Token-level Advantage Policy Optimization from Negative Feedback in Multi-Turn Agents

Xufeng Zhou
Institute of Automation, Chinese Academy of Sciences
Beijing, China
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
zhouxufeng2019@ia.ac.cn

Linjing Li
Institute of Automation, Chinese Academy of Sciences
Beijing, China
Wenge Technology Co., Ltd.
Beijing, China
linjing.li@ia.ac.cn

Daniel Dajun Zeng
Institute of Automation, Chinese Academy of Sciences
Beijing, China
School of Artificial Intelligence, University of Chinese Academy of Sciences
Beijing, China
dajun.zeng@ia.ac.cn

ABSTRACT

Training multi-turn agents for complex tasks is challenged by sparse rewards. Existing methods are inefficient: they either learn exclusively from successes, discarding valuable failure data, or require rigid win-loss pairs, limiting data utilization. We propose Token-level Advantage Policy Optimization (TAPO), a flexible, pair-free method that leverages all trajectories. TAPO translates a trajectory's terminal reward into token-level advantages, effectively reinforcing the entire sequence of actions in successful trajectories while penalizing those in failed ones. Furthermore, TAPO concentrates updates on high-entropy tokens, which represent pivotal moments of model uncertainty and are thus crucial for efficient exploration and policy improvement. As a post-training optimization, TAPO substantially boosts a baseline SFT agent's average score from 74.2 to 89.4 (+20.5% relative) on three challenging multi-turn benchmarks, outperforming RFT and negative-aware baselines and demonstrating consistent gains in both seen and unseen settings. The code is available at https://github.com/Sunrepe/TAPO.

KEYWORDS

Multi-turn agents, Token-level Policy optimization, Entropy-guided Learning, GAAI

ACM Reference Format:
Xufeng Zhou, Linjing Li, and Daniel Dajun Zeng. 2026. Token-level Advantage Policy Optimization from Negative Feedback in Multi-Turn Agents. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/RDEZ2224

1 INTRODUCTION

Multi-turn agents, ranging from conversational assistants to embodied decision systems, often necessitate long interaction horizons before receiving sparse terminal rewards. This characteristic renders policy learning particularly challenging: algorithms must efficiently exploit scarce successful trajectories while simultaneously extracting informative lessons from abundant failure cases.

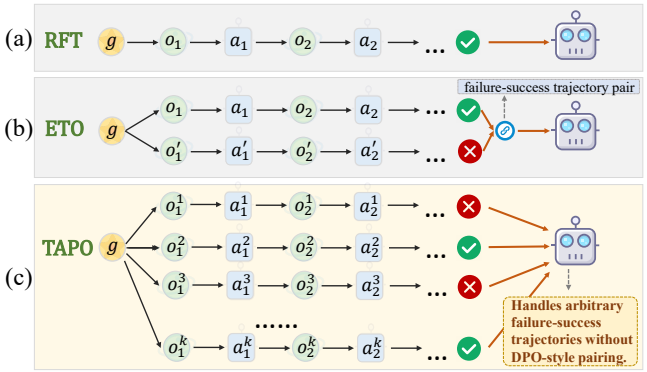
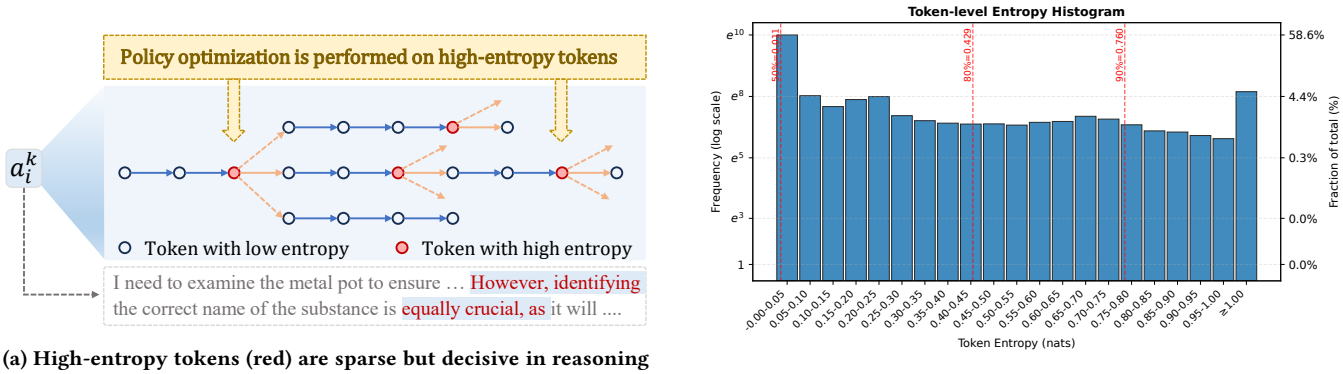


Figure 1: Comparison of RFT, ETO, and TAPO strategies. (a) RFT utilizes only successful data. (b) ETO and DPO-based methods require rigid failure–success trajectory pairs. (c) TAPO leverages arbitrary mixtures of successful and failed trajectories via group-level advantage estimation without explicit pairing.

Existing post-training paradigms exhibit complementary limitations in this context. Rejection Sampling Fine-tuning (RFT) [3] effectively learns from successes but entirely discards the negative signals embedded in failures. Conversely, preference-based methods such as DPO [14], ETO [21], and IPR [27] leverage both positive and negative feedback but typically rely on rigid win-loss pairing (as shown in Figure 1(b)). This constraint leads to data inefficiency, as exploration in open-ended environments naturally yields unpaired, arbitrary numbers of successes and failures. Recent negative-aware methods, such as NSR [33] and PMPO-AR [1], utilize failures but often optimize at the sequence level with coarse credit assignment.

Token-level optimization offers a finer granularity of control but has primarily been explored in restricted settings. Approaches like TPPO [13] and POAD [26] focus on short-text generation or simple interactions. These differ significantly from native long-horizon agent environments (e.g., ScienceWorld [23], ALFWorld [19], WebShop [29]), where trajectories involve extensive Chain-of-Thought (CoT) reasoning [25] and rewards are strictly delayed. In such extended horizons, uniformly distributing a sparse terminal signal to every token introduces substantial noise, as the critical reasoning

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/RDEZ2224



(a) High-entropy tokens (red) are sparse but decisive in reasoning path selection.

