DISTAR: DISENTANGLED TOKEN AGGREGATION FOR REASONING AND ACTION IN LLM AGENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning (RL) has been pivotal in empowering autonomous agents to solve long-horizon, multi-turn tasks. However, most existing methods directly inherit standard token aggregation paradigms in single-turn reasoning, thereby overlooking the fundamental divergence between reasoning and action: *reasoning* requires thorough exploration within a single turn, whereas *action* demands concise and precise sequences to prevent error propagation. In this paper, we first identify and formalize this "aggregation dilemma" in agentic RL: applying token-mean to action tokens inadvertently assigns higher gradient weights to longer, redundant action chains, while enforcing sequence-mean on reasoning tokens hinders the model's Chain-of-Thought (CoT) exploration. To resolve this conflict, we propose **DisTAR** (**Dis**entangled **T**oken **A**ggregation for **R**easoning and Action), a novel framework that decouples the token aggregation paradigm. Specifically, DisTAR enforces a token-mean constraint on reasoning tokens to encourage sufficient reasoning before action, while executing a mathematically equivalent sequence-mean reweighting mechanism on action tokens to optimize for concise action chains. Furthermore, at the trajectory level, an inverse-step penalty is integrated into the advantage estimation to proactively suppress redundant paths. Finally, we introduce a globally normalized surrogate loss to maintain stable training convergence under this heterogeneous token aggregation strategy. Extensive experiments on WebShop, AlfWorld, and SearchQA across multiple foundation models (e.g., Qwen-2.5, Qwen-3) demonstrate that DisTAR significantly boosts agent performance. Notably, on WebShop with Qwen-3-4B, DisTAR enhances the success rate of the baseline GRPO from 68.6% to 82.8%, while consistently inducing shorter, more precise action chains alongside richer CoT. Code is available at https://anonymous.4open.science/r/DisTAR-A387.

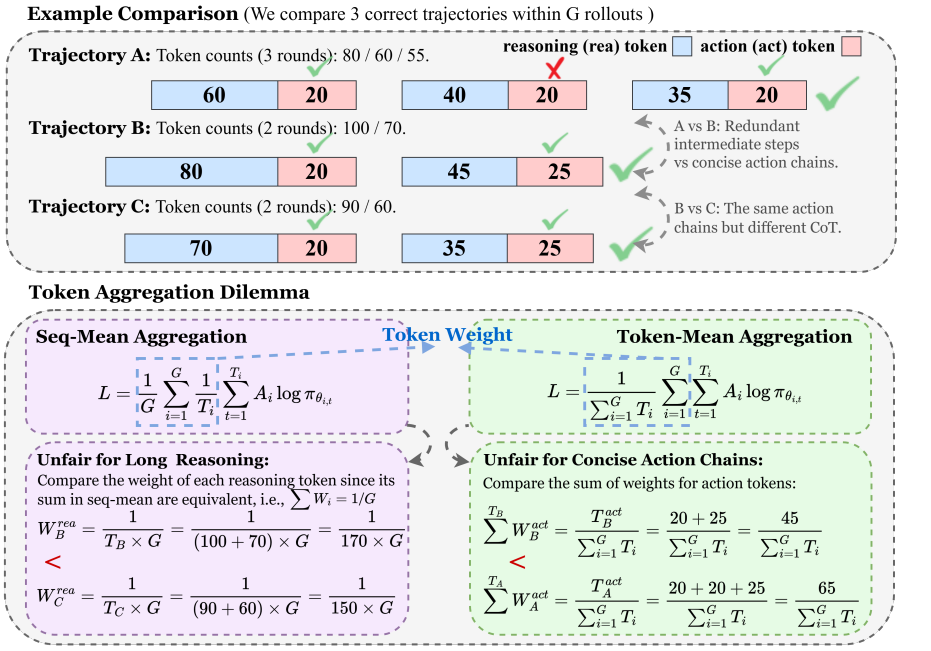
1 INTRODUCTION

As Large Language Models (LLMs) increasingly serving as autonomous agents that integrate reasoning (Yang et al., 2025; Guo et al., 2025; Kumar et al., 2025), planning (Qiao et al., 2024; Song et al., 2023; Zhu et al., 2026a), and tool invocation (Zheng et al., 2025; Feng et al., 2025; Qian et al., 2026) to accomplish multi-turn, long-horizon tasks, Reinforcement Learning (RL) (Shao et al., 2024; Schulman et al., 2017; Ouyang et al., 2022; Wen et al., 2025) has emerged as the standard post-training paradigm for sequential decision-making in interactive environments. In such agentic RL settings, the sequential coupling of multi-turn reasoning, tool execution, and state feedback presents unique scaling and optimization bottlenecks. To mitigate these challenges, recent research has actively pioneered specialized optimization methods that focus on efficient trajectory selection (Dong et al., 2025b;a; Guo et al., 2026b; Zhou et al., 2026a) and precise credit assignment (Feng et al., 2026; Fang et al., 2026; Tan et al., 2026; Tu et al., 2026).

Despite the rapid progress of agentic RL methods, most frameworks directly inherit traditional token aggregation paradigms from single-turn reasoning, such as token-mean (Yu et al., 2026a) or sequence-mean aggregation (Shao et al., 2024). Although some recent methods recognize the behavioral differences between reasoning and action, they address these differences mainly through reweighting or disjoint adaptation, yet still retain a shared aggregation strategy (He et al., 2026; Li et al., 2026b). Consequently, existing aggregation mechanisms overlook the distinct requirements of reasoning and action: *reasoning* demands extensive intra-turn exploration through Chain-of-Thought (CoT) to thoroughly comprehend the state space, whereas *action* represents a deterministic execution process where conciseness and precision are paramount. Consequently, forcing these two operationally distinct behaviors into an identical token aggregation scheme leads to suboptimal policy optimization.

In this paper, we formalize this conflict as the **aggregation dilemma**. As illustrated in Figure 1, consider a scenario where the model generates three successful trajectories within a rollout group of size G. We analyze these trajectories to reveal how different token aggregation strategies shape the optimization dynamics. When *sequence-mean* aggrega-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.



L: Loss function; A_i : Advantage; G: Rollout number; T_i: Sequence length; π_θ: Policy model; W: Token weight

Figure 1: **Illustration of the Token Aggregation Dilemma in agentic RL.** Within the same action chains, Seq-Mean Aggregation (left) unfairly favors trajectories with shorter CoT (Trajectory B vs. C) due to sequence-length normalization. In contrast, with the different action chains, Token-Mean Aggregation (right) biases the loss toward redundant trajectories (Trajectory A vs. B) by over-weighting longer action chains, showing that reasoning and action tokens require a disentangled token aggregation strategy.

tion is applied, longer Chain-of-Thought (CoT) reasoning is downweighted: as shown in the left part of Figure 1, the summed token weights for trajectory B are lower than for trajectory C, thereby reducing the incentive for the model to deliberate before acting. In contrast, redundant tool interactions are unintentionally amplified as *token-mean* aggregation is adopted: Comparing Trajectory A and B in the right part of Figure 1, longer action sequences of A receive larger optimization weights merely due to containing more action tokens, rather than contributing more effective behaviors. This introduces an optimization bias toward unnecessarily extended action chains, even when shorter trajectories accomplish the same objective. Therefore, existing aggregation strategies inevitably face a fundamental trade-off as detailed in section 3.3: encouraging deeper reasoning tends to promote inefficient tool usage, while favoring concise action sequences risks compromising reasoning quality.

To resolve the aggregation dilemma, we propose **DisTAR** (**Dis**entangled **T**oken **A**ggregation for **R**easoning and Action), a unified policy optimization framework that explicitly disentangles the optimization of reasoning and action tokens. Specifically, DisTAR applies *token-mean* aggregation to reasoning tokens, encouraging the model to perform more thorough deliberation before action. In contrast, action tokens are optimized through a mathematically equivalent *sequence-mean* reweighting to promote concise and efficient actions. Importantly, this disentangled optimization is implemented within a single backward pass, avoiding the computational overhead of separately optimizing the two token streams. Furthermore, we introduce an *inverse-step advantage penalty* to explicitly discourage redundant action chains, encouraging agents to solve tasks with fewer interactions and reducing cascading errors. Finally, to address the heterogeneous token weights introduced by our disentangled optimization, we develop a *globally normalized surrogate loss* that normalizes the effective optimization signal, resulting in more stable and reliable policy learning.

We evaluate **DisTAR** across three mainstream agent benchmarks: WebShop (Yao et al., 2022), AlfWorld (Shridhar et al., 2020), and SearchQA. Extensively utilizing multiple foundation models, including `Qwen2.5-1.5B/3B/7B` and `Qwen3-4B`, our experiments reveal that DisTAR consistently achieves substantial performance gains over conventional single token aggregation baselines, establishing a new state-of-the-art (SOTA). For instance, on the SearchQA multi-domain tasks, DisTAR enhances the average accuracy from 43.2% to 46.4% compared to $\mathbf{T}^2$PO (Wang et al., 2026). Furthermore, on WebShop with `Qwen3-4B`, DisTAR boosts the success rate of the baseline GRPO from 68.6% to 82.8%. Our empirical outcomes consistently validate that DisTAR unlocks a complementary synergy: it induces

shorter, more precise action execution chains while simultaneously fostering richer, higher-quality CoT deliberation. Our primary contributions are summarized below.

- We identify and formalize the "aggregation dilemma" in Agentic RL, proving that standard unified token aggregation methods establish conflicting optimization goals for exploratory reasoning and efficient action execution.
- We propose DisTAR, a novel disentangled token aggregation strategy implemented within a single backward pass, which applies a token-mean constraint on reasoning tokens to foster sufficient exploration and executes a mathematically equivalent sequence-mean reweighting on action tokens to promote concise action chains.
- We incorporate an inverse-step penalty to compress redundant steps, thereby boosting overall accuracy by proactively preventing cascading errors, while designing a globally normalized surrogate loss to stabilize training under this heterogeneous token aggregation.
- Extensive experiments demonstrate that DisTAR consistently achieves substantial performance gains over conventional single-token aggregation strategies and delivers state-of-the-art results across multiple agent benchmarks.

