

🤖 STARK: Strategic Advantage Reinforcement with Confidence-Driven Exploration for Long-Horizon Agentic Learning

Jinyang Wu¹ *[†], Xinan Chang^{1*}, Yuhao Shen², Zhengxi Lu²,
Shuo Yang¹, Mingkuan Feng¹, Jiangyan Yi¹, Shuai Zhang¹,
Zheng Lian³, Zhengqi Wen¹, Jianhua Tao^{1‡}

¹Tsinghua University ²Zhejiang University ³Tongji University
wu-jy23@mails.tsinghua.edu.cn

Abstract

Reinforcement learning (RL) for long-horizon agents faces two key challenges. First, parallel linear rollouts tend to converge toward homogeneous reasoning patterns, leading to cognitive-mode collapse. Second, existing methods refine credit assignment along only a single dimension, making it difficult to simultaneously achieve state-aligned comparison and meta-reasoning discrimination. We propose **STARK** (Strategic and Tactical Advantage Reinforcement via Confidence-Guided Keypoint Exploration) to resolve both issues. STARK performs corrective exploration through confidence-driven branching at uncertain decision points and injects reflection into new branches. To empower these keypoint-expanded trajectories, STARK introduces a dual-track strategic and tactical advantage estimator: The strategic track compares reasoning modes under shared environmental observations, while the tactical track evaluates execution quality within each meta-reasoning tag group. Extensive experiments on ALF-World, ScienceWorld, and WebShop show that STARK consistently outperforms strong baselines across model scales and task domains, and enables a 1.5B model to surpass GPT-5 on the out-of-domain settings.

1 Introduction

Reinforcement learning (RL) has proven highly effective in enhancing LLM reasoning, achieving landmark progress in mathematics (Guo et al., 2025; OpenAI, 2025; Team et al., 2026) and code generation (Wang et al., 2026). Building on this success, the focus of RL for language models is increasingly shifting from static reasoning tasks toward agentic RL, where models are trained to interact with dynamic environments and execute

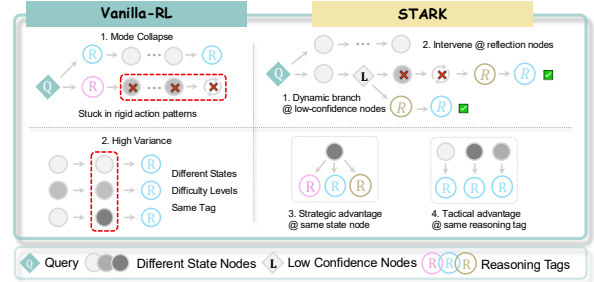


Figure 1: **Paradigm comparison.** Vanilla-RL (Left) relies on passive linear rollouts, which can lead to rigid action patterns and high-variance credit assignment across divergent states. STARK (Right) reallocates the same rollout budget to low-confidence nodes, applies reflection-guided intervention, and computes strategic and tactical advantages over aligned comparison groups.

long-horizon tasks (Zhang et al., 2025; Wu et al., 2026b). Unlike self-contained reasoning tasks, long-horizon agentic settings such as embodied planning (Shridhar et al., 2020) and web navigation (Yao et al., 2022) require the agent to maintain coherent multi-step decision-making across dozens of environmental interactions before receiving terminal reward signals. This makes both exploration and credit assignment substantially more difficult than in conventional RL settings.

The first challenge arises from **redundant rollout sampling**. Conventional RL methods usually generate multiple linear rollouts from the same initial context state, assuming that repeated sampling will expose diverse strategies and provide useful learning signals. However, such rollouts often trace similar high-probability sampling paths, leading different samples to converge toward homogeneous reasoning and action patterns. Although occasionally sampled successful trajectories may still provide informative supervision, a large fraction of rollouts merely reproduce routine steps with limited exploratory value. Thus, nominal exploration gradually collapses into exploitation, and eventually into *cognitive mode collapse*: the agent

* Equal contribution.

[†] Project Lead.

[‡] Corresponding Author.

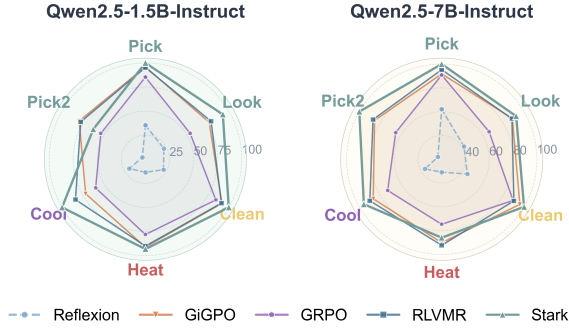


Figure 2: **Performance comparison overview.** Radar charts of per-category success rates on ALFWorld at 1.5B and 7B scales. STARK consistently dominates across all task types and model sizes.

spends substantial computation on redundant steps repeatedly reinforce rigid behavior patterns.

Beyond redundant sampling, outcome rewards provide little guidance for assigning credit to intermediate reasoning steps. Recent studies seek finer-grained credit assignment by decomposing intermediate steps along a single dimension, such as cognitive reasoning tags or shared environment states (Zhang et al., 2026b; Feng et al., 2025). However, such single-axis decompositions expose a fundamental **credit assignment dilemma**: reliable intermediate credit requires both baseline alignment and reasoning-mode discrimination. Tag-based methods (Zhang et al., 2026b) preserve cognitive semantics, but normalize advantages within same-tag groups without aligning the underlying states, causing state difficulty to be mistaken for reasoning quality. State-matching methods (Feng et al., 2025) provide better state-aligned baselines by grouping identical observations, but leave the model’s reasoning modes undifferentiated. Consequently, credit may become more fine-grained in form, but not more discriminative in function: it still fails to cultivate the model’s ability to explicitly distinguish, evaluate, and regulate its own meta-reasoning modes and action patterns under complex state variations over long horizons.

To address these issues, we propose **STARK** (Strategic and Tactical Advantage Reinforcement via Confidence-Guided Keypoint Exploration, a unified RL framework for long-horizon agent training (Figure 1). STARK first estimates the agent’s confidence at each keypoint step by combining token-level generation probability with action validity. At low-confidence keypoints, it initiates dynamic branching from the current state and injects reflective behaviours into the expanded branches,

prompting it to re-evaluate the decision through corrective and self-discriminative reasoning modes.

Based on this branching infrastructure, STARK introduces a *strategic and tactical advantage* strategy for credit assignment. The *strategic* advantage uses hash-based state anchors to compare alternative reasoning modes under the same environmental observation, while the *tactical* advantage groups steps by meta-reasoning tags to evaluate mode-specific execution quality. This couples keypoint exploration with advantage learning, guiding the model toward reasoning and action modes that are both diverse and effective for long-horizon decision making. Across ALFWorld, ScienceWorld, and WebShop, **STARK** delivers strong performance gains (Figure 2), efficiency, and cross-domain generalization. Our contributions are:

- **Confidence-driven exploration.** We introduce a dynamic branching mechanism that estimates epistemic uncertainty at each decision point and applies reflection-directed intervention to diversify cognitive mode, reallocating the computational budget toward more informative exploratory branching rollouts.
- **Strategic and tactical advantage estimation.** We propose a dual-track advantage framework that couples state-grounded mode selection with mode-specific execution optimization: strategic advantage compares cognitive modes under shared environment observations while the tactical advantage evaluates the quality of each cognitive behavior within tag-aligned groups.
- **Strong empirical results.** Extensive experiments show that STARK with a 1.5B backbone surpasses GPT-5 on the hardest out-of-domain splits and reduces cognitive mode collapse by up to 92% relative to baselines.

2 Methodology

2.1 Confidence-Driven Exploration

Standard RL training relies on *passive* linear rollouts: the agent samples a single trajectory per state and receives no corrective signal until episode termination. In complex long-horizon environments, this exposes the agent to *cognitive mode collapse*: a failure mode in which the model, anchored by pre-trained priors, repeatedly selects the same ineffective cognitive tag (e.g., issuing dozens of futile `<explore>` actions without autonomously triggering `<reflection>`). To overcome this, STARK

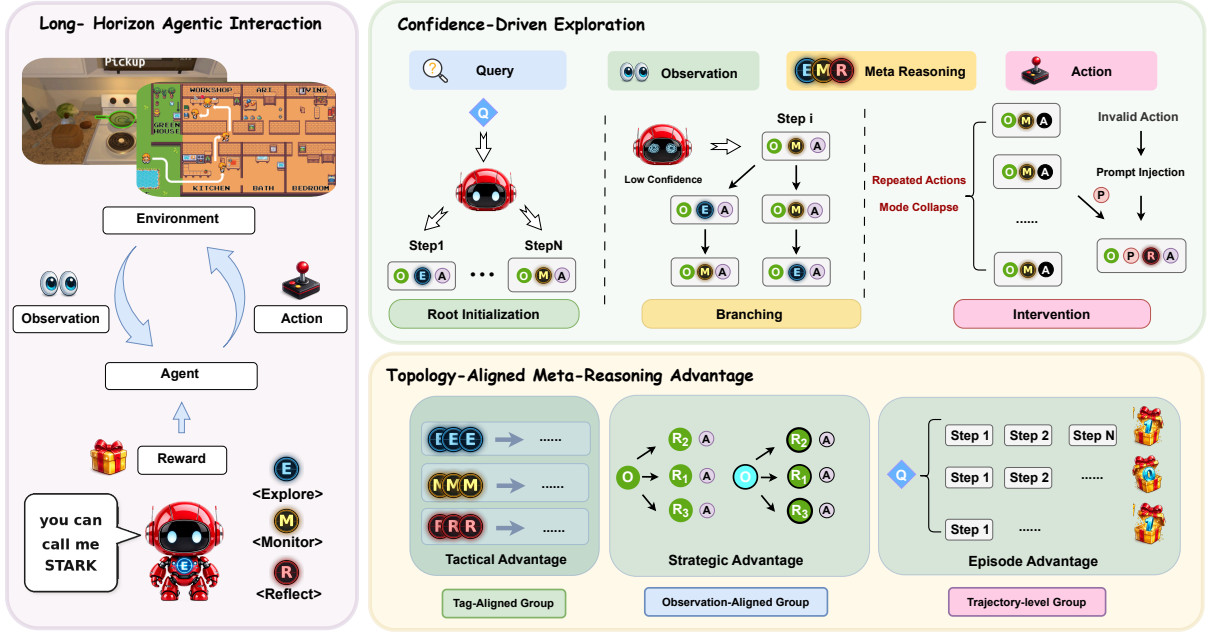


Figure 3: Overview of the STARK framework. **(Left) Long-horizon agentic interaction:** the agent must sustain coherent reasoning over sparse-reward trajectories. **(Upper Right) Confidence-driven keypoint exploration:** at low-confidence nodes, STARK branches from the current physical state $s_{i,t}$ and applies directed cognitive intervention via prefix injection (e.g., reflection) to elicit counterfactual reasoning modes. **(Lower Right) Topology-aligned meta-reasoning advantage:** tag-aligned groups yield tactical advantages; anchor-aligned groups ($u_{i,t} = \text{Hash}(o_{i,t})$) yield strategic advantages; trajectory-level groups preserve episode-level success signals.

replaces passive sampling with a *confidence-driven dynamic branching* mechanism that actively intervenes at high-uncertainty decision nodes.

1. Multi-Root Initialization. For each task prompt p , we initialize M independent parallel trajectory roots to ensure sufficient initial diversity:

$$\{\tau^{(1,0)}, \dots, \tau^{(M,0)}\} \sim \pi_{\theta}(\cdot \mid h_0). \quad (1)$$

2. Dynamic Branching. For an active trajectory i at step t , we quantify the agent’s epistemic confidence $C_{i,t} \in [0, 1]$ by combining token-level generation probability with action validity:

$$C_{\text{token}}^{(i,t)} = \exp\left(\frac{1}{K} \sum_{k=1}^K \log \pi_{\theta}(a_{i,t,k} \mid s_{i,t})\right), \quad (2)$$

$$C_{i,t} = \frac{\omega_c C_{\text{token}}^{(i,t)} + \omega_v \mathbb{I}_{\text{valid}}}{\omega_c + \omega_v}, \quad (3)$$

where K is the number of generated tokens, $\mathbb{I}_{\text{valid}}$ indicates whether the action is executable in the current environment, and ω_c, ω_v are balancing weights. A lower $C_{i,t}$ signals greater epistemic uncertainty, making the current decision node a prime candidate for additional exploration.