(b) Entropy histogram of token-level advantages.

Figure 2: Comparison of entropy-related visualizations.

steps are often buried amidst numerous syntactically necessary but decision-irrelevant tokens.

To address this credit assignment challenge, we propose **Token-level Advantage Policy Optimization (TAPO)**, a paradigm designed for sample-efficient learning from unpaired trajectories. Unlike traditional dense-reward methods, TAPO does not require step-by-step supervision. Instead, it effectively "distills" the outcome of a long trajectory into its decisive moments. It achieves this through two core innovations:

1. Group-level Advantage Estimation To learn from unpaired data without rigid win-loss constraints, we treat a batch of trajectories sampled from the same start state as a group, inspired by GRPO [8]. We calculate a normalized advantage for each trajectory based on its relative performance within the group. Crucially, this trajectory-level advantage is broadcast to the token level, providing a unified learning signal for the entire sequence that reflects its overall success or failure relative to the baseline, thereby maximizing the utility of both positive and negative samples (Figure 1(c)).

2. Entropy-guided Critical Token Selection: While the advantage is derived at the trajectory level, applying it uniformly to every token is inefficient for long-horizon tasks. Drawing on insights from Wang et al. [24], who observe that generation is dominated by low-entropy tokens serving merely for syntactic coherence, TAPO focuses on the sparse subset of high-entropy positions that govern critical reasoning branching (Figure 2a). We employ an entropy-based indicator to identify these decision-critical tokens and strictly mask the policy update to them. This ensures that the trajectory advantage reinforces only pivotal decisions while ignoring trivial syntactic generations.

Experiments on WebShop, ScienceWorld, and ALFWorld demonstrate consistent gains across model scales. Utilizing Qwen3-4B, TAPO achieves an average score of 89.4, surpassing SFT by 15.2 points and outperforming strong baselines including DPO, RFT, NSR, and PMPO-AR by margins of 3.8 to 9.2 points. These results confirm that integrating *GRPO-style advantage estimation* with *entropy-aware target selection* is a highly effective paradigm for sparse-reward multi-turn agent optimization.

Our main contributions are summarized as follows:

- We propose TAPO, a pair-free policy optimization framework that learns jointly from successful and failed trajectories by adapting GRPO-style group advantages to multi-turn agent tasks.
- We introduce an entropy-guided target selection mechanism that focuses optimization on decision-critical tokens, significantly improving sample efficiency.
- We demonstrate that TAPO achieves state-of-the-art performance on complex agent benchmarks, offering a robust alternative to rigid preference-pairing methods.

2 RELATED WORK

2.1 Multi-turn Agents and Sparse Rewards

Recent AgentRL studies fall into two broad lines. The first converts static single-turn tasks into multi-step tool-using reasoning processes (e.g., ARTIST, ToRL, rStar2-Agent, and Satori) [10, 15, 17, 20], effectively reformulating them as trajectory-level decision problems [31]. The second focuses on *native multi-turn environments* such as ScienceWorld, ALFWorld, WebShop, and WebArena, where agents must act under partial observations across many steps with only sparse terminal rewards [19, 23, 29, 32]. Long-horizon planning and credit assignment under delayed feedback are central challenges in this regime [6]. Prior work addresses these via self-reflection, trajectory ranking, or hybrid policy optimization [18, 21, 27]. TAPO is designed for this second, more demanding setting.

2.2 Policy Optimization with Positive and Negative Samples

Preference optimization methods such as DPO and its agent-oriented variants learn from success–failure comparisons but require explicit win-loss trajectory pairing, limiting the use of unpaired data [14]. More recent methods further exploit failure signals: PMPO-AR [1] extends EM theory to incorporate negative samples under a KL-divergence constraint, while NSR [33] demonstrates that assigning a fixed advantage of +1 to positive and −1 to negative trajectories already yields notable gains. At the algorithmic level, GRPO [8] and DAPO [30] integrate positive and negative signals through group-relative normalization. Despite their strengths, these approaches

operate with sequence-level or fixed-sign signals and still depend on some form of trajectory pairing. TAPO instead performs *pair-free* training by computing token-level advantages directly from whole-trajectory returns of both success and failure rollouts, enabling more flexible and efficient utilization of all collected data.

2.3 Entropy-Guided and Token-Level Policy Updates

Token-level methods have been proposed to achieve finer-grained policy updates. TPPO [13] applies per-token PPO to query generation with short sequences (≤10 tokens), relying on a large auxiliary model for token reward annotation—a procedure that is infeasible for long multi-turn trajectories (≥2K tokens). POAD [26] studies token-level optimization via action decomposition in short-horizon settings without chain-of-thought reasoning, which has been shown to degrade substantially in complex environments [21]. ARPO [5] leverages high-entropy token awareness primarily to improve rollout sampling in tool-integrated reasoning, rather than to guide training-time optimization in native interactive environments.

Separately, entropy collapse has been identified as a key obstacle in LLM RL [4], motivating entropy-aware training methods such as AEnt [16]. Most relevant to TAPO, Wang et al. [24] show that trajectory-defining decisions concentrate in a small number of high-entropy "forking" tokens, and that restricting gradient updates to these tokens alone can match or exceed full-token updates. TAPO adopts this insight at training time and uniquely combines entropy-based token selection with token-level positive and negative advantages derived from whole-trajectory returns—achieving targeted, sample-efficient optimization tailored to complex multi-turn interactive environments with sparse rewards.

3 METHOD

3.1 Preliminaries: Formalizing Multi-turn Interaction

We formalize the agent's interaction as a decision-making process. Given a goal g and an initial observation o_1, the agent's policy π_θ generates an action $a_k \sim \pi_\theta(\cdot \mid g, o_1, a_1, \ldots, o_k)$ at step k. The environment returns a new observation o_{k+1}, forming a trajectory $\tau = (g, o_1, a_1, \ldots, o_T, a_T)$ upon termination. Each trajectory is assigned a final reward R, which can be binary ($R \in \{0, 1\}$), continuous, or negative.