2 Related Work

2.1 Agentic Reinforcement Learning

Reinforcement learning (RL) has become central to improving LLM reasoning and alignment (Yu et al., 2026a; Zhou et al., 2026b; Yu et al., 2026b). Building on PPO-based RLHF (Liang et al., 2025; Zhu et al., 2026b; Zhou et al.), recent critic-free methods like GRPO (Shao et al., 2024) estimate advantages via group-relative comparisons and refine clipping granularity using verifiable rewards. Extending this foundation, current agentic RL methods largely tackle two key challenges: multi-turn credit assignment (Chen et al., 2026; Wang et al., 2026; Li et al., 2026a; Lin et al., 2026) and trajectory selection (Zhao et al., 2026; Xu et al., 2026). For credit assignment, the sparse or delayed rewards in multi-turn agent-environment interactions make it difficult to attribute final outcomes to individual steps. For example, GIGPO (Feng et al., 2026) introduces a two-level structure, i.e., episode-level advantages and step-level advantages via anchor state grouping—to attribute sparse final rewards to individual turns without auxiliary models. For trajectory selection, a key challenge is efficiently allocating limited rollout budgets to the most critical decision points. To address this, ARPO (Dong et al., 2025b) employs an entropy-based adaptive rollout mechanism that dynamically branches at high-uncertainty steps following tool interactions.

However, most methods retain a shared token aggregation scheme for reasoning and action. Although some recent methods recognize the two behavioral differences, they primarily address them through reweighting or disjoint adaptation (He et al., 2026; Li et al., 2026b), leaving their distinct requirements unaddressed at the aggregation level.

2.2 Token Aggregation in Reinforcement Learning

The token aggregation scheme plays a critical role in policy optimization, as it determines how token-level gradients are aggregated into the global training objective. GRPO (Shao et al., 2024) first introduced the *Sequence-Mean Aggregation* paradigm, which normalizes each trajectory's cumulative loss by its sequence length, ensuring balanced optimization among sampled candidates regardless of their lengths. Subsequently, DAPO (Yu et al., 2026a) proposed *Token-Mean Aggregation*, which averages losses over all valid tokens in a batch. By removing per-sequence normalization and assigning equal optimization weight to individual tokens, this strategy implicitly favors longer reasoning trajectories. Most existing agentic RL methods adopt either token-mean (Yu et al., 2026a; 2025; He et al., 2026) or sequence-mean (Zheng et al., 2025; Feng et al., 2025; Hong et al., 2025; Guo et al., 2026a) aggregation when optimizing long-horizon trajectories, while largely overlooking the distinct roles of different token types.

However, agentic trajectories inherently consist of heterogeneous components, where reasoning tokens and action tokens contribute differently to task completion. Applying a uniform aggregation rule across them creates a fundamental mismatch between optimization and functionality, motivating the need for **disentangled token aggregation** in agentic reinforcement learning.

3 PRELIMINARIES

3.1 TASK DEFINITION

We model the multi-turn agentic RL process of one trajectory as a Markov Decision Process (MDP) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, P, R \rangle$ (Sutton et al., 1998), where states $s \in \mathcal{S}$ represent interaction histories, and actions $a \in \mathcal{A}$ denote environment-facing operations. The transition function $P(s_{k+1} \mid s_k, a_k)$ models the environment dynamics, while the reward function $R(\tau)$ evaluates the full trajectory at termination. At each turn $k \in \{1, \ldots, K\}$, the policy π_θ conditions on the current state s_k to generate a structured response:

$$o_k = \langle \texttt{think} \rangle \ldots \langle / \texttt{think} \rangle \langle \texttt{action} \rangle a_k \langle / \texttt{action} \rangle, \tag{1}$$

where the $\langle \texttt{think} \rangle \ldots \langle / \texttt{think} \rangle$ block encloses the inner reasoning process, and the $\langle \texttt{action} \rangle$ block contains the executable action a_k. Upon receiving o_k, the environment extracts a_k, transitions to s_{k+1}, and yields a terminal scalar reward $R(\tau)$ when the episode concludes. Formally, a complete trajectory is denoted as $\tau = (s_1, o_1, s_2, o_2, \ldots, s_K, o_K)$, and the RL objective is to maximize the expected episodic return: $\mathcal{J}(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[R(\tau)]$.

3.2 TOKEN AGGREGATION PARADIGM

To update the policy π_θ using sequence-level reward under the GRPO framework, we consider two mainstream paradigms to aggregate token gradients across a group of G sampled outputs:

Sequence-mean Aggregation. This scheme assigns equal weight to each generated sequence by normalizing individual token gradients by the sequence length T_i. This prevents overly verbose reasoning traces from dominating the policy update:

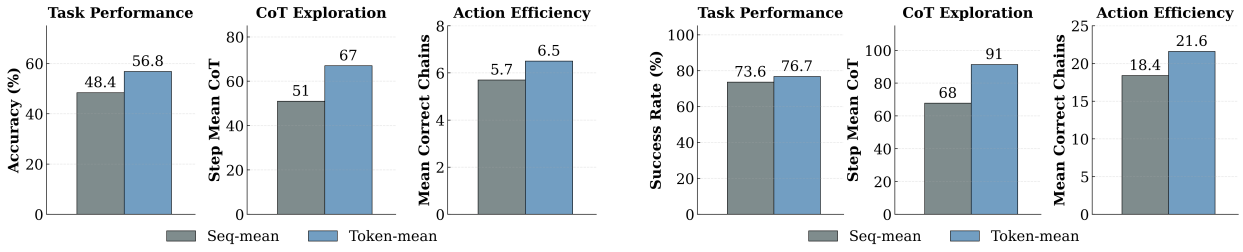
$$\mathcal{L}_{\text{seq-mean}}(\theta) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{T_i} \sum_{t=1}^{T_i} A_i \log \pi_{\theta_{i,t}}, \tag{2}$$

where A_i denotes the relative advantage of the i-th sequence within the group, and $\pi_{\theta_{i,t}}$ represents the action probability at the t-th token step. Note that here we focus on the effect of token aggregation and omit the clipping and KL-regularization terms in GRPO for simplicity.

Token-mean Aggregation. This strategy treats all tokens across the entire group equally, which implicitly biases the optimization towards longer sequences that contain extensive reasoning steps:

$$\mathcal{L}_{\text{token-mean}}(\theta) = \frac{1}{\sum_{i=1}^{G} T_i} \sum_{i=1}^{G} \sum_{t=1}^{T_i} A_i \log \pi_{\theta_{i,t}}. \tag{3}$$

3.3 PILOT EXPERIMENT ON TOKEN AGGREGATION



(a) WebShop (b) AlfWorld

Figure 2: **Empirical comparison of standard token aggregation paradigms across multi-turn agent benchmarks.** We compare Seq-mean and Token-mean aggregation on (a) WebShop and (b) ALFWorld in terms of task performance, reasoning effort, and action-execution efficiency.

To investigate the behavioral divergence induced by different token aggregation, we conduct a pilot experiment with two mainstream paradigms: *Seq-mean* and *Token-mean* aggregation. As illustrated in Figure 2, we comprehensively evaluate the agents across three dimensions: overall task performance, Chain-of-Thought (CoT) exploration depth, and action chain execution efficiency.

Overall Performance Comparison. Experimental results on WebShop and AlfWorld show that Token-mean consistently outperforms Seq-mean. Specifically, Token-mean achieves 56.8% accuracy on WebShop (vs. 48.4% for Seq-mean) and 76.7% success rate on AlfWorld (vs. 73.6% for Seq-mean). These outcomes suggest that uniform token-level weighting generally provides a better optimization landscape than sequence-level averaging for interactive tasks, which is consistent with previous works (Yu et al., 2025; Feng et al., 2026; Zhang, 2026).

Token-level Analysis. A deeper trajectory analysis reveals the pattern distinction: As shown in Figure 2. For Token-mean, it generates more reasoning tokens per step than Seq-mean (67 vs. 51 on WebShop; 91 vs. 68 on AlfWorld), indicating more extensive deliberation before action selection. This richer reasoning process likely contributes to its stronger performance. However, Token-mean also requires more interaction steps in successful trajectories (6.5 vs. 5.7 on WebShop; 21.6 vs. 18.4 on AlfWorld), suggesting greater redundancy in action execution. For Seq-mean, although its results are inferior to Token-mean, it produces more concise successful trajectories with fewer interaction steps, which may reduce exposure to execution errors and yield higher efficiency (Wang et al., 2026; Xu et al., 2026). Overall, these results reveal a clear trade-off between reasoning depth and action efficiency: Token-mean favors richer CoT exploration at the cost of behavioral redundancy, whereas Seq-mean improves action conciseness but constrains reasoning exploration.

The Aggregation Dilemma. These empirical insights expose the fundamental limitation of standard uniform token aggregation paradigms. While Token-mean fosters intensive intra-turn reasoning necessary for complex task planning, its uniform weighting leads to potential redundant action sequences. Conversely, Seq-mean compresses interaction steps but restricts the CoT exploration required for correct decision-making before action. Neither paradigm is globally optimal: forcing reasoning exploration and action execution into a single uniform aggregation strategy creates an optimization dilemma, directly motivating our disentangled token aggregation framework.

4 METHODOLOGY

In this section, we present DisTAR, a novel framework to resolve the reasoning-action aggregation dilemma in agent training. Firstly, we propose the disentangled token aggregation to tackle the optimization conflict between reasoning and action. Secondly, to explicitly suppress path redundancy at the trajectory level, we incorporate an inverse-step advantage penalty directly into the generalized advantage estimation. Thirdly, to handle the potential gradient variance arising from our heterogeneous token aggregation, we develop a globally normalized surrogate loss to stabilize training.