3. Directed Cognitive Intervention. The branching probability is set to $P_{\text{branch}} = 1 - C_{i,t}$. When a Bernoulli trial $b \sim \text{Ber}(P_{\text{branch}})$ succeeds and the global leaf-node budget N is not exhausted, STARK spawns one or more cloned branches from the *identical* physical state $s_{i,t}$. Crucially, rather than allowing unconstrained re-sampling—which would reproduce the same cognitive bias, we apply a **directed cognitive intervention** via prompt injection. Specifically, a candidate meta-cognitive tag $c = \langle \text{reflection} \rangle$ is force-prepended to the history h_t before generation:

$$(z_t^{\text{new}}, a_t^{\text{new}}) \sim \pi_{\theta}(\cdot \mid h_t \oplus c). \quad (4)$$

By systematically injecting the cognitive prefix across parallel clones, STARK encourages broader coverage of the search space at high-uncertainty nodes, mitigating the homogeneous exploration that passive sampling induces.

2.2 Meta-Reasoning Framework

Task Formulation. We formulate each long-horizon agentic task as a Partially Observable Markov Decision Process (POMDP), defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{O}, \mathcal{P}, \mathcal{R}, \gamma)$. At each timestep t , the agent receives an observation $o_t \in \mathcal{O}$ and maintains an interaction history $h_t =$

$(o_0, a_0, r_0, \dots, o_t)$. The agent’s policy π_θ first generates an explicit meta-cognitive reasoning trace z_t and then commits to a physical action a_t , i.e., $(z_t, a_t) \sim \pi_\theta(\cdot \mid h_t)$. The environment transitions according to $\mathcal{P}(s_{t+1} \mid s_t, a_t)$ and emits reward r_t . Our objective is to optimize π_θ to maximize the expected cumulative return $J(\pi) = \mathbb{E} \left[\sum_{t=0}^T \gamma^t r_t \right]$ in scenarios where T is large and rewards are sparse.

Operationalizing Meta-Reasoning. To decouple high-level cognitive planning from low-level physical execution and enable fine-grained supervision, we adopt a structured meta-reasoning framework consistent with Zhang et al. (2026b). Four mutually exclusive meta-cognitive tags explicitly represent the agent’s internal state at each step:

- **<planning>**: Decomposes the task into high-level sub-goals and formulates an overall strategy.
- **<explore>**: Generates hypotheses or candidate actions to navigate uncertainty or bottlenecks.
- **<reflection>**: Diagnoses previous errors and formulates corrective actions.
- **<monitor>**: Tracks current task progress against the overall plan during routine execution.

Prepending one of these tags to each reasoning trace z_t serves three purposes: it regulates the agent’s decision-making process, enables tag-conditioned reward attribution, and provides the structured anchor points required by our branching and advantage mechanisms (§2.1–2.3).

2.3 Dual-Track Advantage Attribution

Even with diverse branching, effective policy learning requires a *credit assignment* mechanism that accurately reflects the true quality of each decision. Standard GRPO (Shao et al., 2024) normalizes rewards globally within the group of all rollout trajectories generated for a given prompt p :

$$\hat{A}_{i,t}^{\text{standard}} = \frac{r_{i,t} - \mu_{\mathcal{T}_p}}{\sigma_{\mathcal{T}_p} + \varepsilon}, \quad (5)$$

where $\mathcal{T}_p$ denotes the set of all steps for prompt p . This formulation conflates the *objective* state difficulty $V(s)$ with the *subjective* action quality $Q(s, a)$, introducing high gradient variance. To resolve this, STARK introduces **topology alignment** to anchor comparisons at the same physical state, and decomposes the advantage signal into two orthogonal tracks.

Topology Alignment via Anchor Identifiers.

We map the raw environmental observation $o_{i,t}$ to a hash-based anchor identifier:

$$u_{i,t} = \text{Hash}(o_{i,t}). \quad (6)$$

All steps sharing the same prompt p and the same anchor $u_{i,t}$ are grouped into a *local control set* $\mathcal{G}_{\text{anchor}}(p, u_{i,t})$. Because these steps originate from identical physical environment states, any reward difference within this group is attributable solely to the agent’s cognitive choices—not to extrinsic state difficulty.

Track 1: Strategic Advantage (Anchor-Aligned).

The strategic advantage evaluates *which* action mode is most beneficial at a particular decision node. Within the local control set $\mathcal{G}_{\text{anchor}}(p, u_{i,t})$, we perform intra-anchor normalization:

$$\begin{aligned} \mu_{\text{anchor}} &= \frac{1}{|\mathcal{G}_{\text{anchor}}|} \sum_{j \in \mathcal{G}_{\text{anchor}}} r_j^{\text{MR}}, \\ \hat{A}_{i,t}^{\text{MR_anchor}} &= \frac{r_{i,t}^{\text{MR}} - \mu_{\text{anchor}}}{\sigma_{\text{anchor}} + \varepsilon}. \end{aligned} \quad (7)$$

Because the actions in $\mathcal{G}_{\text{anchor}}(p, u_{i,t})$ are compared under the same observed environment anchor, μ_{anchor} serves as a local empirical baseline that controls for much of the state-difficulty variation. The resulting advantage estimates the relative gain of selecting one cognitive mode over alternatives at the same decision point, answering the strategic question: *What should I do next at this specific decision node?* This anchor-based tournament is only made possible by the confidence-driven branching in §2.1, which encourages reflective counterfactual branches to be realized alongside the model’s naturally sampled cognitive modes.

Track 2: Tactical Advantage (Tag-Aligned).

The tactical advantage evaluates *how well* the agent executes a specific cognitive behavior. For a given prompt p and cognitive tag c , let $\mathcal{G}_{\text{tag}}(p, c)$ denote the set of all steps across its trajectories that share tag c . The tactical advantage is normalized within this group:

$$\hat{A}_{i,t}^{\text{MR_tag}} = \frac{r_{i,t}^{\text{MR}} - \mu_{\text{tag}}}{\sigma_{\text{tag}} + \varepsilon}, \quad \mu_{\text{tag}} = \frac{1}{|\mathcal{G}_{\text{tag}}|} \sum_{j \in \mathcal{G}_{\text{tag}}} r_j^{\text{MR}}, \quad (8)$$

where r^{MR} denotes the step-level meta-reasoning reward computed by the same rule-based verifier as RLVMR (Zhang et al., 2026b). STARK does not

modify the definition of r^{MR} ; instead, it reuses this reward signal and improves credit assignment by organizing it into tag-aligned and anchor-aligned comparison groups. This signal answers the question: *Is this exploratory (or reflective) action better than other actions for the same task?*

2.4 Policy Optimization

The final advantage signal fuses the macroscopic episode outcome $\hat{A}_{i,t}^{\text{traj}}$ with the two tracks:

$$A_{i,t} = \hat{A}_{i,t}^{\text{traj}} + \omega_{\text{step}} \left(w_t \hat{A}_{i,t}^{\text{MR_tag}} + w_a \hat{A}_{i,t}^{\text{MR_anchor}} \right), \quad (9)$$

where w_t and w_a balance tactical execution quality and strategic cognitive selection, respectively. This three-level fusion ensures that STARK optimizes simultaneously for (i) long-term task success, (ii) high-quality execution of individual cognitive behaviors, and (iii) adaptive selection of the most beneficial cognitive mode at each environmental state. The computed advantages $A_{i,t}$ are then used directly in the standard GRPO clipped surrogate objective for policy updates. The complete procedure is summarized in Algorithm 1.

3 Experiments

3.1 Experimental Settings

Benchmarks. We evaluate on three long-horizon interactive environments: (1) **Embodied Decision Making (ALFWorld)** (Shridhar et al., 2020), a text-simulated household requiring embodied task grounding and execution; (2) **Scientific Reasoning (ScienceWorld)** (Wang et al., 2022), a scientific reasoning benchmark requiring over 30 interaction steps per task; and (3) **Web Navigation (WebShop)** (Yao et al., 2022), a large-scale e-commerce simulator with 1.1 million products testing broad exploration and retrieval.

Baselines. We compare against closed-source LLMs (GPT-4o (OpenAI, 2024), GPT-5-mini/GPT-5 (OpenAI, 2025), Gemini-2.5-Pro (Comanici et al., 2025)), prompting methods (ReAct (Yao et al., 2023), Reflexion (Shinn et al., 2023)), RL methods (GRPO (Shao et al., 2024), ETO (Song et al., 2024), GiGPO (Feng et al., 2025), RLVMR (Zhang et al., 2026b)), and recent long-horizon frameworks (SLEA-RL (Wang and Jiang, 2026), HCAPO (Tan et al., 2026), HGPO (He et al., 2026)).

Evaluation Metrics. We report Success Rate at generalization levels (L0, L1, L2) on ALFWorld

and ScienceWorld, fine-grained sub-category success rates on ALFWorld, and both Success Rate (SR) and Average Score on WebShop.

Implementation Details. Following prior work (Zhang et al., 2026b; Wu et al., 2026b), we use Qwen2.5-1.5B/7B-Instruct (Qwen Team, 2024) as base models, with cold-start SFT on 200 trajectories to bootstrap meta-reasoning tag generation. We set rollout budget $N=8$, initial roots $M=4$, and branching factor $B=2$. RL training runs for 100 steps with batch size 16. Full details are in Appendix C.4.

3.2 Main Results

Table 1 presents results across complex long-horizon tasks. We can draw three key findings:

✧ *STARK achieves strong performance across model scales and task domains.* STARK consistently surpasses all baselines at both 1.5B and 7B model scales. At 1.5B, it attains 93.8% on ALFWorld and 81.6% on WebShop, outperforming the strongest RL baseline SLEA-RL by +6.3% and +6.2%. Notably, STARK-1.5B exceeds GPT-5 by wide margins (e.g., 93.8% vs. 70.0% on ALFWorld), showing that principled confidence-driven exploration enables compact open-source models to rival larger proprietary systems.

✧ *Gains are amplified on tasks demanding multi-step coordination and sequential reasoning.* Relative to GRPO, STARK delivers its largest improvements on complex sub-tasks: +40.3% on Cool and +39.2% on Look (spatial search) at 1.5B, with substantial gains also observed on routine sub-tasks such as Clean (+15.5%). This pattern supports our core design: concentrating branching budget at high-uncertainty nodes allocates search precisely where it yields the greatest marginal value.

✧ *STARK generalizes robustly to structurally distinct environments.* On WebShop, which demands broad exploration over 1.1M real-world products, STARK achieves 81.6% success at both scales, surpassing RLVMR by +10.5% (1.5B) and +7.4% (7B). Consistent gains across scales and datasets confirm that improvements stem from framework design rather than model capacity, underscoring the broad applicability of confidence-driven branching and topology-aligned advantage attribution.

3.3 Exploration Efficiency Analysis

Beyond task success, we evaluate STARK on sample efficiency and token consumption. We provide more details in Appendix D.1.