To simplify notation, we denote the context at step k as $x_k \triangleq (g, o_{\le k}, a_{<k})$ and the agent's generated action as $y_k = a_k$. We use x and y to generally refer to the context and target action sequences, respectively.

3.2 Efficient Policy Updates via Entropy-Guidance

A central observation is that, for many sequence decision tasks, only a subset of tokens—those with high predictive uncertainty under the policy model—are critical for improving decision quality. TAPO concentrates policy updates on such decision-critical tokens to improve compute-efficiency and agent performance.

Specifically, the entropy for each token in sequence y_i can be computed using the policy model π_θ as:

$$H_{i,t} = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid y_{i,<t}, x_i) \log \pi_\theta(v \mid y_{i,<t}, x_i), \tag{1}$$

where $\mathcal{V}$ denotes the vocabulary. We then define a selection indicator $S_{i,t}$ with an entropy threshold η:

$$S_{i,t} = M_{i,t} \cdot \mathbf{1}\big(H_{i,t} \ge \eta\big). \tag{2}$$

The indicator combines two components: (1) an **entropy filter** $\mathbf{1}(\cdot)$ which is active for high-entropy tokens, and (2) a **mask** $M_{i,t}$ that ensures gradients are applied only to agent-generated action tokens, excluding context such as environment feedback or prompts.

3.3 Token-level Advantage Policy Optimization (TAPO)

3.3.1 Token-level Advantage Computation. The core of TAPO is to adapt the concept of advantage functions from reinforcement learning for policy optimization at the token level. It enables efficient learning from an unpaired mix of success and failure samples by assigning advantages based on final trajectory outcomes.

For a given task with same task objective, we use the reference policy π_{ref} to sample G trajectories $\{\tau_i\}_{i=1}^{G}$. Each trajectory receives a binary reward $R_i \in \{0, 1\}$, where $R_i = 1$ if the trajectory fully succeeds in accomplishing the task and $R_i = 0$ otherwise. To compute the advantage, we standardize the rewards across these G trajectories as follows:

$$\hat{A}_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G}) + \epsilon}, \tag{3}$$

where ϵ is a small constant for numerical stability. The advantage $\hat{A}_i$ is assigned uniformly to all tokens within trajectory i, so that successful trajectories obtain positive values while failed ones obtain negative values.

3.3.2 TAPO Loss Formulation. To effectively propagate trajectory-level signals to token-level optimization, we draw inspiration from GRPO's clipped objective[8] and introduce TAPO loss as

$$\mathcal{L}_{\text{TAPO}}(\theta) = -\mathbb{E}_{\substack{x \sim \mathcal{D} \\ \{y_i\}_{i=1}^{B} \sim \pi_{\text{ref}}}} \left[\frac{1}{\sum_{i=1}^{B} \sum_{t=1}^{|y_i|} S_{i,t}} \sum_{i=1}^{B} \sum_{t=1}^{|y_i|} S_{i,t}\, \ell\big(w_{i,t}(\theta), \hat{A}_i\big) \right], \tag{4}$$

where

$$\ell\big(w_{i,t}, \hat{A}_i\big) = \min\Big(w_{i,t}\,\hat{A}_i,\ \text{clip}\big(w_{i,t}, 1-\varepsilon, 1+\varepsilon\big)\,\hat{A}_i\Big).$$

Here $w_{i,t}(\theta) = \dfrac{\pi_\theta(y_{i,t} \mid y_{i,<t}, x_i)}{\pi_{\text{ref}}(y_{i,t} \mid y_{i,<t}, x_i)}$ is the importance ratio and ε is the clipping hyperparameter. The token selector $S_{i,t}$, as detailed in Section 3.2, determines whether token (i, t) is included in the loss computation.

Notably, our approach differs from GRPO [8] in its optimization scope. While the advantage estimation remains strictly group-level, the policy optimization is performed across the entire batch B. In contrast, GRPO performs optimization within distinct groups of trajectories. This modification is a practical compromise motivated by computational constraints; processing a large number of

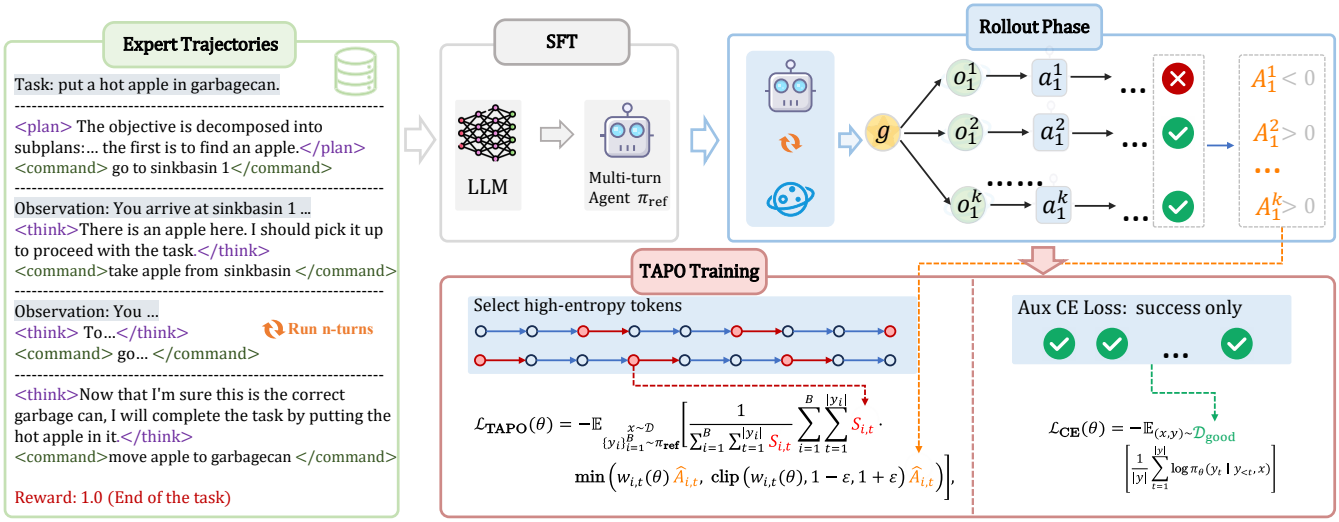


Figure 3: Overview of the TAPO framework. During each interaction, the agent generates both a reasoning process (reason) and the corresponding command. After SFT on expert trajectories, we sample trajectories for each task and compute the corresponding advantages, which are then used to build the TAPO loss for training.

