundation Set With Frame, Twin Size 74" x 38" | $520.0
B07SXLVPF3 | Mayton 12-Inch King Size Mattress, Box Spring And Bed Frame - Foam Encased Soft Pillow Top Hybrid Contouring Comfort, No Assembly Required 78x79 | $884.03
B01A0L5FXA | Spring Coil 1 Mattress, Queen | $565.71
B09H3N5P74 | Foamma 3" x 24" x 72" Mattress for RV with Water Resistant Organic Cotton Cover, Firm High Density Foam, USA Made, CertiPUR-US Certified Foam | $149.99

Agent (Step 2)

<action>click[b078pklzfb]</action>

Environment (Step 3)

Back to Search | < Prev
Mattress Solution, 13-Inch Soft Foam Encased Hybrid Eurotop Pillowtop Innerspring Mattress And Split Wood Traditional Box Spring/Foundation Set With Frame, Twin Size 74" x 38"
Price: $520.0 | Rating: N.A. | Description | Features | Reviews | Buy Now

Agent (Step 3)

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

Environment (Final Feedback)

Thank you for shopping with us! | Your code: | ... | Your score (min 0.0, max 1.0) | 1.0 | Reward Details | None
The task is completed successfully with a score of 1.0.

C.3. SciWorld

Environment (Initial Observation)

Task: Your task is to find the animal with the longest life span. The animals are in the 'outside' location. Focus on the animal with the longest life span.
This room is called the living room. In it, you see:
the agent
a substance called air
a chair. On the chair is: nothing.
a couch. On the couch is: a white pillow.
a desk. On the desk is: a drawer.
a painting
You also see:
A door to the hallway (that is closed)

Agent (Strategy Generation)

<strategy>To achieve this goal, the first step is to open the hallway door to gain visibility of the hallway. This can be done by moving towards and then opening the hallway door. After observing this new environment, you can then identify and compare the life spans of any animals you encounter outside.</strategy>

Environment (Step 1)

This room is called the living room. In it, you see:
the agent
a substance called air
a chair. On the chair is: nothing.
a couch. On the couch is: a white pillow.
a desk. On the desk is: a drawer.
a painting
You also see:
A door to the hallway (that is closed)

Agent (Step 1)

<action>open("door to hallway")</action>

Environment (Step 2)

The door is now open.

Agent (Step 2)

<action>goto("hallway")</action>

Environment (Step 3)

You move to the hallway.

Agent (Step 3)

<action>lookaround()</action>

Environment (Step 4)

This room is called the hallway. In it, you see:
the agent
a substance called air
a finger painting
You also see:
A door to the art studio (that is closed)
A door to the bedroom (that is closed)
A door to the greenhouse (that is closed)
A door to the kitchen (that is closed)
A door to the living room (that is open)
A door to the workshop (that is closed)

Agent (Step 4)

<action>open("door to greenhouse")</action>

Environment (Step 5)

The door is now open.

Agent (Step 5)

<action>goto("greenhouse")</action>

Environment (Step 6)

You move to the greenhouse.

Agent (Step 6)

<action>lookaround()</action>

Environment (Step 7)

This room is called the greenhouse. In it, you see:
the agent
a substance called air
a bee hive. The bee hive door is closed.
a jug (containing nothing)
a sink, which is turned off. In the sink is: nothing.
You also see:
A door to the hallway (that is open)
A door to the outside (that is closed)

Agent (Step 7)

<action>open("door to outside")</action>

Environment (Step 8)

The door is now open.

Agent (Step 8)

<action>goto("outside")</action>

Environment (Step 9)

You move to the outside.

Agent (Step 9)

<action>lookaround()</action>

Environment (Step 10)

This outside location is called the outside. Here you see:
the agent
a substance called air
an axe
a frog egg
a baby elephant
a fire pit (containing nothing)
a fountain (containing a substance called water)
the ground
a baby hedgehog
a shovel
a substance called wood
You also see:
A door to the foundry (that is closed)
A door to the greenhouse (that is open)
A door to the kitchen (that is closed)

Agent (Step 10)

<action>focus("baby elephant")</action>

Environment (Final Feedback)

The task is completed successfully.