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Closed-Source Model										
Prompting	GPT-4o	91.7	50.0	34.4	52.4	45.0	15.8	52.3	12.7	6.2
Prompting	GPT5-mini	91.7	58.3	28.1	47.6	60.0	52.6	57.0	14.2	7.0
Prompting	GPT5	79.2	75.0	43.8	42.9	75.0	57.9	70.0	34.9	29.7
Prompting	Gemini-2.5-Pro	100.0	75.0	40.6	0.0	45.0	84.2	56.3	42.9	32.0
Base: Qwen2.5-1.5B-Instruct										
Prompting	Base	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
Prompting	ReAct	17.4	20.5	15.7	6.2	7.7	2.0	12.8	40.1	11.3
Prompting	Reflexion	35.3	22.2	21.7	13.6	19.4	3.7	21.8	55.8	21.9
RL Training	PPO (with critic)	64.8	40.5	57.1	60.6	46.4	47.4	54.4	73.8	51.5
RL Training	RLOO	88.3	52.8	71.0	62.8	66.4	56.9	69.7	73.9	52.1
RL Training	GiGPO	96.0	76.5	91.8	91.3	71.7	79.5	86.1	83.5	67.4
RL Training	RLVMR	95.2	78.8	91.2	90.2	83.9	77.6	87.9	86.7	71.1
RL Training	SLEA-RL	92.6	80.0	96.2	94.7	95.0	76.5	87.5	88.7	75.4
RL Training	HCAPO	88.6	75.0	97.6	90.7	84.2	74.2	87.0	83.8	68.5
RL Training	GRPO	85.3	53.7	84.5	78.2	59.7	53.5	72.8	75.8	56.8
RL Training	STARK (Ours)	100.0	92.9	100.0	93.3	100.0	62.5	93.8	90.0	81.6
	Δ vs. GRPO	+14.7	+39.2	+15.5	+15.1	+40.3	+9.0	+21.0	+14.2	+24.8
Base: Qwen2.5-7B-Instruct										
Prompting	Base	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 (with critic)	92.3	64.0	92.5	89.5	80.3	68.8	80.4	81.4	68.7
RL Training	RLOO	87.6	78.2	87.3	81.3	71.9	48.9	75.5	80.3	65.7
RL Training	GiGPO	91.8	88.6	95.9	90.2	86.5	85.2	90.2	86.2	75.2
RL Training	RLVMR	95.3	88.2	90.1	92.4	89.8	86.7	91.8	84.4	74.2
RL Training	SLEA-RL	98.5	85.1	99.0	93.8	96.4	90.4	93.5	87.6	76.3
RL Training	HCAPO	99.1	90.3	97.3	81.8	90.8	81.9	91.4	85.1	73.8
RL Training	GRPO	90.8	66.1	89.3	74.7	72.5	64.7	77.6	79.3	66.1
RL Training	STARK (Ours)	100.0	92.3	100.0	86.0	95.6	100.0	96.9	92.1	81.6
	Δ vs. GRPO	+9.2	+26.2	+10.7	+11.3	+23.1	+35.3	+19.3	+12.8	+15.5

Table 1: **Comparative analysis against recent strong RL baselines.** We report detailed success rates across distinct task types in ALFWorld and WebShop.

Sample Efficiency. Figure 4 shows that STARK consistently dominates in low-data regimes. On ALFWorld (L0), it achieves 71.8% success with only 20% of training data, matching or exceeding GRPO’s full-data performance (76.6%) at just 40%. The advantage is even more pronounced on ScienceWorld, where GiGPO and GRPO collapse to near 0% under data scarcity while STARK sustains above 30% throughout. This confirms that concentrating exploration at high-uncertainty nodes yields denser training signals than uniform sampling, translating into faster policy convergence.

Token Consumption. As shown in Table 2, STARK consumes only 84.0%, 54.0%, and 41.7% of the tokens used by GRPO on ALFWorld, Sci-

enceWorld, and WebShop, respectively, while delivering superior performance across all three benchmarks. This indicates that reflection-directed branching produces more purposeful, information-dense trajectories, reducing redundant token expenditure of passive exploratory loops.

3.4 Cross-Domain Generalization

As shown in Table 3, STARK exhibits robust generalization to unseen environments and tasks. Existing long-horizon RL methods often suffer from severe performance degradation in out-of-distribution (OOD) scenarios. In contrast, STARK maintains high robustness across all generalization levels (L0 to L2). In the most challenging ALFWorld L2 (un-

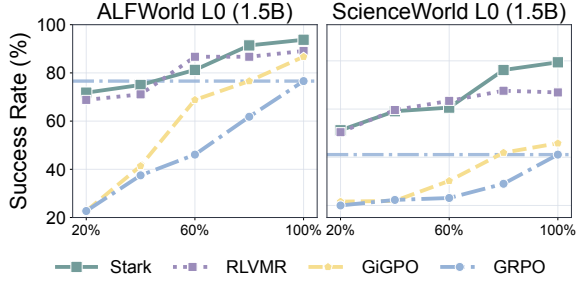


Figure 4: **Sample efficiency comparison.** STARK demonstrates superior performance even in low-data regimes, outperforming GRPO and other baselines.

Method	ALFWorld↓	ScienceWorld↓	WebShop↓
GRPO (Chain)	100.0%	100.0%	100.0%
RLVMR (Chain)	181.0%	121.0%	98.5%
STARK (Ours)	84.0%	54.0%	41.7%
Reduction vs. GRPO	16.0%	46.0%	58.3%

Table 2: **Token efficiency (1.5B).** Relative token consumption normalized to GRPO (100%). Lower is better.

seen task categories and instances), STARK (1.5B) achieves a 73.4% success rate, representing a massive +43.7% improvement over GRPO. A similar substantial gain of +36.8% is observed in ScienceWorld L2. This validates our core hypothesis: exploration mechanisms based on dynamic branching prevent overfitting to specific trajectories, encouraging the agent to learn highly transferable, generalized exploration strategies.

3.5 Multimodal Extension

To examine the broader applicability of STARK, we test on two multimodal visual benchmarks: Sokoban and EZPoints (Table 4). Both strip away explicit textual state descriptions, forcing the agent to reason directly from raw visual observations. Sokoban is particularly demanding: as illustrated in Figure 6, boxes can only be pushed, never pulled. Therefore, a single erroneous move can render the puzzle permanently unsolvable, requiring deep forward-looking spatial reasoning. Despite this, STARK achieves +19.6% over GRPO on Sokoban and +12.5% on average, confirming that confidence-driven branching is modality-agnostic and generalizes to multimodal long-horizon settings. Dataset details are provided in Appendix C.1.

3.6 Ablation Study and Other Analysis

Ablation Study. Table 5 reports compact ablations on ScienceWorld (1.5B). Removing confidence-based branching causes the largest degradation (L2: 47.7%→26.5%), confirming that

Method	ALFWorld			ScienceWorld		
	L0	L1	L2	L0	L1	L2
<i>Closed-Source Models</i>						
GPT-4o	52.3	48.4	38.3	32.0	32.8	32.8
GPT-5-mini	57.0	54.7	46.1	39.8	31.3	31.4
GPT-5	70.0	60.2	63.3	53.9	33.6	33.6
Gemini-2.5-Pro	56.3	55.0	55.5	32.8	27.3	30.5
<i>Qwen2.5-1.5B</i>						
ReAct	11.3	13.7	10.2	1.2	0.8	0.8
SFT	43.0	38.7	17.6	20.3	18.0	12.5
ETO	64.1	66.4	25.8	39.1	22.7	15.6
GRPO	76.6	71.1	29.7	21.1	13.7	10.9
GiGPO	86.7	83.2	48.0	25.8	15.2	4.7
RLVMR	89.1	87.9	56.3	46.9	34.4	26.5
STARK (Ours)	93.8	89.1	73.4	64.8	50.0	47.7
Δ vs. GRPO	+17.2	+18.0	+43.7	+43.7	+36.3	+36.8
<i>Qwen2.5-7B</i>						
ReAct	23.1	28.5	27.0	7.8	11.3	6.3
SFT	63.3	57.0	37.5	36.7	32.0	23.4
ETO	70.3	74.2	51.6	62.5	40.6	28.1
GRPO	79.3	77.3	52.3	49.1	30.1	26.6
GiGPO	89.5	90.2	67.2	53.4	35.2	25.8
RLVMR	91.4	91.8	83.6	67.2	43.0	32.2
STARK (Ours)	96.9	96.1	90.6	75.0	63.3	60.2
Δ vs. GRPO	+17.6	+18.8	+38.3	+25.9	+33.2	+33.6

Table 3: **Cross-domain generalization analysis.** We evaluate success rates (%) on ALFWorld and ScienceWorld across three progressive levels: **L0** (seen categories and instances), **L1** (seen categories, unseen instances), and **L2** (unseen categories and instances).

Method	Sokoban[6×6]↑	EZPoints↑	Avg.↑
GPT-5	45.3	98.4	71.9
ReAct	11.7	3.1	7.4
GRPO	67.1	86.9	77.0
STARK	86.7	92.2	89.5
Δ vs. GRPO	+19.6	+5.3	+12.5

Table 4: **Multimodal extension analysis.** We present success rates (%) on two multimodal visual benchmarks, Sokoban[6×6] and EZPoints, using Qwen2.5-VL-3B-Instruct as the underlying backbone model.

targeted branching is essential for constructing informative counterfactual trajectories. Jointly removing the reflection intervention and anchor-aligned advantage also reduces performance, indicating that the generated branches must be both cognitively redirected and converted into state-aligned learning signals. Appendix D.2 provides single-component controls and a random-branching control that further disentangle these effects.

Exploration Quality. We evaluate exploration quality via three diagnostic metrics: *invalid action rate* (proportion of unexecutable actions), *repetitive action rate* (proportion of actions repeated within

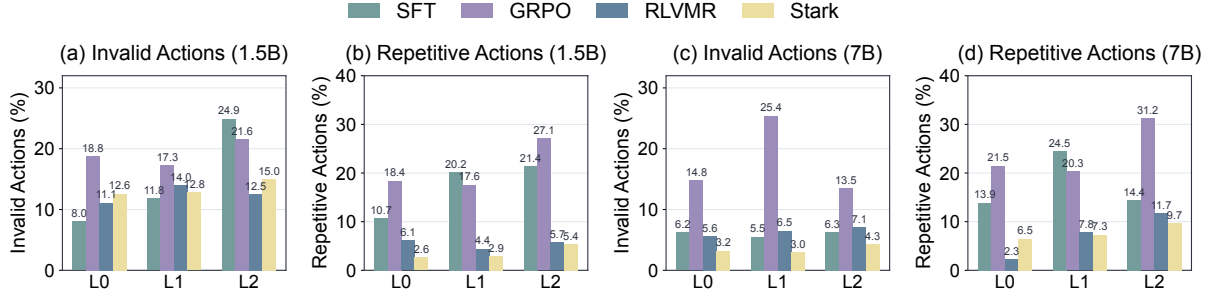


Figure 5: **Exploration quality comparison on ALFWorld.** STARK consistently suppresses invalid and repetitive actions across both 1.5B and 7B scales, revealing more purposeful exploration and improved execution precision.

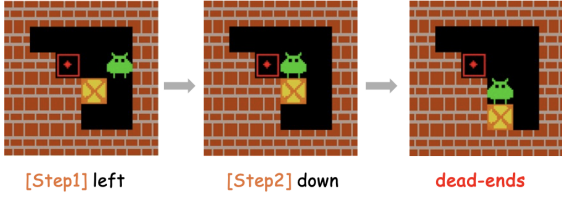


Figure 6: **Irreversible dead-ends in Sokoban.** Boxes can only be pushed, so a single erroneous move (e.g., pushing a box into a corner) renders the puzzle permanently unsolvable, demanding deep strategic foresight.

a trajectory), and *mode collapse rate* (proportion of consecutive steps repeating the immediately preceding action). As shown in Figure 5, STARK substantially suppresses both invalid and repetitive actions across all generalization levels and model scales, indicating that confidence-driven branching produces more purposeful, non-redundant exploration. Further analysis is in Appendix D.2.

4 Related Work

LLM Agents. Deploying LLMs as interactive agents has attracted growing interest (Zhang et al., 2025; Wu et al., 2026b). Prompting strategies like ReAct (Yao et al., 2023) and Reflexion interleave reasoning with actions but lack parameter updates. ETO (Song et al., 2024) uses contrastive trajectory pairs for offline fine-tuning, bypassing online credit assignment. More recent work applies online RL to agentic benchmarks (Shridhar et al., 2020; Wang et al., 2022), achieving substantial gains over prompting. Our work builds on this paradigm but targets two persistent failure modes: cognitive mode collapse from passive rollout sampling, and credit assignment distortion from conflating state difficulty with reasoning quality.

Exploration for Agentic RL. Standard RL assumes repeated sampling provides sufficient exploration (Guo et al., 2025; Wu et al., 2026a;

Configuration	L0	L1	L2
STARK (Full)	64.8	50.0	47.7
w/o Reflection & Anchor Advantage	53.1	51.5	41.4
w/o Branching & Reflection & Anchor	46.9	34.4	26.5

Table 5: **Compact ablation on ScienceWorld (1.5B).** Success rates (%; higher is better) compare the full model with variants that remove confidence-based branching or jointly remove the reflection intervention and anchor-aligned advantage. Individual component controls are reported in Appendix D.2.