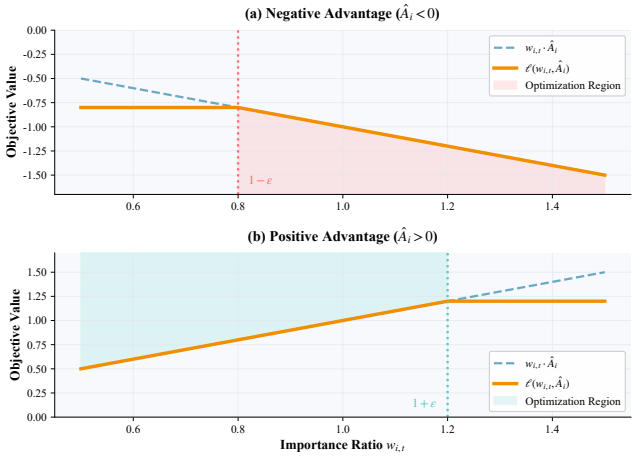


Figure 4: Gradient and behavior analysis.

groups simultaneously within a single mini-batch often leads to out-of-memory (OOM) errors. By operating on a standard batch, our method can scale to any number of trajectories. As our experiments will demonstrate, this removal of the intra-group optimization constraint still yields excellent performance.

3.3.3 Gradient Analysis of TAPO. To elucidate the mechanics of TAPO, we analyze the gradient of its objective function. The clipping mechanism, illustrated in Figure 4, constrains the gradient of the loss term ℓ with respect to the importance ratio $w_{i,t}$:

$$\frac{\partial \ell(w_{i,t}, \hat{A}_i)}{\partial w_{i,t}} = \begin{cases} \hat{A}_i, & \text{if } w_{i,t} \text{ is within the trust region,} \\ 0, & \text{if } w_{i,t} \text{ is outside the trust region (clipped).} \end{cases}$$

This structure enforces conservative policy updates: $w_{i,t}$ moves in the direction indicated by the sign of $\hat{A}_i$ but stops once it leaves the trust interval $[1-\varepsilon, 1+\varepsilon]$, preventing overly large policy shifts. We analyze the two primary cases below.

(a) Negative Advantage ($\hat{A}_i < 0$). For failure trajectories, the objective is to reduce their likelihood under the policy. TAPO penalizes unfavorable actions only as long as the current policy remains close to the reference policy (within the trust region defined by $1-\varepsilon$). Once the policy diverges sufficiently—such that $w_{i,t} < 1-\varepsilon$—the penalty is deactivated, thereby avoiding excessive suppression that could destabilize training. Since $\hat{A}_i < 0$ in this case, the update decreases $\pi_\theta(y_{i,t} \mid y_{i,<t}, x_i)$, but in a controlled manner that respects the trust region.

(b) Positive Advantage ($\hat{A}_i > 0$). For successful trajectories, the goal is to increase their probability under the policy. TAPO reinforces beneficial actions only within the trust region around the reference policy. The clipping mechanism prevents overly aggressive policy updates when the current policy deviates too far (i.e., $w_{i,t} > 1+\varepsilon$), thereby maintaining training stability and guarding against policy collapse. In this regime, the positive advantage leads to an increase in $\pi_\theta(y_{i,t} \mid y_{i,<t}, x_i)$, but in a bounded and stable fashion.

3.4 Overall Training Objective and Procedure

To maintain the model's language capabilities, we add an auxiliary cross-entropy (CE) loss, computed exclusively on successful trajectories ($\mathcal{D}_{\text{good}}$):

$$\mathcal{L}_{\text{CE}}(\theta) = -\mathbb{E}_{(x,y)\sim \mathcal{D}_{\text{good}}} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid y_{<t}, x) \right]. \quad (5)$$

The final training objective is a weighted sum of the TAPO and CE losses:

$$\widehat{\mathcal{L}}_{\text{total}}(\theta) = \lambda \cdot \mathcal{L}_{\text{TAPO}}(\theta) + \alpha \cdot \mathcal{L}_{\text{CE}}(\theta), \tag{6}$$

where λ and α are balancing coefficients.

3.4.1 Optimization Procedure. The training procedure is as follows (see Figure 3 for an overview):

(1) **Initialization and SFT:** Train a reference policy π_{ref} via Supervised Fine-Tuning (SFT) on expert trajectories. Initialize the learnable policy π_θ with the weights of π_{ref}.
(2) **Rollout Phase:** For a given task, use π_{ref} to sample G trajectories $\{\tau_i\}_{i=1}^{G}$ by interacting with the environment.
(3) **Computation:** Compute the standardized advantage $\widehat{A}_i$ for each trajectory. Additionally, calculate token-level entropies $H_{i,t}$ and identify informative tokens for optimization.
(4) **TAPO Training:** Update the policy π_θ by minimizing the total loss $\widehat{\mathcal{L}}_{\text{total}}(\theta)$, which consists of the TAPO loss and an auxiliary cross-entropy loss. Notably, if a group of G trajectories contains no successes (i.e., all $R_i = 0$), it is discarded to prevent learning from purely negative signals.

4 EXPERIMENTS

4.1 Experimental Setup

Benchmarks. We conducted comprehensive experiments on three classic agent environments to evaluate our proposed TAPO algorithm. These environments are: *WebShop* [29], which simulates real-world web navigation and shopping tasks; *ScienceWorld* [23], designed for conducting scientific experiments in a text-based setting; and *ALFWorld* [19], focused on completing multi-step, embodied household tasks in a simulated environment.

Models and Baselines. We consider both one-shot prompt baselines (GPT-4 [2], GPT-3.5-Turbo [12]) and trainable open-source models (Llama3-8B [7], Qwen3 series models [28]). In addition, we report `Llama2-7B + ETO` [21] as an important baseline. Furthermore, we compare TAPO against the following baselines: **SFT** [11] (direct fine-tuning); **DPO** [14] (using win–loss preference pairs); **RFT** [3] (learning exclusively from successes); **NSR** [33] (fixed signed advantages $+1/-1$); and **PMPO-AR** [1] (EM-style updates with KL-constraint).