4.1 DISENTANGLED TOKEN AGGREGATION

Existing agentic RL methods typically adopt a unified token aggregation strategy, applying either token-mean or sequence-mean uniformly across the entire trajectory. However, as discussed in Figure 1, reasoning and action tokens exhibit fundamentally different optimization characteristics. Reasoning tokens correspond to an open-ended exploration process, where longer Chain-of-Thought (CoT) trajectories often encode richer state understanding and should therefore be encouraged. In contrast, action tokens correspond to environment interactions, where redundant tool invocations unnecessarily extend trajectories and increase the risk of cascading errors. To reconcile these conflicting objectives, we propose a *disentangled token aggregation* strategy that optimizes reasoning and action tokens under different aggregation principles.

For reasoning tokens, we adopt token-level aggregation, where each reasoning token contributes equally to the optimization objective. Specifically, let $\mathcal{R}_i$ denote the set of reasoning tokens in the i-th trajectory. The reasoning objective is defined as

$$\mathcal{L}_{\text{reason}}(\theta) = \frac{1}{\sum_{i=1}^{G} T_i} \sum_{i=1}^{G} \sum_{t \in \mathcal{R}_i} A_i \log \pi_{\theta_{i,t}}. \tag{4}$$

This formulation assigns uniform optimization weight to reasoning tokens, preserving the exploration-friendly property of token-level aggregation and encouraging richer reasoning.

For action tokens, however, directly applying token-level aggregation introduces an undesirable bias toward longer tool trajectories, since trajectories containing more action tokens receive larger cumulative optimization signals. Let $\mathcal{A}_i$ denote the set of action tokens. To eliminate the dependence on action sequence length, we instead optimize action

tokens using sequence-mean aggregation:

$$\mathcal{L}_{\text{action}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{A}_i|}\sum_{t\in\mathcal{A}_i} A_i \log \pi_{\theta_{i,t}}. \tag{5}$$

A straightforward implementation of Eq. (4) and Eq. (5) requires explicitly separating reasoning and action streams, optimizing their objectives independently, and maintaining separate computational graphs. This decoupled optimization introduces additional backward passes and memory overhead, which becomes increasingly costly for long-horizon agent trajectories with extensive reasoning and tool interactions. To overcome this limitation, we derive a reweighting formulation that embeds the sequence-mean normalization of action tokens into a unified token-level objective. This enables joint optimization of reasoning and action tokens with a single backward pass while preserving their respective aggregation properties. Specifically, we define the action weight as

$$\alpha_i = \frac{\sum_{j=1}^{G} T_j}{G|\mathcal{A}_i|}, \tag{6}$$

which compensates for the trajectory-length bias introduced by token-level aggregation. The action objective can then be equivalently rewritten as

$$\mathcal{L}_{\text{action}}(\theta) = \frac{1}{\sum_{i=1}^{G} T_i}\sum_{i=1}^{G}\sum_{t\in\mathcal{A}_i} \alpha_i A_i \log \pi_{\theta_{i,t}}, \tag{7}$$

where shorter action trajectories receive larger token weights, while longer trajectories receive smaller weights. This reweighting removes the length dependency of action optimization and recovers the sequence-mean aggregation behavior (proof in section A.1). However, Eq. 7 is normalized within the action-token space and does not explicitly align the optimization scale between reasoning and action tokens. To ensure consistent contribution across the two token types, we further introduce a global scaling factor that maps the action objective into the same normalization space as the full trajectory:

$$\beta_i = \frac{\sum_{j=1}^{G} T_j}{G|\mathcal{A}_i|} \times \frac{\sum_{j=1}^{G} |\mathcal{A}_j|}{\sum_{j=1}^{G} T_j} = \frac{\sum_{j=1}^{G} |\mathcal{A}_j|}{G|\mathcal{A}_i|}. \tag{8}$$

Consequently, the entire objective can be unified into a single token-level loss:

$$\mathcal{L}_{\text{DisTAR}}(\theta) = \frac{1}{\sum_{i=1}^{G} T_i}\sum_{i=1}^{G}\sum_{t=1}^{T_i} \omega_{i,t} A_i \log \pi_{\theta_{i,t}}, \tag{9}$$

where the token-dependent weight is defined as

$$\omega_{i,t} = \begin{cases} 1, & t \in \mathcal{R}_i, \\ \beta_i, & t \in \mathcal{A}_i. \end{cases} \tag{10}$$

With this disentangled aggregation strategy, reasoning tokens preserve the exploration-friendly property of token-level aggregation, whereas action tokens become invariant to trajectory length and follow the sequence-mean optimization principle. DisTAR therefore mitigates the aggregation dilemma by simultaneously encouraging richer deliberation and more concise action trajectories within a single optimization pass.

4.2 Inverse-step Advantage Penalty

Although disentangled aggregation removes the optimization bias caused by token weighting, it does not explicitly encourage efficient trajectories. In practice, redundant tool interactions often increase the risk of error propagation and degrade the final success rate. Therefore, we further introduce an inverse-step penalty that directly biases optimization toward concise action sequences.

Let S_i denote the action chains length in trajectory τ_i. We define an efficiency-aware advantage as

$$\tilde{A}_i = \frac{A_i}{S_i}, \tag{11}$$

where A_i is the original trajectory-level advantage.

The final policy objective becomes

$$\mathcal{L}_{\text{DisTAR}} = \frac{1}{\sum_{i=1}^{G} T_i} \sum_{i=1}^{G} \sum_{t=1}^{T_i} \omega_{i,t}\, \tilde{A}_i \log \pi_{\theta_{i,t}}, \tag{12}$$

where $\omega_{i,t}$ denotes the disentangled token weight defined in Eq. (10). This formulation can be viewed as an implicit trajectory-level regularizer. For two trajectories with identical rewards, the shorter one receives a larger optimization signal. Consequently, the policy is encouraged to solve tasks with fewer interactions, thereby suppressing redundant tool usage and reducing cascading errors.

4.3 Globally Normalized Surrogate Loss

The proposed disentangled aggregation and inverse-step penalty jointly induce heterogeneous token-level contributions during policy optimization. Specifically, each token (i, t) is assigned a weight

$$\omega_{i,t}^{all} = \begin{cases} 1/S_i, & \text{if } (i,t) \in \mathcal{R}_i, \\ \beta_i/S_i, & \text{if } (i,t) \in \mathcal{A}_i, \end{cases} \tag{13}$$

where S_i denotes the number of action steps in trajectory τ_i, $|\mathcal{A}_i|$ is the total number of action tokens, and $\beta_i = (\sum_{j=1}^{G} |\mathcal{A}_j|)/G|\mathcal{A}_i|$ denotes the action reweighting constant.

Since both the trajectory length S_i and the action-token count M_i can vary across samples, the resulting token weights may differ significantly between trajectories and mini-batches. Directly optimizing a weighted sum, $\sum_{i,t} \omega_{i,t} \mathcal{L}_{i,t}$, may cause the overall gradient magnitude to fluctuate with the raw weight sum, yielding inconsistent optimization signals and occasionally unstable training. To address this issue, we normalize the surrogate objective by the total token weight, transforming the weighted sum into a weighted average. The final policy objective is given by

$$\mathcal{L}(\theta) = -\frac{1}{\sum_{i=1}^{G} \sum_{t=1}^{T_i} \omega_{i,t}^{all}} \sum_{i=1}^{G} \sum_{t=1}^{T_i} \omega_{i,t} \max\Big(-\rho_{i,t}(\theta)\tilde{A}_i,\ -\text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\tilde{A}_i \Big), \tag{14}$$

where $\rho_{i,t}(\theta) = \pi_\theta(y_{i,t})/\pi_{\theta_{\text{old}}}(y_{i,t})$ is the PPO importance ratio, and $\tilde{A}_i$ denotes the query-level normalized advantage, which is shared by all tokens within trajectory i.

By explicitly dividing by the total effective weight, the expected gradient scale becomes approximately invariant to the composition of reasoning and action tokens in each mini-batch. Importantly, the global normalization does not alter the relative weighting structure induced by disentangled token aggregation. Instead, it only rescales the aggregated objective to stabilize gradient magnitude. As a result, the optimization remains sensitive to the ratio between reasoning and action token contributions, while being invariant to batch-level fluctuations in total token weight.

5 Experiments

5.1 Experimental Setup

Datasets. We comprehensively evaluate **DisTAR** across three representative agent benchmarks: (1) **ALFWorld** (Shridhar et al., 2020), an embodied environment covering six distinct text-based household tasks (Pick, Look, Clean, Heat, Cool, and Pick2); (2) **WebShop** (Yao et al., 2022), a realistic e-commerce simulation requiring web navigation, product search, and multi-turn purchase decision-making; and (3) **Search-augmented QA**, incorporating single-hop datasets (*i.e.*, NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), PopQA (Mallen et al., 2023)) and multi-hop environments (*i.e.*, HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), Bamboogle (Press et al., 2023)).

Implementation Details. We adopt Qwen2.5-1.5B/3B/7B-Instruct and Qwen3-4B as backbone models and implement DisTAR on top of GRPO with the step-level advantage estimator of (Feng et al., 2026). For ALFWorld and WebShop, all methods are trained using identical hyperparameter configurations for fair comparison, with the rollout group size set to $N = 8$. For search-augmented QA, we follow the training setup of Search-R1 (Jin et al., 2025), employing E5 (Wang et al., 2022) as the retriever, a rollout group size of $N = 5$, and a maximum interaction of four turns. Additional implementation details and hyperparameters are provided in section A.2. We also discuss the limitations in section A.5.

Table 1: Performance on ALFWorld and WebShop multi-turn agent benchmarks. Results are averaged over 3 random seeds.