Yue et al., 2025), but LLM policies collapse toward high-probability trajectories (Snell et al., 2024), especially where high-uncertainty nodes are sparse yet consequential. Recent structured rollout methods (Zhang et al., 2025) still reproduce the same cognitive bias via unconstrained re-sampling. STARK instead estimates epistemic confidence from token-level probability and action validity, triggering directed reflection at high-uncertainty keypoints without an external model.

Credit Assignment in RL. Fine-grained credit assignment remains the central bottleneck for long-horizon learning. GRPO (Shao et al., 2024) conflates state difficulty with action quality; GiGPO (Feng et al., 2025) groups by observation but leaves cognitive modes undifferentiated; RLVMR (Zhang et al., 2026b) groups by tag but spans divergent states. HCAPO (Tan et al., 2026), and HGPO (He et al., 2026) each enrich credit along one axis, leaving state alignment and mode discrimination entangled. STARK resolves this through hash-anchored strategic evaluation of reasoning-mode selection and tag-aligned tactical assessment of mode-specific execution.

5 Conclusion

STARK couples confidence-driven exploration with dual-track advantage estimation for long-horizon agents. Reflective counterfactual branching concen-

trates rollout budget on the most informative states; the tactical track refines meta-cognitive execution via tag-aligned comparison while the strategic track guides mode selection via anchor-aligned comparison. Experiments show consistent gains in task success, efficiency, and generalization, with multi-modal results suggesting broader applicability.

Limitations

Confidence-driven branching introduces additional environment interactions at high-uncertainty nodes, increasing training time relative to single-trajectory methods. Although this overhead is concentrated on informative decision points rather than uniformly spent across full rollouts, the wall-clock cost may still grow in environments with expensive state replay or slow simulators. In addition, our current directed intervention uses a fixed prefix for counterfactual branching. This choice is simple and effective in our experiments, but the optimal intervention may vary across task types, training stages, and model capacities. A natural extension is to learn not only when to branch, but also which cognitive mode to instantiate.

Acknowledgments

This work is supported by the National Key R&D Program of China (2024YFB2808802), and the Guangdong S&T Program (2025B0101130002).

References

- Arash Ahmadian, Chris Cremer, Matthias Gall , Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet  st n, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for LLM agent training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. 2026. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *The Fourteenth International Conference on Learning Representations*.
- Zhengxi Lu, Yuxiang Chai, Yaxuan Guo, Xi Yin, Liang Liu, Hao Wang, Han Xiao, Shuai Ren, Pengxiang Zhao, Guangyi Liu, Guanqing Xiong, and Hongsheng Li. 2026a. UI-R1: Enhancing efficient action prediction of GUI agents by reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 17608–17616.
- Zhengxi Lu, Zhiyuan Yao, Zhuowen Han, Zi-Han Wang, Jinyang Wu, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026b. Self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2605.15155*.
- Zhengxi Lu, Zhiyuan Yao, Jinyang Wu, Chengcheng Han, Qi Gu, Xunliang Cai, Weiming Lu, Jun Xiao, Yueting Zhuang, and Yongliang Shen. 2026c. SKILL0: In-context agentic reinforcement learning for skill internalization. *arXiv preprint arXiv:2604.02268*.
- Zhengxi Lu, Jiabo Ye, Fei Tang, Yongliang Shen, Haiyang Xu, Ziwei Zheng, Weiming Lu, Ming Yan, Fei Huang, Jun Xiao, and Yueting Zhuang. 2025. UI-S1: Advancing GUI automation via semi-online reinforcement learning. *arXiv preprint arXiv:2509.11543*.
- OpenAI. 2024. [Hello gpt-4o](#). Accessed: 2024-05.
- OpenAI. 2025. [Introducing gpt-5](#). Accessed: 2025-08.
- Qwen Team. 2024. [Qwen2.5: A party of foundation models](#). Accessed: 2024-09.
- Max-Philipp B. Schrader. 2018. gym-sokoban. <https://github.com/mpSchrader/gym-sokoban>.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseek-math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.
- Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and error: Exploration-based trajectory optimization for llm agents. *arXiv preprint arXiv:2403.02502*.
- Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. 2026. Hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*.
- Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, and 1 others. 2026. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*.
- Junqiao Wang, Zeng Zhang, Yangfan He, Zihao Zhang, Xinyuan Song, Yuyang Song, TIANYU SHI, Yuchen Li, Hengyuan Xu, Kunyu Wu, Yi Xin, Zhongwei Wan, Xinhang Yuan, Zijun Wang, Kuan Lu, Menghao Huo, Jingqun Tang, Guangwu Qian, Keqin Li, and 2 others. 2026. Enhancing code LLMs with reinforcement learning in code generation: A survey. In *ICLR 2026 Workshop on Logical Reasoning of Large Language Models*.
- Prince Zizhuang Wang and Shuli Jiang. 2026. Slearl: Step-level experience augmented reinforcement learning for multi-turn agentic training. *arXiv preprint arXiv:2603.18079*.
- Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
- Weihang Wang, Xinhao Li, Ziyue Wang, Yan Pang, Jielei Zhang, Peiyi Li, Qiang Zhang, and Longwen Gao. 2025. [Diving into mitigating hallucinations from a vision perspective for large vision-language models](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 17271–17289.
- Jinyang Wu, Chonghua Liao, Mingkuan Feng, Shuai Zhang, Zhengqi Wen, Haoran Luo, Ling Yang, Huazhe Xu, and Jianhua Tao. 2026a. [TemplateRL: Structured template-guided reinforcement learning for LLM reasoning](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 4050–4080, San Diego, California, United States. Association for Computational Linguistics.
- Jinyang Wu, Shuo Yang, Yuhao Shen, Shuai Zhang, Zhengqi Wen, and Jianhua Tao. 2026b. [SPARK: Strategic policy-aware exploration via dynamic branching for long-horizon agentic learning](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23981–24004, San Diego, California, United States. Association for Computational Linguistics.
- Hang Yan, Xinyu Che, Fangzhi Xu, Qiushi Sun, Zichen Ding, Kanzhi Cheng, Jian Zhang, Tao Qin, Jun Liu, and Qika Lin. 2026a. TIDE: Trajectory-based diagnostic evaluation of test-time improvement in LLM agents. *arXiv preprint arXiv:2602.02196*.
- Hang Yan, Fangzhi Xu, Qiushi Sun, Jinyang Wu, Zixian Huang, Muye Huang, Jingyang Gong, Zichen Ding, Kanzhi Cheng, Yian Wang, Xinyu Che, Zeyi Sun, Jian Zhang, Zhangyue Yin, Haoran Luo, Ben Kao, and Qika Lin. 2026b. OdysseyArena: Benchmarking large language models for long-horizon, active and inductive interactions. *arXiv preprint arXiv:2602.05843*.
- Run Yang, Weihang Wang, Boheng Sheng, Yuchen He, Jielei Zhang, Pengyu Chen, Zhiyu Wu, Qiang Sun, Huyang Sun, and Longwen Gao. 2026. MemeMind: Reference-guided trace construction for offline context optimization. *arXiv preprint arXiv:2608.09316*.
- Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023. [React: Synergizing reasoning and acting in language models](#). In *The Eleventh International Conference on Learning Representations*.
- Zonghao Ying, Yangguang Shao, Jianle Gan, Gan Xu, Wenxin Zhang, Quanchen Zou, Junzheng Shi, Zhenfei Yin, Mingchuan Zhang, Aishan Liu, and Xianglong Liu. 2026a. [SecureWebArena: A holistic security evaluation benchmark for LLM-based web agents](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 11986–11998.
- Zonghao Ying, Le Wang, Yisong Xiao, Jiakai Wang, Yuqing Ma, Jinyang Guo, Zhenfei Yin, Mingchuan Zhang, Aishan Liu, and Xianglong Liu. 2026b. AGENTS SAFE: Benchmarking the safety of embodied agents on hazardous instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 37664–37673.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?](#) In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. 2024. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 37, pages 110935–110971. Curran Associates, Inc.

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

Yanfei Zhang. 2025. Agent-as-tool: A study on the hierarchical decision making with reinforcement learning. *arXiv preprint arXiv:2507.01489*.

Yanfei Zhang, Xu Lin, and Chenglin Wu. 2026a. StepOPSD: Step-aware online preference self-distillation for agent reinforcement learning. *arXiv preprint arXiv:2605.27140*.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2026b. RLVMR: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. In *The Fourteenth International Conference on Learning Representations*.

D Supplementary Results	17
D.1 Sample Efficiency	17
D.2 Additional Ablation Study	17
D.3 Sensitivity Analysis	21
E Case Study	21
F Additional Discussion	21
F.1 Relation to Prior Agentic RL Methods	21
F.2 Compatibility with Alternative Policy Optimizers	22
F.3 Discussion on Intervention Tag Selection	23
F.4 Failure Mode Analysis	23
F.5 Multi-Seed Robustness	23
F.6 Future Work	23

Appendix of STARK

This supplementary material provides in-depth insights into our STARK method, covering additional theoretical analysis, additional experimental details, results and analysis. The appendix is organized as follows:

Contents

A Theoretical Analysis	12
A.1 Problem Setup and Notations . . .	12
A.2 Strategic Branching Efficiency . . .	12
A.3 Topology-Aligned Credit Assignment	12
A.4 Environment-Aware Confidence Estimation	13
A.5 Summary	13
B Algorithm Details	13
C Additional Experimental Details	13
C.1 Datasets	13
C.2 Baselines	15
C.3 Evaluation Details	16
C.4 Implementation Details	16

A Theoretical Analysis

In this section, we provide a heuristic theoretical analysis to build intuition for why STARK improves exploration efficiency and credit assignment in long-horizon agentic reinforcement learning. Our goal is not to provide rigorous theoretical guarantees, but rather to conceptually explain the mechanisms underlying STARK and connect them with the empirical observations in the main text.

A.1 Problem Setup and Notations

We model long-horizon agentic interaction as a Partially Observable Markov Decision Process (POMDP). At each timestep t , the policy π_θ generates a meta-cognitive reasoning trace z_t , followed by an executable environment action a_t .

We denote the environment observation as o_t and define a topology anchor:

$$u_t = \text{Hash}(o_t),$$

which identifies locally comparable decision contexts across trajectories.

Following prior long-horizon reasoning work, we assume that only a sparse subset of decision nodes are truly critical for downstream task success. Let:

$$\mathcal{S}_{\text{crit}} = \{s \mid U(s) > \tau\},$$

where $U(s)$ denotes the agent’s uncertainty at state s , and τ is a predefined uncertainty threshold. Intuitively, these critical states correspond to decision bottlenecks where additional exploration is most valuable.

Let:

$$K_c = |\mathcal{S}_{\text{crit}} \cap \text{trajectory}|,$$

and typically $K_c \ll K$ for long-horizon tasks with horizon length K .

STARK dynamically allocates exploration budget around these uncertain nodes through strategic branching and local comparative optimization.

A.2 Strategic Branching Efficiency

Analysis 1 (Concentrated Exploration Around Critical States). Consider two exploration strategies under the same rollout budget N :

- **Uniform trajectory sampling**, where each rollout independently explores the environment;
- **STARK’s strategic branching**, where additional branches are selectively allocated to uncertain decision nodes.

Under uniform exploration, most rollout budget is spent repeatedly traversing routine states. Since critical states are sparse ($K_c \ll K$), only a small fraction of generated trajectories contribute meaningful exploration signal at important decision points.

In contrast, STARK reallocates exploration budget toward uncertain nodes by branching under shared trajectory prefixes. As a result, the effective number of locally comparable exploratory decisions around critical states increases substantially.

Heuristically, if each critical node receives B additional branch samples, then the effective exploration density around informative states scales approximately with:

$$O(B \cdot K_c),$$

rather than being uniformly diluted across the entire horizon length K .

Importantly, STARK does not assume deterministic correction of reasoning failures. Instead, strategic intervention increases the probability of exploring alternative reasoning behaviors under uncertain states, thereby reducing persistent cognitive mode collapse observed in purely linear rollouts.

This intuition is consistent with the empirical observations in our ablation studies, where strategic branching significantly improves exploration diversity and downstream task success compared with passive trajectory sampling.

A.3 Topology-Aligned Credit Assignment

Analysis 2 (Variance Reduction via Local Topology Alignment). A major challenge in long-horizon reinforcement learning is the large variance introduced by comparing trajectories originating from substantially different environment states.