Table 1: Learning rates for different models in SFT and post-training stages.

Model	SFT Stage	Post-training Stage
Llama3-8B	6.0×10^{-6}	8.0×10^{-7}
Qwen3-4B	1.8×10^{-5}	6.0×10^{-6}
Qwen3-8B	1.2×10^{-5}	4.0×10^{-6}
Qwen3-0.6B	8.0×10^{-5}	1.2×10^{-5}
Qwen3-1.7B	5.0×10^{-5}	8.0×10^{-6}

Implementation Details. Our TAPO algorithm was implemented by modifying the `openrlhf` [9] codebase. All experiments were conducted on a single server equipped with eight NVIDIA A800 GPUs. The training hyperparameters are as follows:

- **Learning Rate:** The learning rates for different models in both the SFT and post-training stages are summarized in Table 1.
- **TAPO Parameters:** The core TAPO coefficient λ was set to 0.8, and the auxiliary loss coefficient α was set to 0.2. The entropy filtering threshold η was set to 0.01, which filters out approximately 50% of the tokens with the lowest entropy.

Evaluation Metrics. The reward mechanisms differ across environments: WebShop provides a floating-point reward between 0 and 1; ALFWorld offers a binary success reward (0 or 1); ScienceWorld provides sparse, step-level rewards ranging from 0 to 100. To unify the evaluation, we linearly scale all scores to a 0-100 range. To ensure the stability of our results, we run 8 independent trials for each task in the test sets and report the average score. The final performance metric is the mean of these average scores across all tasks within a benchmark.

4.2 Analysis of Main Experimental Results

To comprehensively evaluate the performance of our proposed TAPO method, we conducted extensive experiments across three representative interactive decision-making environments. The main results are presented in Table 2, with the learning rates for different models specified in Table 1. Our analysis centers on the following aspects.

Overall Gain. The results clearly demonstrate that our TAPO method achieves significant performance improvements across all benchmarks. Notably, the Qwen3-4B model enhanced with TAPO achieves a new state-of-the-art (SOTA) average score of 89.4. Compared to the strongest baseline, NSR (85.6), TAPO yields an absolute improvement of 3.8 percentage points. Furthermore, when compared to prompt-based methods using powerful closed-source models like GPT-4 (average score of 54.7), our fine-tuning approach on open-source models shows a compelling advantage, highlighting the immense potential of post-training to unlock model capabilities.

Cross-task Generalization. TAPO's performance on "Unseen" tasks is particularly outstanding, directly validating its strong generalization capabilities. For instance, in the ScienceWorld and ALFWorld environments, Qwen3-4B + TAPO scores 84.5 and 94.6 on unseen tasks, respectively, setting the best results in both. This indicates that TAPO does not merely memorize successful patterns from the training tasks but learns a generalizable and transferable set of policies for decision-making and reasoning. This ability is crucial for agents to handle novel scenarios in the real world.

Consistency Across Base Models. As shown in Table 2, TAPO's effectiveness is not limited to a specific model. It delivers consistent and significant performance gains across various base models, including Llama3-8B, Qwen3-4B, and Qwen3-8B. This confirms the robustness of TAPO as a general-purpose post-training algorithm. An interesting observation is that, under our experimental setup, the Qwen3-4B model fine-tuned with TAPO (89.4 average) slightly outperforms its larger counterpart, Qwen3-8B (87.7 average). We provide a more detailed analysis of this cross-model performance in Section 4.4.

Together, these results indicate that TAPO's loss design and selective-token filtering provide reliable gains in both in-distribution and out-of-distribution (Unseen) task performance.

Table 2: Main results across three environments.

Paradigm	Method	WebShop	ScienceWorld		ALFWorld		Average
			Seen	Unseen	Seen	Unseen	
Prompt-based	GPT-4	63.2	64.8	64.4	42.9	38.1	54.7
	GPT-3.5-Turbo	62.4	16.5	13.0	7.9	10.5	22.1
Baselines	Llama2-7B + ETO	67.4	73.8	65.0	68.6	72.4	69.4
	Qwen 3-4B + SFT	68.7	81.7	69.5	78.2	72.8	74.2
	Qwen 3-4B + DPO	74.3	82.4	77.6	85.5	81.3	80.2
	Qwen 3-4B + RFT	**82.1**	84.7	79.2	92.2	83.8	84.4
	Qwen 3-4B + NSR	78.4	87.9	80.7	91.2	89.8	85.6
	Qwen 3-4B + PMPO-AR	72.5	86.5	81.3	90.7	91.2	84.4
Our Method	Llama3-8B + TAPO	77.3	89.1	81.3	87.1	86.7	84.3
	Qwen 3-4B + TAPO	81.9	90.2	**84.5**	**95.6**	**94.6**	**89.4**
	Qwen 3-8B + TAPO	80.1	**92.1**	81.6	93.2	91.7	87.7