Type	Method	ALFWorld							WebShop	
		Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Closed-Source Model										
Prompting	GPT-4o (Achiam et al., 2023)	75.3	60.8	31.2	56.7	21.6	49.8	48.0	31.8	23.7
Prompting	Gemini-2.5-Pro (Comanici et al., 2025)	92.8	63.3	62.1	69.0	26.6	58.7	60.3	42.5	35.9
Qwen2.5-1.5B-Instruct										
Prompting	Qwen2.5 (Yang et al., 2025)	5.9	5.5	3.3	9.7	4.2	0.0	4.1	23.1	5.2
RL Training	PPO (with critic) (Schulman et al., 2017)	$64.8_{\pm 3.5}$	$40.5_{\pm 6.9}$	$57.1_{\pm 4.9}$	$60.6_{\pm 6.6}$	$46.4_{\pm 4.0}$	$47.4_{\pm 1.9}$	$54.4_{\pm 3.1}$	$73.8_{\pm 3.0}$	$51.5_{\pm 2.9}$
RL Training	GRPO (Shao et al., 2024)	$85.3_{\pm 1.5}$	$53.7_{\pm 8.0}$	$84.5_{\pm 6.8}$	$78.2_{\pm 7.9}$	$59.7_{\pm 5.0}$	$53.5_{\pm 5.6}$	$72.8_{\pm 3.6}$	$75.8_{\pm 3.5}$	$56.8_{\pm 3.8}$
RL Training	GiGPO (Feng et al., 2026)	$94.4_{\pm 5.9}$	$67.5_{\pm 4.6}$	$94.8_{\pm 3.8}$	$94.4_{\pm 7.8}$	$79.8_{\pm 4.7}$	$76.4_{\pm 5.4}$	$86.7_{\pm 1.7}$	$83.1_{\pm 1.6}$	$65.0_{\pm 3.2}$
RL Training	$\mathbf{T^2}$PO (Wang et al., 2026)	$95.2_{\pm 4.8}$	$83.3_{\pm 4.4}$	$94.4_{\pm 2.8}$	$95.0_{\pm 5.2}$	$82.3_{\pm 3.4}$	$\mathbf{78.8}_{\pm 2.0}$	$89.8_{\pm 1.2}$	$86.4_{\pm 1.3}$	$69.2_{\pm 2.6}$
RL Training	**DisTAR**	$\mathbf{97.6}_{\pm 4.4}$	$\mathbf{87.8}_{\pm 3.8}$	$\mathbf{97.8}_{\pm 4.2}$	$\mathbf{97.2}_{\pm 3.8}$	$\mathbf{86.4}_{\pm 5.4}$	$76.5_{\pm 6.7}$	$\mathbf{92.3}_{\pm 1.4}$	$\mathbf{90.8}_{\pm 1.8}$	$\mathbf{76.5}_{\pm 3.4}$
Qwen3-4B-RFT										
Prompting	Qwen3 (Yang et al., 2025)	25.6	63.5	18.3	18.7	24.3	10.1	26.8	25.2	5.9
RL Training	PPO (with critic) (Schulman et al., 2017)	$83.3_{\pm 9.5}$	$75.1_{\pm 6.3}$	$74.5_{\pm 7.9}$	$62.6_{\pm 1.6}$	$84.2_{\pm 0.0}$	$58.3_{\pm 7.4}$	$75.4_{\pm 3.8}$	$70.3_{\pm 8.6}$	$61.9_{\pm 5.9}$
RL Training	GRPO (Guo et al., 2025)	$85.3_{\pm 6.8}$	$64.6_{\pm 4.3}$	$91.2_{\pm 0.8}$	$90.2_{\pm 7.1}$	$73.9_{\pm 9.6}$	$60.2_{\pm 5.3}$	$77.4_{\pm 0.6}$	$80.0_{\pm 7.9}$	$68.6_{\pm 4.1}$
RL Training	GiGPO (Feng et al., 2026)	$87.9_{\pm 8.9}$	$77.3_{\pm 8.4}$	$88.0_{\pm 6.9}$	$86.9_{\pm 4.3}$	$79.1_{\pm 4.7}$	$71.4_{\pm 7.1}$	$80.5_{\pm 2.4}$	$86.0_{\pm 4.2}$	$73.8_{\pm 3.0}$
RL Training	T^2PO (Wang et al., 2026)	$97.4_{\pm 6.9}$	$87.8_{\pm 4.9}$	$98.3_{\pm 2.8}$	$85.1_{\pm 7.6}$	$85.8_{\pm 2.6}$	$\mathbf{80.4}_{\pm 1.3}$	$90.2_{\pm 1.4}$	$93.8_{\pm 0.2}$	$81.6_{\pm 0.4}$
RL Training	**DisTAR**	$\mathbf{98.8}_{\pm 4.2}$	$\mathbf{90.9}_{\pm 5.8}$	$\mathbf{98.8}_{\pm 3.6}$	$\mathbf{95.6}_{\pm 4.8}$	$\mathbf{88.9}_{\pm 5.2}$	$78.8_{\pm 3.2}$	$\mathbf{93.2}_{\pm 1.8}$	$\mathbf{94.4}_{\pm 1.6}$	$\mathbf{82.8}_{\pm 2.4}$

5.2 Experimental Results

Table 2: Performance on search-augmented QA tasks. † and ⋆ indicate in-domain and out-of-domain datasets, respectively.

		Single-Hop QA			**Multi-Hop QA**				
Type	Method	NQ†	TriviaQA⋆	PopQA⋆	HotpotQA†	2Wiki⋆	MuSiQue⋆	Bamboogle⋆	**Avg.**
Qwen2.5-3B-Instruct									
RL Training	Search-R1 (Jin et al., 2025)	34.1	54.5	37.8	32.4	31.9	10.3	26.4	32.5
RL Training	ZeroSearch (Sun et al., 2025)	41.4	57.4	44.8	27.4	30.0	9.8	11.1	31.7
RL Training	StepSearch (Wang et al., 2025)	-	-	-	34.5	32.0	**17.4**	34.4	-
RL Training	GiGPO (Feng et al., 2026)	42.0	59.5	42.4	36.9	37.0	12.6	64.1	42.1
RL Training	T^2PO (Wang et al., 2026)	43.1	59.8	43.6	37.5	39.2	14.3	65.0	43.2
RL Training	**DisTAR**	**46.4**	**62.6**	**47.5**	**42.5**	**41.1**	15.5	**69.4**	**46.4**
Qwen2.5-7B-Instruct									
RL Training	Search-R1 (Jin et al., 2025)	39.3	61.0	39.7	37.0	40.1	14.6	36.8	38.5
RL Training	ZeroSearch (Sun et al., 2025)	43.6	61.8	51.5	34.6	35.2	18.4	27.8	39.1
RL Training	StepSearch (Wang et al., 2025)	-	-	-	38.6	36.6	**22.6**	40.0	-
RL Training	GiGPO (Feng et al., 2026)	46.4	64.7	46.1	41.6	43.6	18.9	68.9	47.2
RL Training	T^2PO (Wang et al., 2026)	47.8	65.5	47.2	42.0	44.8	20.4	70.3	48.3
RL Training	**DisTAR**	**50.6**	**68.9**	**52.3**	**47.2**	**46.8**	22.0	**73.2**	**51.6**

Performance on ALFWorld and WebShop. Table 1 shows that DisTAR consistently achieves the best performance across both benchmarks and model scales. With Qwen2.5-1.5B-Instruct, DisTAR improves the overall ALFWorld success rate from 72.8% to 92.3% and the WebShop success rate from 56.8% to 76.5% compared with GRPO. On Qwen3-4B-RFT, DisTAR further increases the overall ALFWorld success rate from 77.4% to 93.2% and the WebShop success rate from 68.6% to 82.8%. The consistent improvements validate our hypothesis that encouraging richer reasoning while suppressing redundant actions leads to more effective agent training, as shown in Figure 3b.

Performance on QA tasks. Table 2 presents the results on search-augmented QA benchmarks. DisTAR consistently achieves the best average performance across both model scales. On Qwen2.5-3B-Instruct, DisTAR improves the average score from 43.2 to 46.4, surpassing the strongest T^2PO by 3.2 points. On Qwen2.5-7B-Instruct, DisTAR further improves the average performance from 48.3 to 51.6. Notably, DisTAR achieves substantial gains on both in-domain (NQ and HotpotQA) and out-of-domain datasets such as Bamboogle, demonstrating strong generalization ability.

5.3 Analytical Experiments

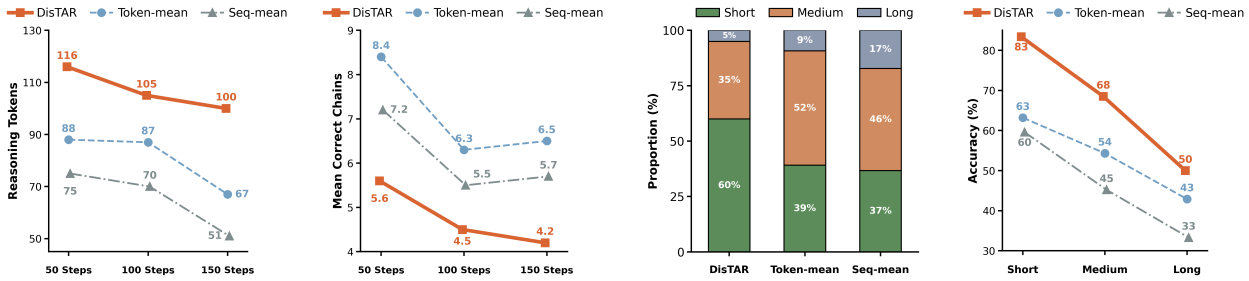
Ablation Studies. As summarized in Table 3, all three core modules jointly underpin the efficacy of DisTAR. Specifically, removing Disentangled Token Aggregation triggers the sharpest performance drop—reducing Task Score to 84.8 and Success Rate to 66.2. This underscores our central hypothesis that uniform token treatment causes subop-

timal credit assignment, whereas decoupling reasoning and action optimization is essential. Furthermore, discarding the Inverse-step Advantage Penalty primarily degrades the Success Rate ($76.5 \rightarrow 73.5$), demonstrating that penalizing operational redundancy directly improves decision efficiency. Lastly, omitting the Globally Normalized Surrogate Loss leads to higher variance (69.8 ± 7.2), confirming its vital role in stabilizing optimization under heterogeneous token weights. More comprehensive ablation studies can be found in Appendix.