In standard group-relative optimization methods, trajectory-level rewards implicitly mix two sources of variation:

$$\text{Var}(r) \approx \text{Var}(Q(s, a)) + \text{Var}(V(s)),$$

where:

- $\text{Var}(Q(s, a))$ reflects action-quality differences;
- $\text{Var}(V(s))$ reflects state-difficulty differences.

In long-horizon environments, the state-induced variance term often dominates, making it difficult to determine whether a low reward originates from poor local reasoning or simply from encountering a difficult environment state.

STARK mitigates this issue through topology-aligned comparison. Specifically, trajectories are grouped according to shared local topology anchors:

$$u_t = \text{Hash}(o_t),$$

so that comparisons are performed only among trajectories originating from highly similar local contexts.

Under this restricted comparison structure, state-induced variance is substantially suppressed because trajectories within the same topology group share similar environmental conditions and historical prefixes.

Heuristically, the variance becomes dominated by local decision quality:

$$\text{Var}(r \mid u_t) \ll \text{Var}(r).$$

Although shared prefixes may still introduce correlations between branch samples, local topology alignment improves the comparability and information density of reinforcement signals by avoiding unnecessary cross-state comparisons.

This provides an intuitive explanation for the improved optimization stability and exploration efficiency observed empirically in STARK.

A.4 Environment-Aware Confidence Estimation

Analysis 3 (Environment-Aware Uncertainty Estimation). A common exploration trigger in reasoning agents is token-level entropy. However, in embodied or interactive environments, token uncertainty alone may not accurately reflect true decision uncertainty.

For example, multiple semantically equivalent actions may produce high entropy despite being equally valid. Conversely, an invalid environment action may be generated with high confidence and low entropy.

To better characterize actionable uncertainty, STARK incorporates environment-aware confidence estimation:

$$C_t = \frac{\omega_c C_t^{\text{token}} + \omega_v I_t^{\text{valid}}}{\omega_c + \omega_v},$$

where:

- C_t^{token} denotes token-level confidence;
- I_t^{valid} measures environment executability or action validity.

This mechanism allows STARK to identify vulnerable decision nodes where the policy is both uncertain and environmentally inconsistent. Strategic branching can then be selectively triggered around these regions.

Rather than relying solely on language-model uncertainty, STARK integrates lightweight environmental feedback into exploration control, improving robustness in unfamiliar or out-of-distribution environments.

This intuition is consistent with our empirical findings, where environment-aware confidence estimation improves generalization under unseen tasks and unseen environment configurations.

A.5 Summary

Overall, our heuristic analysis suggests three key insights underlying STARK:

1. **Strategic Branching:** Concentrating exploration budget around uncertain decision nodes improves exploration efficiency in sparse critical-state settings.
2. **Topology-Aligned Optimization:** Local topology-aware comparison suppresses state-induced variance and improves reinforcement signal comparability.
3. **Environment-Aware Confidence:** Combining token confidence with environment validity provides a more reliable trigger for adaptive exploration in long-horizon environments.

Overall, these analyses provide conceptual intuition for the empirical effectiveness of STARK, highlighting the role of strategic branching, topology-aligned comparison, and environment-aware exploration in long-horizon agentic reinforcement learning.

B Algorithm Details

We provide the pseudo-algorithm in Algorithm 1.

C Additional Experimental Details

C.1 Datasets

ALFWorld. This benchmark aligns TextWorld with the ALFRED environment, requiring agents to

Algorithm 1 STARK Algorithm

Input: Policy π_θ ; task dataset $\mathcal{Q}$; rollout budget N ; root count M ; branching factor B ; weights

$\omega_c, \omega_v, \omega_{\text{step}}, w_t, w_a$

Output: Optimized policy π_θ^*

```
1: for iteration = 1, 2, ... do
2:    $\mathcal{D} \leftarrow \emptyset$ 
3:   for task prompt  $p \sim \mathcal{Q}$  do ▷ Confidence-Driven Dynamic Branching
4:     Initialize  $M$  parallel roots and an empty trajectory group  $\mathcal{T}_p$ 
5:     while there are active trajectories and rollout budget  $N$  remains do
6:       for each active trajectory  $i$  at step  $t$  do
7:          $u_{i,t} \leftarrow \text{Hash}(o_{i,t}), \quad \mathcal{C}_{i,t} \leftarrow \frac{\omega_c C_{i,t}^{\text{token}} + \omega_v I_{i,t}^{\text{valid}}}{\omega_c + \omega_v}$ 
8:         Sample  $b_t^{(i)} \sim \text{Bernoulli}(1 - \mathcal{C}_{i,t})$ 
9:         if  $b_t^{(i)} = 1$  and budget remains then
10:          Clone up to  $B$  branches from  $s_{i,t}$  and prepend directed tag  $c$  to each branch history
11:          for each cloned branch  $j$  do
12:            Generate  $(z_t^{(j)}, a_t^{(j)}) \sim \pi_\theta(\cdot \mid h_t^{(i)} \oplus c)$ 
13:            Execute  $a_t^{(j)}$  and append the transition to  $\mathcal{T}_p$ 
14:          end for
15:        else
16:          Generate  $(z_t^{(i)}, a_t^{(i)}) \sim \pi_\theta(\cdot \mid h_t^{(i)})$ 
17:          Execute  $a_t^{(i)}$  and append the transition to  $\mathcal{T}_p$ 
18:        end if
19:      end for
20:    end while
21:    Add  $\mathcal{T}_p$  to  $\mathcal{D}$ 
22:  end for
23:  for each trajectory group  $\mathcal{T}_p \in \mathcal{D}$  do ▷ Dual-Track Advantage Attribution
24:    Form tag groups  $\mathcal{G}_{\text{tag}}(p, c)$  and anchor groups  $\mathcal{G}_{\text{anchor}}(p, u)$ 
25:    Compute  $\hat{A}_{i,t}^{\text{MR\_tag}}$  within  $\mathcal{G}_{\text{tag}}$  and  $\hat{A}_{i,t}^{\text{MR\_anchor}}$  within  $\mathcal{G}_{\text{anchor}}$ 
26:     $A_{i,t} \leftarrow \hat{A}_{i,t}^{\text{traj}} + \omega_{\text{step}}(w_t \hat{A}_{i,t}^{\text{MR\_tag}} + w_a \hat{A}_{i,t}^{\text{MR\_anchor}})$ 
27:  end for
28:  Update  $\theta$  using the GRPO objective with advantages  $A_{i,t}$  ▷ Policy Update
29: end for
```

execute household tasks (e.g., *Pick & Place*, *Clean & Place*) in a text-based interactive world (Shridhar et al., 2020). We categorize tasks into three levels: L0 (seen), L1 (unseen instances), and L2 (unseen categories) to evaluate out-of-distribution (OOD) generalization.

ScienceWorld. ScienceWorld is a complex agentic environment consisting of 30 distinct scientific tasks (Wang et al., 2022). It requires the agent to apply the scientific method over long horizons often exceeding 30 steps, serving as a robust testbed for strategic exploration and long-horizon reasoning.

WebShop. WebShop is a realistic e-commerce simulation containing over 1.1 million prod-

ucts (Yao et al., 2022). Agents must navigate through search and button-click interactions to purchase items satisfying specific user requirements. The benchmark evaluates open-ended exploration and decision making in large-scale semi-structured web environments.

To rigorously evaluate the generalizability and robustness of the STARK framework, we conduct experiments across five diverse benchmarks, including three text-based environments (ALFWorld, ScienceWorld, and WebShop) and two multimodal interactive benchmarks: **Sokoban** and **EZPoints**. In these multimodal settings, the agent relies solely on raw RGB observations paired with brief textual instructions.

Sokoban (6×6). Sokoban is a classic grid-based puzzle game widely used for evaluating spatial reasoning and long-term planning (Schrader, 2018). Within a 6×6 maze, the agent must push all boxes onto designated target locations.

- **Observation:** At each timestep, the agent receives a top-down RGB image representing the environment state, including the spatial distribution of walls, boxes, targets, free spaces, and the agent’s own position.
- **Core Challenges:** The primary difficulty arises from **irreversible dead-ends**. Since boxes can only be pushed and not pulled, a single erroneous move may render the puzzle permanently unsolvable. This irreversibility requires deep forward-looking exploration before committing to actions.

EZPoints (Gym Cards). EZPoints is a task from the Gym Cards suite introduced in RL4VLM (Zhai et al., 2024). It is a visual reasoning benchmark requiring tight coupling between visual perception and symbolic mathematical reasoning. Given a set of playing cards, the agent must construct an arithmetic expression step-by-step such that the final result matches a target value (typically 12).

- **Observation:** The agent observes an RGB image displaying the card faces together with a status bar representing the current arithmetic expression state.
- **Core Challenges:** Unlike text-based environments, EZPoints requires sophisticated **visual-symbolic alignment**. The agent must accurately recognize card values from pixels while performing multi-step arithmetic reasoning without intermediate textual guidance.

C.2 Baselines

We evaluate STARK against a broad range of representative baselines spanning prompting-based agents, reinforcement learning methods, and strong closed-source LLMs.

(1) Closed-source LLMs:

- **GPT-4o (OpenAI, 2024):** A natively multi-modal large language model capable of integrating text, image, and audio understanding within a unified architecture, enabling strong interactive reasoning and real-time decision making.

- **GPT-5-mini (OpenAI, 2025):** A lightweight variant of the GPT-5 family designed for efficient reasoning and language understanding under reduced computational budgets.
- **GPT-5 (OpenAI, 2025):** A frontier large language model optimized for long-context reasoning, complex planning, and multi-domain problem solving.
- **Gemini-2.5-Pro (Comanici et al., 2025):** An advanced multimodal reasoning model developed by Google DeepMind with strong capabilities in long-context understanding, planning, and tool-augmented reasoning.

(2) Prompting Methods:

- **ReAct (Yao et al., 2023):** A prompting framework that interleaves reasoning traces with executable actions, enabling language models to interact with external environments through explicit thought-action loops.
- **Reflexion (Shinn et al., 2023):** A self-reflective prompting strategy that enables agents to revise future decisions based on verbal feedback and historical failures without parameter updates.

(3) RL Training Methods:

- **PPO (Schulman et al., 2017):** A classical actor-critic reinforcement learning algorithm that optimizes policies through clipped policy gradients and value estimation.
- **RLOO (Ahmadian et al., 2024):** A critic-free reinforcement learning approach that replaces explicit value models with leave-one-out baselines for policy optimization.
- **ETO (Song et al., 2024):** An exploration-oriented training paradigm that improves agent behaviors through trial-and-error trajectory optimization and contrastive success-failure supervision.
- **GRPO (Shao et al., 2024):** A group-relative policy optimization framework that estimates advantages through reward normalization within sampled trajectory groups, eliminating the need for a separate critic model.
- **GiGPO (Feng et al., 2025):** A hierarchical policy optimization framework that performs

fine-grained credit assignment by grouping trajectories according to shared anchor states.

- **RLVMR** (Zhang et al., 2026b): A meta-reasoning reinforcement learning framework that introduces verifiable cognitive tags and rule-based rewards to improve long-horizon reasoning behaviors.
- **SLEA-RL** (Wang and Jiang, 2026): A step-level reinforcement learning framework that combines observation clustering with an evolving experience library for long-horizon agent optimization.
- **HCAPO** (Tan et al., 2026): A hindsight credit assignment framework that leverages post-hoc reasoning and multi-scale advantage estimation for long-horizon decision making.
- **HGPO** (He et al., 2026): A hierarchical group-based policy optimization framework that improves long-horizon credit assignment through context-consistent trajectory grouping and hierarchical variance reduction.

C.3 Evaluation Details

In our evaluation, we consistently adopt Success Rate (SR.) as the primary performance metric. The success rate measures whether an agent successfully completes a given task.

For ALFWorld and ScienceWorld, following prior long-horizon RL works (Zhang et al., 2026b), we evaluate generalization under three progressively challenging settings: L0 (seen task categories and instances), L1 (seen categories with unseen instances), and L2 (unseen categories and unseen instances).

For the WebShop environment, in addition to success rate, we also report the task completion score (Score), which reflects the degree of partial task completion. Specifically, even when an agent fails to fully satisfy all user requirements, it can still receive partial rewards for completing intermediate sub-goals during web navigation.