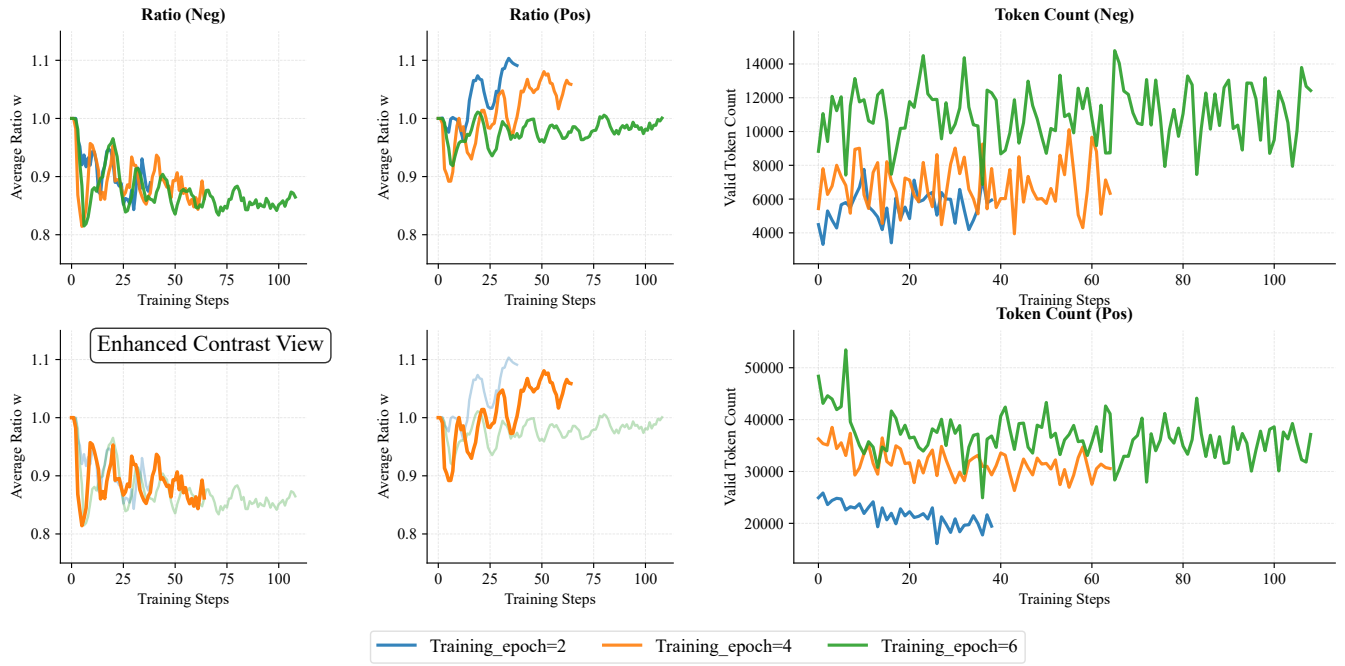


Figure 5: Training process of TAPO on ALFWorld using Qwen3-4B as the base model. The 2×2 subplots on the left depict the dynamics of the ratio during training, where *neg* indicates tokens with Advantage < 0 and *pos* indicates tokens with Advantage > 0. The enhanced contrast view plots highlight the results across training epochs for clearer observation. The two plots on the right show the changes in the number of tokens used for parameter updates throughout training.

Training Dynamics and Stability (Figure 5). To better understand the behavior of TAPO during training, we record and visualize key parameters in Figure 5. The observations align with the following expectations:

First, the relationship between advantage estimates $\hat{A}_i$ and importance ratio $w_{i,t}(\theta)$ behaves as expected in Section 3.3.3: when $\hat{A}_i > 0$, $w_{i,t}(\theta)$ gradually increases; when $\hat{A}_i < 0$, it decreases. This consistent trend across epochs confirms that TAPO dynamically amplifies advantageous trajectories while suppressing detrimental ones.

Second, thanks to the clipping constraint (restricting the policy ratio within $[0.8, 1.1]$), parameter updates remain moderate, preventing policy collapse or instability. As verified in later ablation studies, combining this with the entropy threshold helps the model retain reasoning quality from SFT while enabling stable improvement.

Table 3: Ablation study on training methods and training epochs. The base model is Qwen3-4B.

Training Scheme	Setting	WebShop	Unseen		Average
			ScienceWorld	ALFWorld	
Module Ablation	w/o S_j^t	79.9	82.1	88.9	83.6
	w/o $\mathcal{L}_{\text{CE}}$	79.3	82.8	93.7	85.3
	w/o $\mathcal{L}_{\text{TAPO}}$	**82.1**	79.2	83.8	81.7
	w/ $\widehat{\mathcal{L}}_{\text{total}}$	81.9	**84.5**	**94.6**	**87.0**
Training Iteration	Epoch=1	77.2	81.5	79.2	79.3
	Epoch=2	79.3	88.6	81.3	83.1
	Epoch=3	81.9	**84.5**	91.6	86.0
	Epoch=4	78.7	82.3	93.1	84.7
	Epoch=5	79.5	81.5	**94.6**	85.2
	Epoch=6	75.1	80.1	91.2	82.1

Finally, the figure also illustrates the effect of the entropy threshold over different training epochs. We observe two key trends: (1) as training progresses, the amount of training data sampled per epoch decreases (indicated by shorter training steps in the figure); and (2) concurrently, the number of high-entropy tokens selected by the filter diminishes over epochs. This is expected, as the model becomes more confident in its predictions for these tokens, their entropy falls below the threshold, and they are no longer included in the TAPO loss computation.

4.3 Ablation study

To validate the contribution of each key component within our TAPO framework, we conducted a series of ablation studies. We ablate three main components on the Qwen3-4B model: the entropy-based token selector S_j^t, the auxiliary cross-entropy loss $\mathcal{L}_{\text{CE}}$, and the core TAPO loss term $\mathcal{L}_{\text{TAPO}}$. The results are reported in Table 3.

Primary Role of TAPO Loss. Removing the core $\mathcal{L}_{\text{TAPO}}$ term (row "w/o $\mathcal{L}_{\text{TAPO}}$"), where the model is effectively trained only with SFT, causes the average performance to drop sharply from 87.0 to 81.7. This 5.3-point decrease is the largest among all ablations, strongly confirming that $\mathcal{L}_{\text{TAPO}}$ is the primary driver of the performance gains by enabling the model to learn from the distinction between successful and failed trajectories.

Auxiliary CE Stabilizes Learning. Removing the auxiliary cross-entropy loss $\mathcal{L}_{\text{CE}}$ (row "w/o $\mathcal{L}_{\text{CE}}$") results in a performance drop to 85.3. While still superior to the SFT baseline, this demonstrates that $\mathcal{L}_{\text{CE}}$ plays a crucial role in stabilizing the training process and preventing the model from deviating excessively from the language knowledge acquired during SFT, thereby mitigating catastrophic forgetting.

Entropy Filter Contributes to Robustness. When the entropy filter S_j^t is removed (row "w/o S_j^t"), causing all tokens to be updated, the average performance falls to 83.6. This shows that focusing updates on the most uncertain and critical decision points not only improves training efficiency but also enhances the model's robustness by concentrating the learning signal.

Table 4: Parameters sensitivity of entropy threshold η.