Table 3: Ablation study of key modules on WebShop.

Key Modules	**Task Score**	**Success Rate**
w/o *Disentangled Token Aggregation*	$84.8_{\pm 2.2}$	$66.2_{\pm 3.6}$
w/o *Inverse-step Advantage Penalty*	$88.6_{\pm 1.6}$	$73.5_{\pm 2.4}$
w/o *Globally Normalized Surrogate Loss*	$86.6_{\pm 4.8}$	$69.8_{\pm 7.2}$
DisTAR	$\mathbf{90.8}_{\pm 1.8}$	$\mathbf{76.5}_{\pm 3.4}$

Token Level Analysis. To verify that DisTAR achieves the desired optimization behavior, we analyze the step-mean reasoning tokens and average action length during training on WebShop with Qwen2.5-1.5B. As shown in Figure 3a and Figure 3b, compared with both token-mean and sequence-mean aggregation, DisTAR consistently generates longer reasoning trajectories while maintaining the shortest action chains of correct trajectories. At 150 training steps, DisTAR produces 100 reasoning tokens on average while reducing the average correct action length to 4.2, demonstrating that our disentangled optimization encourages sufficient deliberation without introducing redundant tool interactions. We also provide additional case studies to further demonstrate the effectiveness of DisTAR in section A.6.



(a) Step-mean reasoning (b) Mean correct chains (c) Trajectory length (d) Group-wise accuracy

Figure 3: **Token Level and Trajectory Level Analysis.** We evaluate the operational dynamics on WebShop across different dimensions, including step-mean reasoning tokens, average action length, trajectory length distribution, and group-wise accuracy.

Trajectory Level Analysis. As shown in Figure 3c, we categorize rollout trajectories from Qwen2.5-1.5B on WebShop into Short, Medium, and Long groups to analyze behavioral dynamics. We observe a consistent performance decline as trajectories become longer across all methods (e.g., Token-mean decreases from 63% to 43% in Figure 3d), highlighting the importance of trajectory efficiency in mitigating error propagation. DisTAR demonstrates two key advantages: (1) it consistently achieves the highest accuracy across all trajectory groups; and (2) it shifts a larger proportion of samples toward the Short group (60% vs. 39% for Token-mean and 37% for Seq-mean), indicating reduced operational redundancy and more concise agent behaviors.

6 Conclusion

This work argues that the optimization of reasoning and action should be explicitly disentangled in agentic reinforcement learning. While longer reasoning trajectories generally facilitate deliberation, unnecessarily long action sequences often introduce redundant environment interactions and error propagation. Motivated by this observation, we propose DisTAR, which jointly combines disentangled token aggregation, inverse-step advantage regularization, and globally normalized optimization into a unified training framework. Extensive experiments across embodied agents, web agents, and retrieval-augmented QA demonstrate that DisTAR consistently encourages richer reasoning, more concise action chains, and superior task performance.

AI USE STATEMENT

In this work, we used generative AI tools for writing polishing and language refinement. We have not used generative AI tools for research ideation, algorithm derivation, or experimental evaluation. All AI-assisted text and code were thoroughly reviewed and verified by the authors. We take full responsibility for the final content of this work.

ETHICS STATEMENT

This study strictly adheres to the ICLR Code of Ethics. The datasets utilized in our experiments are publicly available, fully anonymized, and do not involve human subjects, privacy infringement, or harmful discrimination concerns.

REPRODUCIBILITY STATEMENT

To ensure reproducibility, we provide full implementation details, hyperparameter configurations, and training scripts in the anonymized code repository submitted as supplementary material. Complete mathematical proofs for Theorem 1 are detailed in Appendix A.1.

REFERENCES

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

Dingwei Chen, Zefang Zong, Zhipeng Ma, Leo Luo, Yang Li, Chengming Li, Peng Chen, and Jie Jiang. A ̂2 tgpo: Agentic turn-group policy optimization with adaptive turn-level clipping. *arXiv preprint arXiv:2605.06200*, 2026.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, et al. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*, 2025a.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025b.

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

Jiazhan Feng, Shijue Huang, Xingwei Qu, Ge Zhang, Yujia Qin, Baoquan Zhong, Chengquan Jiang, Jinxin Chi, and Wanjun Zhong. Retool: Reinforcement learning for strategic tool use in llms. *arXiv preprint arXiv:2504.11536*, 2025.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081): 633–638, 2025.

Weiyang Guo, Zesheng Shi, Liye Zhao, Jiayuan Ma, Zeen Zhu, Junxian He, Min Zhang, and Jing Li. E3-tir: Enhanced experience exploitation for tool-integrated reasoning. 2026a.

Yifu Guo, Jiaye Lin, Huacan Wang, Yuzhen Han, Sen Hu, Ziyi Ni, Licheng Wang, and Mingguang Chen. Se-agent: Self-evolution trajectory optimization in multi-step reasoning with llm-based agents. *Advances in Neural Information Processing Systems*, 38:116314–116341, 2026b.

Langzhou He, Junyou Zhu, Yue Zhou, Zhengyao Gu, Junhua Liu, Wei-Chieh Huang, Henry Peng Zou, David Wipf, Philip S Yu, and Qitian Wu. Resolving action bottleneck: Agentic reinforcement learning informed by token-level energy. *arXiv preprint arXiv:2605.14558*, 2026.

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

Jack Hong, Chenxiao Zhao, ChengLin Zhu, Weiheng Lu, Guohai Xu, and Xing Yu. Deepeyesv2: Toward agentic multimodal model. *arXiv preprint arXiv:2511.05271*, 2025.

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

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1601–1611, 2017.

Komal Kumar, Tajamul Ashraf, Omkar Thawakar, Rao Muhammad Anwer, Hisham Cholakkal, Mubarak Shah, Ming-Hsuan Yang, Phillip HS Torr, Fahad Shahbaz Khan, and Salman Khan. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321*, 2025.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

Junbo Li, Peng Zhou, Rui Meng, Meet P Vadera, Lihong Li, and Yang Li. Turn-ppo: Turn-level advantage estimation with ppo for improved multi-turn rl in agentic llms. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 6227–6243, 2026a.

Yu Li, Mingyang Yi, Xiuyu Li, Ju Fan, Fuxin Jiang, Binbin Chen, Peng Li, Jie Song, and Tieying Zhang. Reasoning and tool-use compete in agentic rl: From quantifying interference to disentangled tuning. *arXiv preprint arXiv:2602.00994*, 2026b.

Xize Liang, Chao Chen, Shuang Qiu, Jie Wang, Yue Wu, Zhihang Fu, Hanzhu Chen, Feng Wu, and Jieping Ye. Ropo: Robust preference optimization for large language models. In *International Conference on Machine Learning*, pp. 37131–37161. PMLR, 2025.

Jianxin Lin, Chunzheng Zhu, Peter J Kneuertz, Yunfei Bai, and Yuan Xue. When models learn to ask why: Adaptive causal reasoning for trustworthy medical vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5556–5568, 2026.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

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

Ofir Press, Muru Zhang, Sewon Min, Ludwig Schmidt, Noah A Smith, and Mike Lewis. Measuring and narrowing the compositionality gap in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 5687–5711, 2023.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tur, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *Advances in Neural Information Processing Systems*, 38:105523–105553, 2026.

Shuofei Qiao, Runnan Fang, Ningyu Zhang, Yuqi Zhu, Xiang Chen, Shumin Deng, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. Agent planning with world knowledge model. *Advances in Neural Information Processing Systems*, 37:114843–114871, 2024.

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

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

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

Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 2998–3009, 2023.

Hao Sun, Zile Qiao, Jiayan Guo, Xuanbo Fan, Yingyan Hou, Yong Jiang, Pengjun Xie, Yan Zhang, Fei Huang, and Jingren Zhou. Zerosearch: Incentivize the search capability of llms without searching. *arXiv preprint arXiv:2505.04588*, 2025.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

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. Hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*, 2026.

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

Songjun Tu, Chengdong Xu, Qichao Zhang, Yiwen Ma, Yaocheng Zhang, Linjing Li, Dong Li, Xiangyuan Lan, and Dongbin Zhao. Ucob: Learning to utilize and evolve agentic skills via credit-aware on-policy bidirectional self-distillation. *arXiv preprint arXiv:2606.29502*, 2026.

Haixin Wang, Hejie Cui, Chenwei Zhang, Xin Liu, Shuowei Jin, Shijie Geng, Xinyang Zhang, Nasser Zalmout, Zhenyu Shi, and Yizhou Sun. T^2po: Uncertainty-guided exploration control for stable multi-turn agentic reinforcement learning. *arXiv preprint arXiv:2605.02178*, 2026.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*, 2022.

Ziliang Wang, Xuhui Zheng, Kang An, Cijun Ouyang, Jialu Cai, Yuhang Wang, and Yichao Wu. Stepsearch: Igniting llms search ability via step-wise proximal policy optimization. *arXiv preprint arXiv:2505.15107*, 2025.

Xumeng Wen, Zihan Liu, Shun Zheng, Shengyu Ye, Zhirong Wu, Yang Wang, Zhijian Xu, Xiao Liang, Junjie Li, Ziming Miao, et al. Reinforcement learning with verifiable rewards implicitly incentivizes correct reasoning in base llms. *arXiv preprint arXiv:2506.14245*, 2025.