For multimodal evaluation, we report average success rates across Sokoban and EZPoints benchmarks using the same evaluation protocol for all methods.

For all evaluation experiments, we set the test batch size to 128 and fix the random seed to 0 to ensure reproducibility and fair comparison across methods.

For proprietary models including GPT-5, GPT-5-mini, GPT-4o, and Gemini-2.5-Pro, we evaluate all methods under the same environment interaction protocol and task constraints as open-source baselines. Specifically, all closed-source models interact with the environment using the same action space, observation format, maximum interaction horizon, and stopping criteria.

During inference, we set the sampling temperature to 0.4 and limit the maximum response length to 512 tokens per interaction step. For each task instance, we perform a single complete rollout without additional search, majority voting, or self-consistency sampling unless otherwise specified. All evaluations are conducted under identical benchmark settings to ensure fair comparison across methods.

C.4 Implementation Details

Details for SFT. To cold-start the generation of meta-cognitive reasoning tags, we construct a lightweight supervised fine-tuning dataset containing 200 trajectories for each environment, following the same low-resource setting adopted in RLVMR (Zhang et al., 2026b). We employ GPT-4o (OpenAI, 2024) to synthesize the training trajectories.

Specifically, we adopt a retro-annotation strategy: GPT-4o first reviews successful interaction trajectories and then selectively inserts four meta-cognitive tags, including <explore>, <reflection>, <planning>, and <monitor>, according to the local reasoning context and decision patterns. These tags serve as structured cognitive signals that initialize the model’s ability to emit explicit reasoning modes during reinforcement learning.

All environments use the same amount of SFT data and identical annotation protocols to ensure fair comparison across domains.

Details for RL. Response lengths are set to 512 tokens. Each episode is capped at 30 environment interaction steps. We employ a learning rate of 1×10^{-6} for the actor policy. A rule-based sparse reward system is adopted, assigning a reward of 10 for successful task completion and 0 for failure.

To encourage valid environment interactions, we apply a penalty of -0.1 for invalid actions generated by the agent. For all group-based RL methods, the total rollout budget is set to $N = 8$, the number of initial roots is set to $M = 4$, and the branching factor is set to $B = 2$. The batch size is fixed at 16.

[Branching Constraint]

In this step, your output MUST start with <reflection>.
Use <reflection>...</reflection> before any <action>.

Figure 7: Reflection-guided branching prompt in STARK.

During rollout, we use a sampling temperature of 0.4. The KL-divergence regularization coefficient is set to 0.01. To avoid context overflow in long-horizon interaction histories, we truncate the conversation memory to a maximum history length of 5 interaction turns.

For ALFWorld, ScienceWorld, and WebShop, the maximum prompt lengths are set to 2,048, 4,096, and 6,000 tokens, respectively.

The prompts are shown in Figures 7 and 8.

Meta-reasoning reward. We compute the step-level meta-reasoning reward r^{MR} using the same rule-based verifier as RLVMR (Zhang et al., 2026b). The reward is initialized as zero and assigned according to the emitted meta-cognitive tag and environment feedback. Invalid actions receive no positive meta-reasoning reward. A <planning> step is rewarded only on successful trajectories, with a discount for repeated replanning. An <explore> step is rewarded when it induces a non-redundant state-action transition. A <reflection> step is rewarded when it follows a failed or invalid action and changes the subsequent decision. The <monitor> tag is used as a valid structured reasoning mode for generation and advantage grouping, without adding a separate task-specific reward rule. We use the same reward templates and hyperparameters across all text-based environments; only the action parser, validity checker, and success condition are environment-specific. STARK does not introduce a learned reward model or LLM judge, but reuses this r^{MR} for tag-aligned and anchor-aligned advantage estimation.

Computing Details. All experiments are conducted on 4 NVIDIA A100-80GB GPUs.

D Supplementary Results

D.1 Sample Efficiency

We provide detailed sample efficiency results in Tables 6 and 7. Across both ALFWorld and ScienceWorld, STARK achieves competitive or superior performance even when trained with only a

Method	20%	40%	60%	80%	100%
GRPO	22.7	37.5	46.1	61.8	76.6
GiGPO	22.7	41.4	68.8	76.6	86.7
RLVMR	68.8	71.1	86.7	86.7	89.1
STARK	71.8	75.0	81.3	91.4	93.8

Table 6: **Sample efficiency comparison on ALFWorld L0.** We report success rates (%) on Qwen2.5-1.5B under varying training data percentages.

Method	20%	40%	60%	80%	100%
GRPO	0.0	2.3	3.1	9.0	21.1
GiGPO	1.6	1.9	10.2	21.9	25.8
RLVMR	30.4	39.6	43.4	47.6	46.9
STARK	31.2	39.1	40.6	56.2	64.8

Table 7: **Sample efficiency comparison on ScienceWorld L0.** We report success rates (%) on Qwen2.5-1.5B under varying training data percentages.

small fraction of the data. Notably, with 20% of the training data, STARK already reaches 71.8% on ALFWorld L0 and 31.2% on ScienceWorld L0, substantially outperforming GRPO and GiGPO. As the data ratio increases, STARK further improves and obtains the best full-data performance on both benchmarks, demonstrating stronger sample efficiency and better utilization of additional interaction data.

Furthermore, Table 8 reveals that the action sequences generated by STARK are significantly shorter than those of all baselines on unseen data (e.g., reduced to 15.6 steps in ScienceWorld L2). These metrics are consistent with our hypothesis: precise resource allocation not only reduces redundant sampling on routine steps but also prevents meaningless trial-and-error in erroneous directions through forward-looking branch pruning.

D.2 Additional Ablation Study

Branching Strategy. Table 9 further compares different branching strategies on ALFWorld. For the entropy-based variant, we replace the token-confidence term C_{token} in Eq. 2 with a normalized entropy-derived confidence score computed from

System Prompt for STARK

[Context & State]

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

Your task is: {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}

Your current observation is: {current_observation}

Your admissible actions of the current situation are: [{admissible_actions}]

Your previous overall plan is: {planning}

Please strictly adhere to your plan.

[Reasoning Strategies]

You must first reason using **ONLY ONE** of the following reasoning modes.

<planning>

Plan or replan the entire task by breaking it down into high-level steps.

Focus on outlining the full sequence required to complete the overall task, rather than only deciding the immediate next action.

Use this at the beginning of complex tasks, or whenever the previous plan is incorrect or insufficient.

The plan should list key points separately, e.g., step 1: ..., step 2: ..., step 3: ...

</planning>

<explore>

When results are unexpected or information is lacking, use current observations to think outside the box and list as many possible locations, items, or actions as possible.

Use this approach when facing obstacles that require creative and innovative thinking.

</explore>

<reflection>

Analyze the reasons for errors in task execution and correct them by exploring alternative approaches.

The message "No known action matches that input." indicates that the action is invalid.

This mode is typically used when several consecutive actions yield no substantial progress.

</reflection>

<monitor>

Continuously track the current progress and history of reasoning and execution throughout the task.

Recall the current subgoal and consider the next concrete action, ensuring the agent remains aligned with the overall plan.

Typically used when task outcomes are as expected and no other reasoning mode is required.

</monitor>

[Output Format]

Now it is your turn to first reason and then take an action.

1. First, reason about the current situation in one concise, brief sentence.

This reasoning process **MUST** be enclosed within **ONLY ONE** of the reasoning tag pairs above, such as:

`<planning>...</planning>`

`<explore>...</explore>`

`<reflection>...</reflection>`

`<monitor>...</monitor>`

2. Once you have finished your reasoning, choose an admissible action for the current step and present it within:

`<action>...</action>`

Figure 8: System prompt for RL in STARK.

Environment	Level	Vanilla	SFT	GRPO	GiGPO	RLVMR	STARK
ALFWorld	Seen-L0	28.8	23.2	15.8	<u>12.6</u>	10.8	<u>12.6</u>
	Unseen-L1	29.1	24.5	18.1	14.7	11.6	<u>13.0</u>
	Unseen-L2	28.9	27.5	21.7	19.4	15.4	<u>16.2</u>
ScienceWorld	Seen-L0	25.8	22.9	15.4	14.7	12.5	<u>13.5</u>
	Unseen-L1	27.9	25.7	26.7	25.2	<u>18.8</u>	16.2
	Unseen-L2	26.8	26.2	27.6	26.3	<u>20.5</u>	15.6

Table 8: **Average action sequence length across environments and generalization levels.** Lower values indicate higher execution efficiency. **Bold** and underline denote the shortest and second shortest lengths, respectively.

Method	L0	L1	L2
1.5B Entropy (Shannon Entropy)	93.8	91.4	52.0
1.5B STARK (Confidence)	93.8	89.1	73.4
1.5B Deterministic (w/o Bernoulli)	93.8	89.1	60.2

Table 9: **Ablation of branching strategies on ALF-World (1.5B).** Success rates (%; higher is better) compare confidence-based, entropy-derived, and deterministic branching under the same rollout budget.

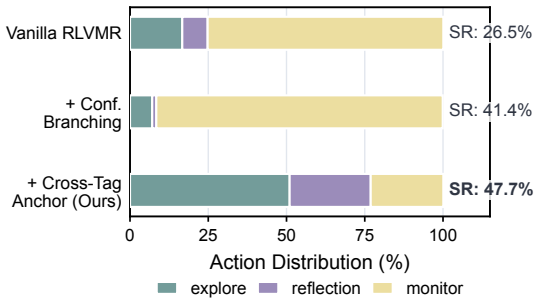


Figure 9: **Action distribution vs. task success on ScienceWorld L2.** Vanilla RLVMR and confidence branching overuse the safe monitor action while underusing explore and reflection. Cross-tag anchoring restores a more balanced action distribution and achieves the highest success rate (SR).

the policy’s token distribution during generation, while keeping the remaining confidence formulation in Eq. 3 unchanged. Specifically, Shannon entropy is first computed over the vocabulary distribution and then normalized and inverted so that larger values still indicate higher confidence. For the deterministic variant, denoted as w/o Bernoulli, we remove the Bernoulli sampling step and always branch at the decision node with the highest branching probability under the same rollout budget.

Compared with Shannon entropy and deterministic selection, our confidence-based method matches the best performance on L0 and achieves the best performance on L2, showing stronger robustness

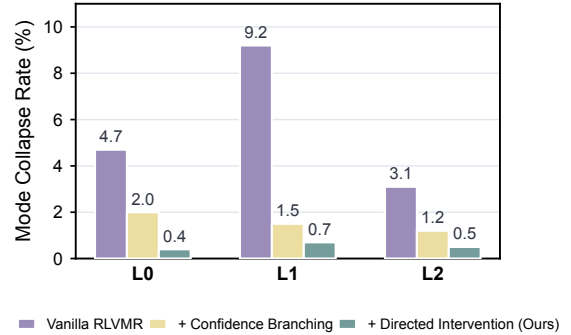


Figure 10: **Ablation on cognitive mode collapse in ScienceWorld.** Vanilla RLVMR and confidence branching remain vulnerable to repetitive exploratory loops; directed intervention reduces mode collapse to below 1.0% across all generalization levels.

especially under the most challenging L2 setting.

Analysis of meta-cognitive behaviors. Figure 9 reveals a critical failure mode in baseline models: they tend to find shortcuts by over-exploiting the low-risk monitor action (accounting for up to 91.5%), while abandoning high-cost but highly valuable explore and reflection processes. After introducing STARK’s anchor-aligned advantage, the proportion of explore healthily rebounds to 51.0%, and reflection to 25.9%, ultimately yielding the highest task success rate (47.7%). This further supports our design rationale: an effective reward mechanism must proactively correct distortions in the action distribution to incentivize the agent to engage in authentic, deep meta-cognitive reasoning.