Threshold	Unseen		Average
	ScienceWorld	ALFWorld	
$\eta = 0$	82.1	88.9	85.5
$\eta = 0.01$	84.5	94.6	89.6
$\eta = 0.04$	81.1	86.8	84.0
$\eta = 0.4$	78.9	78.7	78.8

We also evaluate the parameter sensitivity of the entropy threshold η on both **ScienceWorld** and **ALFWorld** benchmarks, as shown in Table 4. We observe that setting $\eta = 0.01$ yields the best overall performance, achieving an average success rate of 89.6. According to Figure 2b, this threshold roughly filters out about 50% of low-entropy tokens, effectively focusing updates on more informative regions. While increasing the threshold further ($\eta > 0.01$) leads to a gradual decline in performance, the auxiliary loss term $\mathcal{L}_{\text{CE}}$ in our objective prevents excessive degradation, maintaining stability across settings. Notably, this choice of threshold differs from that used in [24], suggesting that the optimal entropy constraint is task-dependent and should be tuned according to the specific environment and policy entropy distribution.

Furthermore, we analyze the impact of **Training Iterations**. As shown in Table 3, TAPO demonstrates high efficiency: performance improves substantially after just one epoch, peaks around epoch 3 on WebShop and ScienceWorld and around epoch 5 on ALFWorld, and remains consistently strong in later epochs. These results indicate that TAPO converges quickly and achieves optimal or near-optimal performance within only a few training epochs, highlighting its effectiveness as a post-training method

4.4 Analysis of Cross-Model Performance

To investigate the scalability and generalizability of TAPO across different model sizes, we conducted comparative experiments on

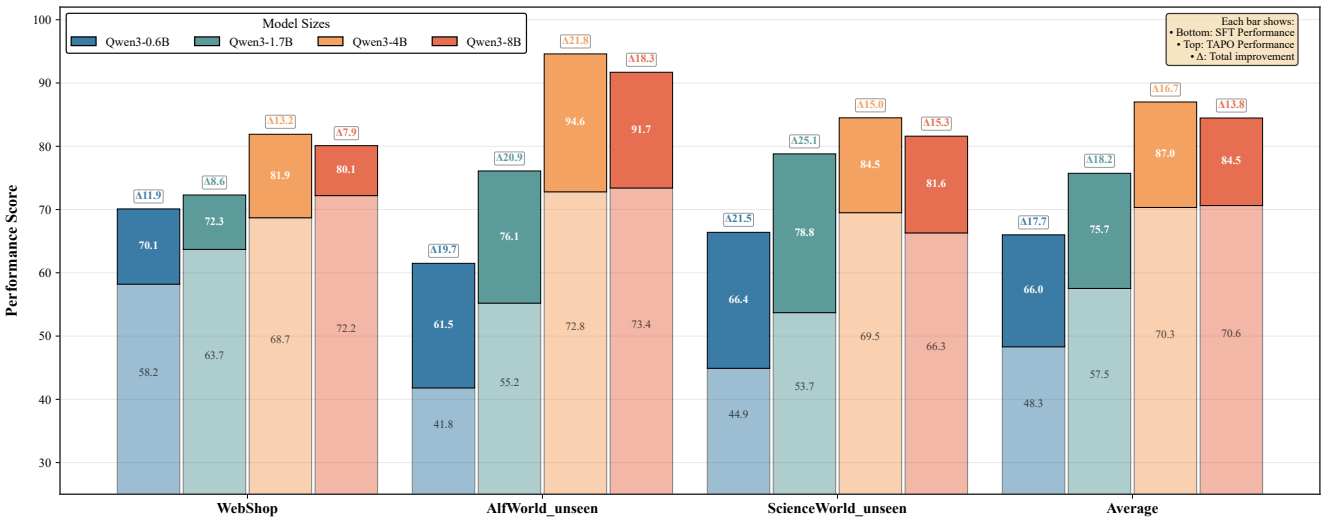


Figure 6: Performance comparison of different models. The figure illustrates the performance of models from the Qwen3 series at different scales across three benchmark environments. Each bar represents two values: the smaller one corresponds to the performance after SFT, while the larger one corresponds to the performance after TAPO post-training.

the Qwen3 model series. Figure 6 illustrates the performance trends of SFT and TAPO with respect to model scale.

Our analysis yields three key findings:

1. **TAPO is universally effective across all model sizes.** From 0.6B to 8B parameters, applying TAPO leads to a substantial leap in performance compared to SFT alone. Specifically, the average performance improvement after applying TAPO exceeds 13 points for all models, with the Qwen3-1.7B model showing the most significant gain of 18.2 points (from 57.5 to 75.7).

2. **TAPO demonstrates outstanding performance across diverse environments.** Despite the significant differences in task formats and domain knowledge between WebShop, ScienceWorld, and ALFWorld, TAPO consistently improves model performance in all three. This highlights its versatility in handling various multi-turn interactive tasks. The improvement is particularly pronounced in the embodied AI environment of ALFWorld, where the average performance gain exceeds 20.0 points.

3. **Performance trends with model scaling reveal an interesting pattern.** For models smaller than 4B parameters, performance for both SFT and TAPO steadily increases with model size. However, a noteworthy phenomenon is that the Qwen3-4B model with TAPO (87.0 average) outperforms the larger Qwen3-8B (84.5 average). We hypothesize two potential reasons for this:

- **Hyperparameter Sensitivity.** Although we set the learning rates (see Table 1) based on the official report [28], these settings may not be optimal for fine-tuning the Qwen3-8B model on these complex interactive environments.
- **Model Capacity versus Task Complexity.** The 4B model might already possess sufficient capacity to capture the necessary reasoning patterns for these environments. As suggested in [22], interactive environments can introduce noisy or conflicting information across steps. A moderately larger model (8B vs. 4B) may not be large enough to effectively filter this noise and could even be more susceptible to it, leading to suboptimal decision-making.