Tianshi Xu, Yuteng Chen, and Meng Li. Cleaner: Self-purified trajectories boost agentic reinforcement learning. *arXiv preprint arXiv:2601.15141*, 2026.

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

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

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

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

Zhaochen Yu, Ling Yang, Jiaru Zou, Shuicheng Yan, and Mengdi Wang. Demystifying reinforcement learning in agentic reasoning. *arXiv preprint arXiv:2510.11701*, 2025.

Zhiqi Yu, Zhangquan Chen, Mengting Liu, Heye Zhang, and Liangqiong Qu. Unveiling implicit advantage symmetry: Why grpo struggles with exploration and difficulty adaptation. *arXiv preprint arXiv:2602.05548*, 2026b.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *arXiv preprint arXiv:2604.09459*, 2026.

Shitian Zhao, Shaoheng Lin, Ming Li, Haoquan Zhang, Wenshuo Peng, Kaipeng Zhang, and Chen Wei. Pyvision-rl: Forging open agentic vision models via rl. *arXiv preprint arXiv:2602.20739*, 2026.

Ziwei Zheng, Michael Yang, Jack Hong, Chenxiao Zhao, Guohai Xu, Le Yang, Chao Shen, and Xing Yu. Deepeyes: Incentivizing" thinking with images" via reinforcement learning. *arXiv preprint arXiv:2505.14362*, 2025.

Yixiao Zhou, Dongzhou Cheng, Zhiliang Wu, Yi Yang, Yu Cheng, and Hehe Fan. One refiner to unlock them all: Inference-time reasoning elicitation via reinforcement query refinement. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 38957–38978, 2026a.

Yixiao Zhou, Yang Li, Dongzhou Cheng, Hehe Fan, and Yu Cheng. Look inward to explore outward: Learning temperature policy from llm internal states via hierarchical rl. *arXiv preprint arXiv:2602.13035*, 2026b.

Zhijian Zhou, Long Li, Xuan Zhang, Zongkai Liu, Yanting Miao, Yuchen Liu, Deshu Chen, Ke Li, Xing Sun, Ruoxi Jiang, et al. Disppo: Quantile-based distributional reinforcement learning for large language models. In *Forty-third International Conference on Machine Learning*.

Chunzheng Zhu, Yangfang Lin, Shen Chen, Yijun Wang, and Jianxin Lin. Medeyes: Learning dynamic visual focus for medical progressive diagnosis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 13916–13924, 2026a.

Chunzheng Zhu, Jiaqi Zeng, Junyu Jiang, Jianxin Lin, and Yijun Wang. Medsynapse-v: Bridging visual perception and clinical intuition via latent memory evolution, 2026b. URL https://arxiv.org/abs/2604.26283.

A Appendix

A.1 Proof of Equivalence between Sequence-Mean and Reweighted Action Aggregation

We prove that the proposed reweighted token-level formulation is mathematically equivalent to the original sequence-mean objective for action tokens.

Recall the sequence-mean objective defined in Eq. (5):

$$\mathcal{L}_{\text{action}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{A}_i|}\sum_{t\in\mathcal{A}_i} A_i \log \pi_\theta(a_{i,t}|s_{i,t}), \tag{15}$$

where G denotes the number of sampled trajectories and $|\mathcal{A}_i|$ represents the number of action tokens in trajectory i.

To incorporate action tokens into a unified token-level optimization framework, we rewrite the objective using a global normalization factor:

$$\widetilde{\mathcal{L}}_{\text{action}}(\theta) = \frac{1}{N}\sum_{i=1}^{G}\sum_{t\in\mathcal{A}_i} \alpha_i A_i \log \pi_\theta(a_{i,t}|s_{i,t}), \tag{16}$$

where $N = \sum_{i=1}^{G} T_i$ denotes the total number of tokens in the rollout batch. Note that N only serves as a shared normalization constant for the unified token-level objective and does not change the intrinsic normalization property of action aggregation. The corresponding weighting coefficient is defined as:

$$\alpha_i = \frac{N}{G|\mathcal{A}_i|}. \tag{17}$$

Substituting α_i into the reweighted objective gives:

$$\begin{aligned}\widetilde{\mathcal{L}}_{\text{action}}(\theta) &= \frac{1}{N}\sum_{i=1}^{G}\sum_{t\in\mathcal{A}_i}\frac{N}{G|\mathcal{A}_i|} A_i \log \pi_\theta(a_{i,t}|s_{i,t}) \\ &= \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{A}_i|}\sum_{t\in\mathcal{A}_i} A_i \log \pi_\theta(a_{i,t}|s_{i,t}) \\ &= \mathcal{L}_{\text{action}}(\theta).\end{aligned} \tag{18}$$

Therefore, under the proposed weighting coefficient α_i, the reweighted token-level formulation exactly recovers the original sequence-mean objective. The proposed formulation can thus be viewed as a token-level realization of sequence-mean aggregation, where the trajectory-level normalization term $1/|\mathcal{A}_i|$ is equivalently implemented through an inverse-length weighting factor.

This equivalence further reveals the underlying motivation of the proposed aggregation strategy. The inverse-length factor prevents the optimization magnitude of each trajectory from scaling with its action sequence length. Consequently, longer tool-use trajectories no longer dominate policy updates merely because they contain more action tokens, avoiding the undesirable length bias introduced by naive token-level averaging.

By preserving the length-invariant property of sequence-mean aggregation while expressing it through token-level weights, the proposed formulation enables a unified optimization framework that naturally accommodates heterogeneous token types, including reasoning and action tokens.

A.2 Implementation Details

This section provides a detailed overview of the hyperparameter settings and configurations used in our experiments.

Hyperparameters for ALFWorld. All methods are trained under identical experimental configurations. The maximum prompt length and response length are set to 2048 and 512 tokens, respectively, and each episode allows at most 50 environment interaction steps. The actor learning rate is set to 1×10^{-6}, while the critic learning rate is set to 1×10^{-5} for PPO only. We employ a rule-based reward function, assigning a reward of 10 for successful task completion and 0 otherwise. To discourage invalid actions generated by the agent, an additional penalty of -0.1 is applied. Since our method introduces no additional learnable parameters or tunable hyperparameters, except for the auxiliary step-level advantage estimator adopted from (Feng et al., 2026), we directly follow its original configuration without further tuning.

For all group-based RL algorithms, we set the group size to 8 and sample 16 groups for each rollout, resulting in a total of $16 \times 8 = 128$ parallel environments. PPO uses 128 independent environments for rollout collection. The rollout and evaluation temperatures are set to 1.0 and 0.4, respectively. The mini-batch size is set to 256, and the KL-divergence regularization coefficient is set to 0.01.

Hyperparameters for WebShop. We use the same training protocol as ALFWorld with task-specific configurations. The maximum prompt length and response length are set to 4096 and 512 tokens, respectively, and each episode is limited to 15 environment steps. The actor learning rate is set to 1×10^{-6}, while PPO uses a critic learning rate of 1×10^{-5}. The reward function follows the same rule-based design, where successful completion receives a reward of 10 and failure receives 0. Invalid actions are assigned a penalty of -0.1.

For group-based RL methods, we use a group size of 8 and sample 16 groups per rollout, yielding 128 parallel environments in total. PPO also performs rollouts with 128 independent environments. The rollout temperature and validation temperature are set to 1.0 and 0.4, respectively. The mini-batch size is set to 64, and the KL-divergence coefficient is set to 0.01.

Hyperparameters for Search-Augmented QA. For search-augmented QA tasks, the maximum prompt length and response length are set to 4096 and 512 tokens, respectively. The maximum number of interaction turns is limited to 4. The actor learning rate is set to 1×10^{-6}. We adopt a rule-based reward scheme, assigning a reward of 1 for successful answers and 0 for unsuccessful ones. Invalid tool calls are penalized with a reward of -0.01.

The training set size is fixed to 256 samples, and the group size is set to 5. The rollout and validation temperatures are set to 1.0 and 0.0, respectively. The mini-batch size is 512, and the KL-divergence regularization coefficient is set to 0.001.

Computing Details. All experiments are conducted on NVIDIA H100 GPUs. For ALFWorld and WebShop, experiments with Qwen2.5-1.5B are performed on 2 H100 GPUs, while Qwen3-4B experiments use 4 H100 GPUs. Each model is trained for 150 optimization iterations. For search-augmented QA, Qwen2.5-3B and Qwen2.5-7B models are trained on 4 and 8 H100 GPUs, respectively, for 200 optimization iterations. Note that for Qwen3 models, we maintain a thinking budget of 450 tokens to activate their intrinsic reasoning capability, following the configuration of $\mathbf{T}^2$PO (Wang et al., 2026).

Table 4: Training hyperparameters for different environments. All methods share the same configurations unless otherwise specified.

Hyperparameter	ALFWorld	WebShop	Search-Augmented QA
Maximum prompt length	2048	4096	4096
Maximum response length	512	512	512
Maximum environment steps / turns	50	15	4
Actor learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}
Critic learning rate (PPO only)	1×10^{-5}	1×10^{-5}	–
Reward for success	10	10	1
Reward for failure	0	0	0
Invalid action penalty	-0.1	-0.1	-0.01
Group size	8	8	5
Number of groups per rollout	16	16	–
Total rollout environments	128	128	–
Training data size	–	–	256
Rollout temperature	1.0	1.0	1.0
Validation temperature	0.4	0.4	0.0
Mini-batch size	256	64	512
KL coefficient	0.01	0.01	0.001

Table 5: Computational resources used for training.