Cognitive Mode Collapse. Figure 10 ablates the contribution of each component to reducing cognitive mode collapse on ScienceWorld. Vanilla RLVMR reaches a mode collapse rate of 9.2% at L1, and confidence branching alone cannot fully resolve this. With directed intervention, STARK

A trajectory of an agent trained with GRPO	A trajectory of an agent trained with STARK
Task: cool some bowl and put it in cabinet. [Step 1] Observation: -= Welcome to TextWorld, ALFRED! -= Reasoning: <think> I need to find a bowl first before cooling it. </think> Action: go to diningtable 1 [Step 2] Observation: On the diningtable, there is a bowl 1. Reasoning: <think> The bowl is on the dining table, so I should pick it up. </think> Action: take bowl 1 from diningtable 1 [Step 5] Observation: The cabinet is open. Reasoning: <think> The bowl is inside the closed cabinet... </think> [Hallucinated world state] Action: close cabinet 1 [Step 11] Observation: The fridge is open. Reasoning: <think> We need to find a cooler somewhere in the room to cool the bowl. </think> Action: cool bowl 1 with fridge 1 [Step 18] Observation: The agent repeatedly revisits countertop, diningtable, cabinet, and fridge. Reasoning: <think> The bowl is currently in a closed cabinet. </think> Action: go to countertop 1 [hallucinated memory and repetitive exploration] [Steps 20–30: Stuck in a Loop] The agent repeatedly switches between the fridge, countertop, diningtable, and cabinet while generating contradictory beliefs. The policy fails to maintain a consistent task state and never completes the final placement step. Failure	Task: cool some bowl and put it in cabinet. [Step 1] Observation: -= Welcome to TextWorld, ALFRED! -= Reasoning: <explore> The bowl might be on the dining table, so I should search there first. </explore> Action: go to diningtable 1 [Step 2] Observation: On the diningtable, there is a bowl 1. Reasoning: <explore> The bowl is found, so I should pick it up. </explore> Action: take bowl 1 from diningtable 1 [Step 3] Observation: The bowl is now in inventory. Reasoning: <monitor> My current sub-goal is to cool the bowl, so I need to go to the fridge. </monitor> [stable sub-goal tracking] Action: go to fridge 1 [Step 5] Observation: The fridge is open. Reasoning: <explore> I can now cool the bowl using the fridge. </explore> Action: cool bowl 1 with fridge 1 [Step 7] Observation: The cabinet is closed. Reasoning: <monitor> My next sub-goal is to place the cooled bowl into the cabinet. </monitor> Action: open cabinet 1 [Step 10] Observation: The placement action initially fails. Reasoning: <reflection> The bowl still needs to be moved into the cabinet explicitly. </reflection> [Failure recovery] Action: move bowl 1 to cabinet 1 success

Figure 11: Qualitative comparison on the embodied planning task (ALFWorld): “cool some bowl and put it in cabinet.” GRPO suffers from hallucinated world states, repetitive exploration, and inconsistent task tracking, eventually becoming trapped in a behavioral loop. In contrast, STARK maintains stable sub-goal execution through the <monitor> mode and rapidly recovers from failed actions via reflective correction.

Method	ALFWorld			WebShop
	L0	L1	L2	Score
SPARK	96.1	92.2	88.3	89.8
STARK (Ours)	96.9	96.1	90.6	92.1
Δ vs. SPARK	+0.8	+3.9	+2.3	+2.3

Table 10: **Comparison with branching methods (7B).** We compare STARK with SPARK, a strong branching-based baseline, on ALFWorld and WebShop.

compresses the mode collapse rate to below 1.0% across all generalization levels, confirming that prompt-injected reflective branching is the key driver of behavioral diversity.

Comparison with Branching Methods. To further assess whether STARK’s gains come only from branching, we compare it with SPARK (Wu et al., 2026b), a strong branching-based agentic RL baseline, using the same 7B backbone. As shown in Table 10, STARK achieves stronger performance across all ALFWorld generalization levels and obtains a higher WebShop score. These results suggest that branching itself is an important component for long-horizon exploration, as evidenced by the strong performance of SPARK. More importantly, the additional gains of STARK indicate that branching becomes more effective when coupled

Configuration	L0	L2
STARK (Full)	64.8	47.7
w/o Reflection	61.7	32.8
w/o Anchor Advantage	57.8	39.8
w/o Reflection & Anchor	53.1	41.4
w/o Branching & Reflection & Anchor	46.9	26.5

Table 11: **Extended component ablation on ScienceWorld (1.5B).** Success rates (%; higher is better) isolate the reflection intervention and anchor-aligned advantage individually and jointly; confidence-based branching is retained unless explicitly removed.

with dual-track advantage estimation: confidence-driven branches provide locally comparable counterfactual samples, while strategic and tactical advantages convert these samples into finer-grained learning signals for both state-grounded execution and meta-reasoning mode selection.

Extended Component Isolation and Random-Branching Control. We further disentangle confidence-driven branching, reflection intervention, and anchor-aligned advantage estimation. All variants use the same 1.5B backbone, rollout budget, and evaluation protocol. Table 11 reports the component-wise results on ScienceWorld.

Removing all three components reduces performance to 46.9% on L0 and 26.5% on L2, whereas confidence branching alone raises the results to 53.1% and 41.4%, respectively. The single-

Branching Strategy	L0	L2
Confidence Branching (STARK)	93.8	73.4
Random Branching	82.0	45.3

Table 12: **Random-branching control on ALFWorld (1.5B)**. Success rates (%; higher is better) are measured under the same branching budget, reflection intervention, and dual-track advantage estimator; only branching-point selection is changed.

w_t	w_a	ω_{step}	Success Rate (%)
0.8	0.2	0.3	77.3
0.8	0.2	0.5	81.6
0.8	0.2	0.8	76.2

Table 13: **Sensitivity to the step-level fusion weight on WebShop (1.5B)**. The default setting is $\omega_{\text{step}} = 0.5$.

component controls reveal an interaction between reflection and anchor-aligned learning: each component is most effective when the other is present, and the complete configuration performs best on both splits. Their effects should therefore not be interpreted as independently additive. Confidence branching determines where counterfactual samples are generated, reflection guides how an alternative branch reconsiders the current decision, and anchor-aligned advantage determines how the resulting local comparisons contribute to learning.

To isolate branching-point selection from the additional branches and the reflection intervention, we replace confidence-based selection with uniformly random selection while retaining the same branching budget, <reflection> intervention, and dual-track advantage estimator.

Under an identical exploration and intervention budget, random branching reduces success by 11.8 points on L0 and 28.1 points on L2. The gains therefore cannot be attributed to additional branches or reflection injection alone. Confidence-based selection concentrates the limited rollout budget on low-confidence decision points, which is particularly important under the harder out-of-distribution setting.

D.3 Sensitivity Analysis

We examine the sensitivity of STARK to the step-level fusion weight ω_{step} in Eq. 9. The analysis is conducted on WebShop with the 1.5B backbone while fixing $w_t = 0.8$ and $w_a = 0.2$.

The default value $\omega_{\text{step}} = 0.5$ achieves the best performance. Increasing or decreasing this weight retains success rates above 76%, showing

that performance does not collapse under substantial reweighting. The optimum at 0.5 also indicates that balancing trajectory-level outcomes with step-level strategic and tactical signals is preferable to either over- or under-weighting the latter.

E Case Study

Qualitative Analysis. Figure 11 illustrates long-horizon reasoning behaviors on an ALFWorld object-manipulation task. The GRPO-trained agent quickly develops hallucinated world states and repeatedly revisits the same locations, eventually becoming trapped in a repetitive behavioral loop without completing the final placement objective. In contrast, STARK maintains consistent sub-goal tracking through the <monitor> mode, enabling stable execution across multiple interaction steps. Moreover, when the initial placement action fails, the agent activates <reflection> to perform corrective reasoning and rapidly recover task progress. These results demonstrate that explicitly decomposing exploration, execution monitoring, and reflective correction substantially improves long-horizon planning stability and robustness.

Trajectory Visualization. Figures 12–14 provide illustrative examples.

F Additional Discussion

F.1 Relation to Prior Agentic RL Methods

STARK is closely related to three recent lines of work on long-horizon LLM agent training: state-aligned credit assignment, verifiable meta-reasoning supervision, and branching-based exploration. We emphasize that STARK is not a simple aggregation of these components, but a unified framework in which confidence-driven branching and dual-track advantage estimation are mutually dependent.

GiGPO (Feng et al., 2025) improves group-based RL by constructing step-level groups from repeated environment states. This state-anchor mechanism provides a stronger local baseline than trajectory-level normalization and reduces variance caused by comparing actions taken under different environment difficulties. However, GiGPO treats actions under the same state primarily as execution alternatives and does not explicitly model the agent’s high-level cognitive mode. As a result, it cannot directly answer whether the agent should plan, explore, reflect, or monitor at a given decision

point. STARK inherits the value of state-aligned comparison, but extends it from action-level execution comparison to cognition-aware strategic comparison. By combining anchor-aligned groups with meta-reasoning tags, STARK estimates not only whether an action is locally effective, but also which reasoning mode is preferable under the same physical observation.

RLVMR (Zhang et al., 2026b) introduces verifiable meta-reasoning tags and dense process-level rewards, enabling RL to supervise cognitive behaviors such as planning, exploration, and reflection. Nevertheless, its tag-based grouping does not guarantee that compared reasoning steps are grounded in the same environment state. In long-horizon interaction, trajectories from the same prompt can quickly diverge into very different physical situations; normalizing rewards within the same tag may therefore mix reasoning quality with state difficulty. STARK addresses this limitation by making meta-reasoning evaluation state-conditioned. The proposed strategic-and-tactical advantage estimator separates two questions that are entangled in prior tag-based methods: how well a cognitive behavior is executed, and which cognitive mode should be selected at a specific state.

SPARK (Wu et al., 2026b) is a representative branching-based baseline for long-horizon agentic RL, where dynamic branching is used to improve rollout allocation over passive linear sampling. STARK adopts a different use of branching. Rather than treating branches only as additional exploratory trajectories, STARK uses them to form locally comparable counterfactual control sets. Multiple branches are expanded from the same physical state with directed meta-cognitive intervention, making it possible to compare different reasoning modes under aligned environmental observations. These branched samples are further coupled with dual-track advantage estimation, allowing the model to receive state-grounded supervision on when to switch cognitive modes in addition to optimizing trajectory-level returns.

Therefore, the central novelty of STARK lies in the explicit coupling between targeted exploration and topology-aware credit assignment. Confidence-driven branching is not used only to obtain more diverse trajectories; it is designed to generate locally comparable counterfactual samples from the same physical state. This provides the structural basis for evaluating alternative meta-cognitive modes under aligned environmental observations. The anchor-

aligned strategic advantage then converts these local counterfactual comparisons into learning signals for mode selection, while the tag-aligned tactical advantage captures the execution quality of each reasoning mode across trajectories.

This design creates a loop between exploration and optimization. Branching provides the local variation required for meaningful cross-mode comparison; dual-track advantage estimation converts that variation into supervision on both which cognitive mode to choose and how well to execute it. Without branching, anchor-aligned groups may be too sparse to support reliable mode-level comparison. Without dual-track advantage estimation, additional branches mainly serve as search trajectories and do not directly teach the policy when to plan, explore, reflect, or monitor. In this sense, STARK differs from methods that improve only one side of the problem: state-aligned credit assignment, meta-reasoning reward design, or dynamic exploration. STARK unifies these elements into a single training objective in which targeted exploration produces the comparison structure required for fine-grained credit assignment in long-horizon agents.

F.2 Compatibility with Alternative Policy Optimizers

The two core mechanisms of STARK operate independently of the specific policy-update rule. Confidence-driven branching changes the data-collection process by determining where counterfactual trajectories are generated. The dual-track estimator changes the advantages assigned to the collected samples, but does not alter the form of the subsequent policy-update objective.

Our current implementation combines the dual-track advantage in Eq. 9 with the standard GRPO clipped surrogate. More generally, the same strategic and tactical advantage terms can augment a base advantage supplied by another policy optimization method. For PPO (Schulman et al., 2017), the base term may be obtained from a critic-based estimator; for RLOO (Ahmadian et al., 2024), it may be obtained from the leave-one-out trajectory baseline. In both cases, the optimizer-specific clipping, importance ratios, regularization, and update schedule can remain unchanged.

This separation makes the dual-track estimator modular at the policy-objective level. The present experiments instantiate STARK with GRPO, so they do not establish empirical superiority under every optimizer. Evaluating its interaction with critic-

based and alternative critic-free optimization methods is an important direction for future work.