5 CONCLUSION

In this work, we addressed the inefficient use of trajectory data in agent policy optimization by introducing Token-level Advantage Policy Optimization (TAPO). Our method effectively learns from unpaired success and failure trajectories by propagating trajectory-level advantages to individual tokens, a process complemented by an entropy-guided mechanism that focuses updates on the most informative tokens. This approach circumvents the limitations of prior methods that either discard data or require rigid pairing. Our extensive experiments on WebShop, ScienceWorld, and ALFWorld validated TAPO's effectiveness, demonstrating substantial performance improvements across multiple models and achieving a state-of-the-art average score of 89.4. TAPO thereby establishes that a flexible, token-level credit assignment strategy—addressing how to assign advantages and select tokens for training—is a highly effective and sample-efficient path toward building more capable agents.

ACKNOWLEDGMENTS

This work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDA0480301), the Major Project of the National Social Science Fund of China (Grant No. 25&ZD043), and the National Natural Science Foundation of China (Grant No. 62206293).

REFERENCES

[1] Abbas Abdolmaleki, Bilal Piot, Bobak Shahriari, Jost Tobias Springenberg, Tim Hertweck, Michael Bloesch, Rishabh Joshi, Thomas Lampe, Junhyuk Oh, Nicolas Heess, Jonas Buchli, and Martin Riedmiller. 2025. Learning from negative feedback, or positive feedback or both. In *The Thirteenth International Conference on Learning Representations*. https://openreview.net/forum?id=4FVGowGzQb

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

[3] Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large Language Models for Mathematical Reasoning: Progresses and Challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*. 225–237.

[4] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617* (2025).

[5] Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. 2026. Agentic Reinforced Policy Optimization. In *The Fourteenth International Conference on Learning Representations*. https://openreview.net/forum?id=TX4k7BF6aO

[6] Lutfi Eren Erdogan, Hiroki Furuta, Sehoon Kim, Nicholas Lee, Suhong Moon, Gopala Anumanchipalli, Kurt Keutzer, and Amir Gholami. 2025. Plan-and-Act: Improving Planning of Agents for Long-Horizon Tasks. In *Forty-second International Conference on Machine Learning*. https://openreview.net/forum?id=ybA4EcMmUZ

[7] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[8] D. Guo, D. Yang, H. Zhang, et al. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* 645 (2025), 633–638. https://doi.org/10.1038/s41586-025-09422-z

[9] Jian Hu, Xibin Wu, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, et al. 2024. Openrlhf: An easy-to-use, scalable and high-performance rlhf framework. *arXiv preprint arXiv:2405.11143* (2024).

[10] Xuefeng Li, Haoyang Zou, and Pengfei Liu. 2025. ToRL: Scaling Tool-Integrated RL. arXiv:2503.23383 [cs.CL] https://arxiv.org/abs/2503.23383

[11] Ziniu Li, Congliang Chen, Tian Xu, Zeyu Qin, Jiancong Xiao, Zhi-Quan Luo, and Ruoyu Sun. 2025. Preserving Diversity in Supervised Fine-Tuning of Large Language Models. In *ICLR*. https://openreview.net/forum?id=NQEe7B7bSw

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

[13] Yichen Ouyang, Lu Wang, Fangkai Yang, et al. 2025. Token-level Proximal Policy Optimization for Query Generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (Eds.). Association for Computational Linguistics, Suzhou, China, 31196–31210. https://doi.org/10.18653/v1/2025.emnlp-main.1589

[14] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems* 36 (2023), 53728–53741.

[15] Ning Shang, Yifei Liu, Yi Zhu, et al. 2025. rStar2-Agent: Agentic Reasoning Technical Report. arXiv:2508.20722 [cs.CL] https://arxiv.org/abs/2508.20722

[16] Han Shen. 2025. On Entropy Control in LLM-RL Algorithms. *arXiv preprint arXiv:2509.03493* (2025).

[17] Maohao Shen, Guangtao Zeng, Zhenting Qi, et al. 2025. Satori: Reinforcement Learning with Chain-of-Action-Thought Enhances LLM Reasoning via Autoregressive Search. arXiv:2502.02508 [cs.CL] https://arxiv.org/abs/2502.02508

[18] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2023), 8634–8652.

[19] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. {ALFW}orld: Aligning Text and Embodied Environments for Interactive Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=0IOX0YcCdTn

[20] Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. 2025. Agentic Reasoning and Tool Integration for LLMs via Reinforcement Learning. arXiv:2505.01441 [cs.AI] https://arxiv.org/abs/2505.01441

[21] Yifan Song, Da Yin, Xiang Yue, Jie Huang, Sujian Li, and Bill Yuchen Lin. 2024. Trial and Error: Exploration-Based Trajectory Optimization of LLM Agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 7584–7600.

[22] Chupei Wang and Jiaqiu Vince Sun. 2025. Unable to Forget: Proactive Interference Reveals Working Memory Limits in LLMs Beyond Context Length. arXiv:2506.08184 [cs.CL] https://arxiv.org/abs/2506.08184

[23] Ruoyao Wang, Peter A. Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. 2022. ScienceWorld: Is your Agent Smarter than a 5th Grader?. In *EMNLP*. 11279–11298.

[24] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. 2025. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939* (2025).

[25] Jason Wei, Xuezhi Wang, Dale Schuurmans, et al. 2022. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems*, Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (Eds.). https://openreview.net/forum?id=_VjQlMeSB_J

[26] Muning Wen, Ziyu Wan, Jun Wang, Weinan Zhang, and Ying Wen. 2024. Reinforcing LLM Agents via Policy Optimization with Action Decomposition. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. https://openreview.net/forum?id=Hz6cSigMyU

[27] Weimin Xiong, Yifan Song, Xiutian Zhao, Wenhao Wu, Xun Wang, Ke Wang, Cheng Li, Wei Peng, and Sujian Li. 2024. Watch Every Step! LLM Agent Learning via Iterative Step-level Process Refinement. In *EMNLP*.

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

[29] 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 (2022), 20744–20757.

[30] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476* (2025).

[31] Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, et al. 2025. The Landscape of Agentic Reinforcement Learning for LLMs: A Survey. arXiv:2509.02547 [cs.AI] https://arxiv.org/abs/2509.02547

[32] Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *The Twelfth International Conference on Learning Representations*. https://openreview.net/forum?id=oKn9c6ytLx

[33] Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in LLM reasoning. *arXiv preprint arXiv:2506.01347* (2025).