Task	Model	GPU	Training Iterations
ALFWorld	Qwen2.5-1.5B	2×H100	150
	Qwen3-4B	4×H100	150
WebShop	Qwen2.5-1.5B	2×H100	150
	Qwen3-4B	4×H100	150
Search-Augmented QA	Qwen2.5-3B	4×H100	200
	Qwen2.5-7B	8×H100	200

A.3 Comprehensive ablation studies

Table 6 presents the extended ablation study on WebShop. First, *Disentangled Token Aggregation* (DTA) acts as the primary driver of DisTAR: when used alone (*w/ DTA only*), it reaches a 86.0% task score and 69.3% success rate, matching the top competitor T^{2}PO ($86.4/69.2\%$) and significantly exceeding GiGPO ($83.1/65.0\%$). This confirms that credit assignment disentanglement accounts for the majority of performance gains.

Second, *Globally Normalized Surrogate Loss* (GNSL) serves as an indispensable auxiliary module for DTA. Without DTA, GNSL inherently degenerates into standard uniform loss (marked as '-'). However, when paired with DTA (*w/ DTA + GNSL*), it boosts Task Score from 86.0 to 88.6 and stabilizes variance under heterogeneous token weights.

Finally, *Inverse-step Advantage Penalty* (IAP) provides orthogonal trajectory regularization. Combining all three components yields the top performance ($90.8/76.5\%$), showcasing clear cumulative synergies: DTA disentangles token credit, GNSL calibrates gradient variance, and IAP further prunes redundant actions to boost performance.

Table 6: Comprehensive Ablation study on WebShop. DTA: Disentangled Token Aggregation, IAP: Inverse-step Advantage Penalty, GNSL: Globally Normalized Surrogate Loss.

Method / Variant	**Task Score**	**Success Rate**
GiGPO (baseline)	$83.1_{\pm 1.6}$	$65.0_{\pm 3.2}$
T^2PO (Best Competitor)	$86.4_{\pm 1.3}$	$69.2_{\pm 2.6}$
DisTAR (Single Module)		
w/ DTA only	$86.0_{\pm 3.3}$	$69.3_{\pm 5.4}$
w/ IAP only	$84.3_{\pm 2.0}$	$66.8_{\pm 2.6}$
w/ GNSL only	-	-
DisTAR (Pairwise Combinations)		
w/ DTA + IAP	$86.6_{\pm 4.8}$	$69.8_{\pm 7.2}$
w/ DTA + GNSL	$88.6_{\pm 1.6}$	$73.5_{\pm 2.4}$
w/ IAP + GNSL	$84.8_{\pm 2.2}$	$66.2_{\pm 3.6}$
DisTAR (Full Model)	$\mathbf{90.8}_{\pm 1.8}$	$\mathbf{76.5}_{\pm 3.4}$

A.4 Pseudo Code

To achieve both high training throughput and algorithmic flexibility, DisTAR is designed as a unified, single-pass optimization framework. As illustrated in Algorithm 1, rather than explicitly partitioning the reasoning and action tokens into separate computational graphs, which would incur unsustainable memory overhead and require multiple backward passes for long-horizon agent trajectories, DisTAR embeds the heterogeneous token-mean and sequence-mean constraints into an online token-reweighting formulation. By dynamically computing and applying the token-dependent weight $\omega_{i,t}$ and tracking the global normalization denominator W_{total} within a single nested loop over the sequences, DisTAR achieves the mathematically equivalent optimization of dual-stream aggregation using only a single backward pass. Furthermore, this decoupling of token-type weights $\omega_{i,t}$ and advantage magnitudes gives DisTAR exceptional architectural modularity. It seamlessly accommodates advanced process-supervised or step-level credit assignment strategies, such as the step-specific advantage computation in GiGPO; the trajectory-level scalar $\tilde{A}_i$ can be directly replaced with a fine-grained advantage vector $\tilde{A}_{i,t}$ with no modification to the core aggregation logic, allowing the policy to simultaneously absorb step-level advantage signals while strictly eliminating the trajectory-length bias of action sequences.

Algorithm 1: Traning LLM Agents with DisTAR

Input : Policy π_θ, old policy $\pi_{\theta_{\text{old}}}$, batch of G trajectories $\{\tau_i\}_{i=1}^{G}$, baseline V_ϕ, clipping ϵ
Output: Updated policy parameters θ

Initialize global objective $\mathcal{L}(\theta) \leftarrow 0$;
Initialize total effective weight $W_{\text{total}} \leftarrow 0$;
Calculate total tokens in batch: $C_{\text{total}} \leftarrow \sum_{j=1}^{G} |\mathcal{A}_j|$;
// Single pass over the entire batch and sequences
for *each trajectory* τ_i *in batch (*$i = 1, \ldots, G$*)* **do**
    Identify reasoning tokens $\mathcal{R}_i$, action tokens $\mathcal{A}_i$, and action steps S_i;
    Compute trajectory advantage: $A_i \leftarrow V_\phi(\tau_i)$;
    Apply Inverse-step Penalty: $\tilde{A}_i \leftarrow A_i / S_i$;
    Compute action reweighting constant: $\beta_i \leftarrow C_{\text{total}}/(G \cdot |\mathcal{A}_i|)$;
    for *each token* t *in trajectory* τ_i *(*$t = 1, \ldots, T_i$*)* **do**
        // 1. Dynamically assign weights based on token type
        if $t \in \mathcal{R}_i$ **then**
            $\omega_{i,t} \leftarrow 1$;
            $W_{\text{total}} \leftarrow W_{\text{total}} + 1/S_i$; // Accumulate global normalization denominator
        else
            $\omega_{i,t} \leftarrow \beta_i$;
            $W_{\text{total}} \leftarrow W_{\text{total}} + \beta_i / S_i$;
        end
        // 2. Compute PPO Surrogate Loss on the fly
        $\rho_{i,t}(\theta) \leftarrow \frac{\pi_\theta(y_{i,t}|\cdot)}{\pi_{\theta_{\text{old}}}(y_{i,t}|\cdot)}$;
        $r_{i,t}(\theta) \leftarrow \max\Big(-\rho_{i,t}(\theta)\tilde{A}_i,\ -\text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\tilde{A}_i \Big)$;
        // 3. Accumulate weighted loss
        $\mathcal{L}(\theta) \leftarrow \mathcal{L}(\theta) + \omega_{i,t} \cdot r_{i,t}(\theta)$;
    end
end
// Apply Global Normalization Scale
$\mathcal{L}(\theta) \leftarrow -\frac{1}{W_{\text{total}}} \mathcal{L}(\theta)$;
Perform backward pass on $\mathcal{L}(\theta)$ to update θ;
return *Updated policy* π_θ

A.5 LIMITATIONS

Despite the consistent improvements demonstrated by DisTAR, several limitations remain. First, the current formulation assumes an explicit separation between reasoning and action tokens, which is naturally supported by agents with structured reasoning–action outputs but may not directly apply to models where reasoning is implicit, hidden, or tightly interleaved with actions. Second, although our experiments cover embodied agents, web agents, and search-augmented question answering across multiple model scales, they are primarily conducted on text-based benchmarks and Qwen-family models. Further evaluation on other model families, multimodal agents, and more complex real-world environments is therefore needed to establish broader generality. Finally, the inverse-step advantage penalty introduces an explicit preference for shorter action trajectories. While this is effective in our evaluated settings, where redundant interactions often increase error propagation, some tasks may inherently require longer yet necessary action sequences. Future work could explore adaptive, task-aware efficiency regularization that distinguishes redundant interactions from necessary long-horizon execution.

A.6 CASE STUDY

In this section, we conduct a case study to qualitatively analyze how DisTAR shapes the reasoning and action behaviors of LLM agents. We compare the resulting trajectories with those produced by GiGPO using Qwen2.5-1.5B as the base model.

1. ALFWorld

GiGPO:

Environment (Step 1)

-= Welcome to TextWorld, ALFRED! =-

Your task is to: put a vase in safe. You are in the middle of a room. Looking quickly around you, you see a armchair 1, a coffeetable 1, a desk 1, a diningtable 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a safe 1, a shelf 3, a shelf 2, a shelf 1, a sofa 2, a sofa 1, and a tvstand 1.

Agent (Step 1)

<think>

1. The task is to put a vase into the safe.
2. To complete this task, I first need to locate the vase and then move it to the target container.
3. The vase is likely placed on a surface such as a table or cabinet, so I should inspect possible locations in the room.
4. The available objects include several pieces of furniture, and I should begin exploring nearby locations to find the target vase.
5. I will first check the tvstand 1.

</think>

<action>go to tvstand 1</action>

Environment (Step 2)

You arrive at tvstand 1. On the tvstand 1, you see a cellphone 2, a creditcard 3, a creditcard 2, and a television 1.

Agent (Step 2)

<think>

1. I have arrived at the tvstand 1, but the observed objects do not include the vase.
2. Therefore, the vase is not located at this position.
3. I should continue searching other furniture locations where the vase may have been placed.
4. The sofa areas are potential locations to inspect next.

</think>
<action>go to sofa 1</action>

Environment (Step 3)

You arrive at sofa 1. On the sofa 1, you see a pillow 2, and a pillow 1.

Agent (Step 3)

<think>

1. I have inspected sofa 1, where only pillows are observed.
2. Since the vase is not present, I need to search another location.
3. The remaining unexplored furniture may contain the target object.
4. I will continue checking sofa 2 before moving to other areas.

</think>
<action>go to sofa 2</action>

Environment (Step 4)

You arrive at sofa 2. On the sofa 2, you see a keychain 2, and a laptop 2.

Agent (Step 4)

<think>

1. Sofa 2 also does not contain the vase.
2. Decorative objects such as vases are more likely to be placed on tables rather than sofas.
3. The dining table is therefore a more promising location to inspect.
4. I should move to the dining table to search for the vase.

</think>
<action>go to diningtable</action>

Environment (Step 5)

Nothing happens

Agent (Step 5)

<think>

1. The previous movement did not reveal the contents of the dining table.
2. I need to move to the specific diningtable 1 instance to inspect it.
3. Checking the exact object location will help determine whether the vase is available for pickup.