F.3 Discussion on Intervention Tag Selection

STARK uses a fixed <reflection> prefix for branched trajectories because branching is triggered specifically at low-confidence decision nodes, where the agent is more likely to suffer from invalid actions, repetitive exploration, or inconsistent state tracking. In this setting, reflection provides a targeted corrective counterfactual rather than merely increasing rollout diversity. This design also keeps the intervention mechanism simple and interpretable: the gains can be attributed to confidence-driven corrective branching and dual-track advantage estimation, rather than to an additional learned tag-selection policy.

Learned or random tag selection are natural alternatives, but they introduce different trade-offs. A learned selector could be beneficial in later training stages, but early in RL it may inherit the policy’s existing mode bias and reproduce the same collapsed reasoning pattern that STARK aims to break. Random tags may increase surface-level diversity, but often create branches whose cognitive mode is mismatched with the low-confidence trigger, weakening the quality of anchor-aligned comparisons. We therefore adopt fixed reflection as a conservative and controlled intervention, while leaving adaptive intervention selection as a promising extension for future work.

F.4 Failure Mode Analysis

STARK is designed to improve exploration and credit assignment in long-horizon agent learning, but its behavior also reveals two boundary cases. First, confidence estimation may miss some risky states when the policy assigns high probability to a fluent but suboptimal action. In such cases, branching is not triggered even though an alternative reasoning path could be useful. This does not undermine the confidence-driven design, but suggests that future confidence estimators could incorporate richer signals beyond token probability and action validity.

Second, observation-based topology anchors may be imperfect in partially observable environments, where visually or textually similar observations can still differ in hidden history, inventory state, or downstream constraints. STARK mitigates this issue by using anchors as local comparison groups rather than assuming full state equivalence,

Method	Seed 0	Seed 1	Seed 2	Mean $\pm$ Std
GRPO	76.6	75.0	75.7	75.8 $\pm$ 0.8
GiGPO	86.7	85.9	86.7	86.4 $\pm$ 0.5
RLVMR	89.1	88.3	89.8	89.1 $\pm$ 0.8
STARK	93.8	92.2	92.2	92.7 $\pm$ 0.9

Table 14: **Multi-seed robustness on ALFWorld L0 (1.5B)**. Results are success rates (%); “Std” denotes the sample standard deviation over three seeds.

but richer state abstractions could further improve the precision of strategic advantage estimation. These boundary cases clarify where the current framework can be extended, while preserving the central contribution of coupling targeted branching with topology-aligned credit assignment.

F.5 Multi-Seed Robustness

We evaluate representative methods on ALFWorld L0 with the 1.5B backbone using three random seeds. Apart from the random seed, all runs use the same training configuration, rollout budget, and evaluation protocol.

All methods exhibit small seed-level variation, with standard deviations below 1.0. STARK consistently outperforms every baseline under each seed and improves the mean success rate over the strongest baseline, RLVMR, by 3.6 points.

F.6 Future Work

One direction is to extend STARK from text-based environments to perceptual GUI, web, and embodied agents. GUI-focused semi-online and rule-based RL, as explored by UI-S1 (Lu et al., 2025) and UI-R1 (Lu et al., 2026a), provide natural testbeds for confidence-driven branching over structured actions. Perceptual signals could further inform the branching criterion, for example through hallucination-aware routing across specialized visual experts (Wang et al., 2025). Beyond task success, TIDE and OdysseyArena motivate evaluating trajectory-level adaptation dynamics and long-horizon inductive behavior (Yan et al., 2026a,b).

A second direction is to combine topology-aligned credit with richer auxiliary supervision. Gated self-distillation (Lu et al., 2026b), skill internalization (Lu et al., 2026c), step-aware preference distillation (Zhang et al., 2026a), hierarchical agent-tool delegation (Zhang, 2025), and reference-guided trace construction (Yang et al., 2026) could convert local counterfactual branches into reusable skills and denser decision-level sig-

nals. A central challenge is to preserve same-state comparability while keeping auxiliary supervision aligned with the current policy distribution.

Finally, branching should become safety-aware for deployment. Risk taxonomies and diagnostic criteria from web- and embodied-agent evaluations ([Ying et al., 2026a,b](#)) could be operationalized as auxiliary signals for the confidence trigger and branch-selection constraints. This extension would prompt corrective alternatives at both uncertain and security-sensitive states while jointly measuring task utility and violation rates.

A trajectory of an agent trained with STARK

Task: put some spraybottle on toilet

[Step 1]

Observation: -= Welcome to TextWorld, ALFRED! -=

Reasoning: <explore> The spray bottle might be on the countertops, so I need to search these areas first.</explore>

Action: go to countertop 1

[Step 2]

Observation: You arrive at countertop 1. On the countertop 1, you see nothing

Reasoning: <explore>The spray bottle is not on countertop 1, so I need to search other locations.</explore>

Action: go to countertop 2

[Step 3]

Observation: You arrive at countertop 2. On the countertop 2, you see a cloth 3, a spraybottle 2, and a tissuebox 1

Reasoning: <explore>The spray bottle is on countertop 2, so I need to pick it up.</explore>

Action: take spraybottle 2 from countertop 2

[Step 4]

Observation: You pick up the spraybottle 2 from the countertop 2

Reasoning: <monitor>Currently, my sub-goal is to place the spray bottle on the toilet, so I need to go to toilet 1.</monitor>

Action: go to toilet 1

[Step 5]

Observation: You arrive at toilet 1. On the toilet 1, you see a soapbar 1, and a toiletpaper 1.",

Reasoning: <monitor>Currently, my sub-goal is to place the spray bottle on the toilet, so I need to put it on the toilet 1.</monitor>

Action: put spraybottle 2 in/on toilet 1

[Step 6]

Observation: Nothing happens.

Reasoning: <reflection>Currently, my sub-goal is to place the spray bottle on the toilet, so I need to examine it to check if it's empty.</reflection>

Action: examine spraybottle 2

[Step 7]

Observation: There's nothing special about spraybottle 2.

Reasoning: <reflection>Currently, my sub-goal is to place the spray bottle on the toilet, so I need to move it to the toilet 1.</reflection>

Action: move spraybottle 2 to toilet 1

Figure 12: A full trajectory of STARK in ALFWorld (Example 1).

Task: determine if glass jar is electrically conductive. The glass jar is located around the workshop. First, focus on the glass jar. If it is electrically conductive, place it in the yellow box. If it is electrically nonconductive, place it in the purple box.

[Step 1]

Observation: This room is called the hallway. In it, you see: \n\tthe agent\n\ta substance called air\n\ta painting\nYou also see:.....

Reasoning: <explore>To find the glass jar, I need to explore the different rooms in the house.</explore>

Action: open door to workshop

[Step 2]

Observation: The door is already open.

Reasoning: <explore>To find the glass jar, I need to explore the workshop where the glass jar might be located.</explore>

Action: go to workshop

■ ■ ■ ■ ■ ■ ■ ■ ■

[Step 7]

Observation: No known action matches that input.

Reasoning: <reflection>Reflecting on the current state, I need to reassess the progress and ensure the task is completed accurately.</reflection>

Action: look around

[Step 8]

Observation: This room is called the workshop. In it, you see: \n\tthe agent\n\ta substance called air\n\ta electric buzzer,
which is off\n\ta glass jar....

Reasoning: <monitor>Currently, my sub-goal is to test the conductivity of the glass jar. I need to connect the battery anode to the green wire terminal 1.</monitor>

Action: connect battery anode to green wire terminal 1

[Step 9]

Observation: anode on battery is now connected to terminal 1 on green wire

Reasoning: <monitor>Currently, my sub-goal is to test the conductivity of the glass jar. I need to connect the battery cathode to the orange wire terminal 1.</monitor>

Action: connect battery cathode to orange wire terminal 1

[Step 10]

Observation: cathode on battery is now connected to terminal 1 on orange wire

Reasoning: <monitor>Currently, my sub-goal is to test the conductivity of the glass jar. I need to connect the green wire terminal 2 to the cathode in the orange light bulb.</monitor>

Action: connect green wire terminal 2 to cathode in orange light bulb

[Step 11]

Observation: terminal 2 on green wire is now connected to cathode on orange light bulb

Reasoning: <monitor>Currently, my sub-goal is to test the conductivity of the glass jar. I need to connect the orange wire terminal 2 to the anode in the orange light bulb.</monitor>

Action: connect orange wire terminal 2 to anode in orange light bulb

[Step 12]

Observation: terminal 2 on orange wire is now connected to anode on orange light bulb

Reasoning: <monitor>Currently, my sub-goal is to test the conductivity of the glass jar. I need to connect the glass jar terminal 1 to the red wire terminal 1.</monitor>

Action: connect glass jar terminal 1 to red wire terminal 1

[Step 13]

Observation: No known action matches that input.

Reasoning: <reflection>Reflecting on the current state, I have not found a valid action to continue the task based on the observation.</reflection>

Action: look around

[Step 14]

Observation: This room is called the workshop. In it, you see: \nOn the table is: a battery, a black wire, a green wire, a orange light bulb, which is on.....

Reasoning: <monitor>Currently, my sub-goal is to determine if the glass jar is electrically conductive. Since the glass jar is nonconductive, I need to move it to the purple box.</monitor>

Action: move glass jar to purple box

Figure 13: A full trajectory of STARK in ScienceWorld (Example 2).

A trajectory of an agent trained with STARK

Task: Find me home office furniture sets for dining room, living room with color: navy-multi, and item shape: round, and size: 7 ft 8 in x 10 ft 7 in, and price lower than 80.00 dollars

[Step 1]

Observation: Search

Reasoning: <planning>First, I need to search for home office furniture sets for dining room, living room with color: navy-multi, and item shape: round, and size: 7 ft 8 in x 10 ft 7 in, and price lower than 80.00 dollars.</planning>

Action: search[home office furniture sets for dining room, living room with color navy-multi, and item shape round, size 7 ft 8 in x 10 ft 7 in, price <80.00]

[Step 2]

Observation: ""Back to Search' [SEP] 'Page 1 (Total results: 50)' [SEP] 'Next >' [SEP] 'B07FKGQKZ1' [SEP] 'Home Dynamix Lyndhurst Rotana Modern Area Rug, Contemporary Blue/Gray/Ivory 5'2"x7'4\" [SEP] '\$44.99 to \$33.98' [SEP] 'B0943TLW3L'

Reasoning: <planning>Currently, I need to refine my search to find the home office furniture set that matches the specifications provided.</planning>

Action: click[b07fkqkz1]

[Step 3]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] '1 ft 9 in x 7 ft 2 in' [SEP][SEP] 'Price: \$44.99 to \$33.98' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'"

Reasoning: <planning>Currently, I need to refine my search to find the specific home office furniture set with the desired shape and size. I will use the 'round' shape as per the observation.</planning>

Action: click[navy-multi]

[Step 4]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] '1 ft 9 in x 7 ft 2 in' [SEP] '3 ft 7 in x 5 ft 2 in' [SEP] '5 ft 2 in x 7 ft 4 in' [SEP][SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'"

Reasoning: <monitor>Currently, I need to refine my search for the home office furniture set with the specified dimensions and color. I will use the 'size' option to confirm the shape and dimensions.</monitor>

Action: click[7 ft 8 in x 10 ft 7 in]

[Step 5]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] '1 ft 9 in x 7 ft 2 in' [SEP][SEP] 'Price: \$44.99 to \$33.98' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'"

Reasoning: <monitor>Currently, my sub-goal is to find the home office furniture set for the dining room. I have already found the set, and now I need to proceed to the 'buy now' option to proceed with the purchase.</monitor>

Action: click[navy-multi]

[Step 6]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] '1 ft 9 in x 7 ft 2 in' [SEP] '3 ft 7 in x 5 ft 2 in' [SEP][SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'"

Reasoning: <planning>Currently, I need to refine the search to find the specific item with the desired size and color.</planning>

Action: click[round]

[Step 7]

Observation: 'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] '1 ft 9 in x 7 ft 2 in' [SEP] '3 ft 7 in x 5 ft 2 in' [SEP] '5 ft 2 in x 7 ft 4 in' [SEP][SEP] 'Price: \$44.99 to \$33.98' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'"

Reasoning: <monitor>Currently, my sub-goal is to find the perfect home office furniture set for the dining room, and I have already found the desired product. I need to proceed to the 'buy now' option to finalize the purchase.</monitor>

Action: click[buy now]

Figure 14: A full trajectory of STARK in WebShop (Example 3).