</think>
<action>go to diningtable 1</action>

Environment (Step 6)

You arrive at diningtable 1. On the diningtable 1, you see a cellphone 1, and a vase 1.

Agent (Step 6)

<think>

1. I have found vase 1 on diningtable 1.
2. Since the goal is to place the vase into the safe, I first need to pick up the vase.
3. The appropriate action is to take vase 1 from its current location.

</think>
<action>take vase 1</action>

Environment (Step 7)

Nothing happens.

Agent (Step 7)

<think>

1. The previous pickup attempt did not succeed.
2. To interact with an object in this environment, I need to specify both the object and its source location.
3. Therefore, I should explicitly take vase 1 from diningtable 1.

</think>
<action>take vase 1 from diningtable 1</action>

Environment (Step 8)

You pick up the vase 1 from the diningtable 1.

Agent (Step 8)

<think>

1. I have successfully obtained the vase.
2. The remaining objective is to place the vase inside the safe.
3. I should now navigate to the safe location before performing the final placement operation.

</think>
<action>go to safe 1</action>

Environment (Step 9)

You arrive at safe 1. The safe 1 is closed.

Agent (Step 9)

<think>

1. I have arrived at safe 1, but the safe is currently closed.
2. The vase cannot be placed inside a closed container.
3. Therefore, I need to open the safe before moving the vase into it.
4. The correct action is to open safe 1.

</think>
<action>open safe 1</action>

Environment (Step 10)

You open the safe 1. The safe 1 is open. In it, you see nothing.

Agent (Step 10)

<think>

1. The safe is now open and ready to receive objects.
2. I already have vase 1 in my possession.
3. The final step is to move the vase into the safe to complete the task.

</think>
<action>move vase 4 to safe 1</action>

Figure 4: An example of agent-environment interaction trajectory in ALFRED for GiGPO. The agent generates reasoning traces enclosed by $\langle think \rangle$ and performs actions enclosed by $\langle action \rangle$.

DisTAR:

Environment (Step 1)

-= Welcome to TextWorld, ALFRED! =-
Your task is to: put a vase in safe.
You are in the middle of a room. Looking quickly around you, you see a armchair 1, a coffeetable 1, a desk 1, a diningtable 1, a drawer 6, a drawer 5, a drawer 4, a drawer 3, a drawer 2, a drawer 1, a garbagecan 1, a safe 1, a shelf 3, a shelf 2, a shelf 1, a sofa 2, a sofa 1, and a tvstand 1.

Agent (Step 1)

<think>

1. The goal is to put a vase into the safe. This task can be decomposed into three sub-goals: locate the vase, obtain the vase, and transfer it into the safe.
2. From the initial observation, the safe 1 is already visible in the room, which means the final destination is known and does not require additional exploration.
3. Therefore, the main challenge is identifying the current location of the vase.
4. In household environments, objects usually follow functional placement patterns. Decorative objects such as vases are more likely to appear on tables, shelves, or other display surfaces rather than sofas, drawers, or electronic furniture.
5. Among the available candidates, the diningtable 1 provides the most reasonable placement prior because tables commonly contain decorative objects.
6. Exploring every visible object sequentially would introduce unnecessary interactions and increase the action trajectory length.
7. Based on the object-location relationship and the current scene layout, I should directly inspect diningtable 1.

</think>

<action>go to diningtable 1</action>

Environment (Step 2)

You arrive at diningtable 1. On the diningtable 1, you see a cellphone 1, and a vase 1.

Agent (Step 2)

<think>

1. I have arrived at diningtable 1 and observed a vase 1 on the table.
2. This observation confirms the previous hypothesis that the diningtable is the correct location of the target object.
3. The cellphone 1 is irrelevant to the current task because only the vase needs to be transferred to the safe.
4. The vase is directly accessible and no additional manipulation is required before picking it up.
5. To minimize future interactions, I should acquire the vase immediately and then proceed toward the destination.

</think>

<action>take vase 1 from diningtable 1</action>

Environment (Step 3)

You pick up the vase 1 from the diningtable 1.

Agent (Step 3)

<think>

1. The vase has been successfully picked up and is now in my possession.
2. The remaining requirement is only to place the vase inside the safe.
3. The initial observation already provided the location of safe 1, so searching other regions is unnecessary.
4. A direct transition to the safe will reduce redundant environmental interactions and shorten the action sequence.
5. Before placing the vase, I should first reach the safe and check whether it can accept the object.

</think>

<action>go to safe 1</action>

Environment (Step 4)

You arrive at safe 1. The safe 1 is closed.

Agent (Step 4)

<think>

1. I have reached safe 1, which is the target container specified by the task.
2. However, the safe is currently closed, meaning that the vase cannot be placed inside at the moment.
3. The current environment state introduces a prerequisite constraint: the container must be opened before performing the placement operation.
4. Opening the safe is therefore the necessary intermediate action.
5. After opening it, I can complete the final transfer operation.

</think>

<action>open safe 1</action>

Environment (Step 5)

You open the safe 1. The safe 1 is open. In it, you see nothing.

Agent (Step 5)

<think>

1. The safe is now open and ready for storing objects.
2. I still possess the vase obtained from the diningtable.
3. All task prerequisites have been satisfied: the target object has been acquired and the destination container is available.
4. The final required operation is transferring the vase into the safe.
5. Executing this action will directly complete the task objective.

</think>

<action>move vase 1 to safe 1</action>

Figure 5: An example of an agent-environment interaction trajectory in ALFWorld for DisTAR. By performing more comprehensive reasoning before acting, the agent accurately identifies relevant objects and completes the task with a shorter and more efficient action chain, reducing unnecessary environment interactions.

2. WebShop

GiGPO:

Environment (Step 1)

Your task is to: Find me long lasting, easy care, machine washable men's pants with elastic waistband with color: green-charcoal, and size: medium, and price lower than 50.00 dollars.

Agent (Step 1)

<think>

1. The user wants men's pants satisfying multiple constraints, including material properties, washing requirements, waistband type, color, size, and price.
2. I should first search for relevant pants products using the main category information.
3. Since the exact product name is unknown, I will start with a broad search query related to men's pants.

</think>

<action>search[men pants]</action>

Environment (Step 2)

Search results show several men's pants products.
Product A: Men's casual pants, black, elastic waistband, $35.
Product B: Men's athletic pants, green-charcoal, $45.
Product C: Men's hiking pants, gray, $42.

Agent (Step 2)

<think>

1. Product B matches the requested color and price constraint.
2. However, I need to verify whether it satisfies all functional requirements such as machine washable and easy care.
3. I should inspect the product details before making a decision.

</think>
<action>click[Product B]</action>

Environment (Step 3)

Product B details:
Color: green-charcoal
Size available: large, medium
Price: $45
Features: Elastic waistband, Machine washable.

Agent (Step 3)

<think>

1. Product B satisfies most constraints including color, size, price, waistband type, and washing requirement.
2. I still need to confirm whether the pants are described as long lasting and easy care.
3. I should check additional product information.

</think>
<action>click[details]</action>

Environment (Step 4)

Additional description:
"Durable fabric designed for long-lasting use. Easy care and machine washable."

Agent (Step 4)

<think>

1. The product satisfies all user requirements.
2. The price is below $50, and the requested size and color are available.
3. Therefore, I can select this product.

</think>
<action>buy[Product B]</action>

Figure 6: An example of an agent-environment interaction trajectory in WebShop for GiGPO. For clarity and ease of comparison, we only present the key summarized information of the interaction process.

DisTAR:

Environment (Step 1)

Your task is to:
Find me long lasting, easy care, machine washable men's pants with elastic waistband with color: green-charcoal, and size: medium, and price lower than 50.00 dollars.

Agent (Step 1)

<think>

1. The task requires finding a pair of men's pants satisfying multiple constraints, including category, functional properties, appearance, size, and price.
2. These requirements can be divided into three groups: semantic attributes (long lasting, easy care, machine washable), structural attributes (men's pants, elastic waistband, green-charcoal color, medium size), and numerical constraints (price below $50).
3. Searching with only a broad category such as "men's pants" would retrieve many irrelevant products and require unnecessary browsing.
4. The most discriminative attributes are the uncommon color "green-charcoal" and the specific design feature "elastic waistband", which can significantly reduce the search space.
5. Therefore, I should construct a targeted search query containing these key constraints before examining products.

</think>

<action>search[green-charcoal men's pants elastic waistband]</action>

Environment (Step 2)

Search results:
Product A: Men's casual pants, black color, elastic waistband. Price: $35.
Product B: Men's performance pants. Color: green-charcoal. Price: $45.
Product C: Men's hiking pants. Color: gray. Price: $42.

Agent (Step 2)

<think>

1. Product B is the only candidate matching the required green-charcoal color.
2. Although Product A satisfies the waistband requirement, its color does not match the user's preference.
3. Product C has an acceptable price but fails the required color constraint.
4. Therefore, Product B is the most promising candidate and I should inspect its detailed attributes.

</think>

<action>click[Product B]</action>

Environment (Step 3)

Product B details:
Name: Men's performance pants
Color: green-charcoal
Available sizes: small, medium, large
Price: $45
Features: - Elastic waistband - Machine washable - Easy care - Durable fabric for long-lasting use

Agent (Step 3)

<think>

1. Product B satisfies all user requirements.
2. The product matches the requested color and provides the required medium size option.
3. The description confirms the semantic requirements including durability, easy care, and machine washability.
4. The price of $45 is within the specified budget constraint.
5. Since all constraints are satisfied, selecting this product completes the task without further exploration.

</think>

<action>buy[Product B]</action>

Figure 7: An example of an agent-environment interaction trajectory in WebShop for DisTAR. Through comprehensive constraint-aware reasoning before acting, the agent identifies relevant products and completes the task with a concise action chain while avoiding unnecessary browsing interactions.