A Practitioner's Guide to Multi-turn Agentic Reinforcement Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We study how to train large language models (LLMs) as autonomous agents that act over multiple turns in agentic environments. While reinforcement learning has driven strong single-turn reasoning, extending to multi-turn environments introduce new challenges yet to be addressed. We formulate multi-turn agentic RL with dense per-turn rewards and token-level credit assignment, and provide a systematic analysis of the impacts three RL pillars – environment, policy, and reward – on multi-turn RL. Under interactive text environments (TextWorld, ALFWorld), we examine scaling with environment complexity and generalization across tasks; we analyze the role of model priors in subsequent multi-turn RL training; we compare the impact of sparse and dense per-turn rewards on RL learning. We provide an extensible code framework for multi-turn agentic RL. Together, our formulation, analysis, and toolkit offer practical guidance for building LLM agents capable of robust multi-turn decision making in agentic environments.

1 Introduction

Training LLMs as autonomous agents to navigate open-ended environments presents unique challenges: planning across extended horizons, making multi-turn sequential decisions, and optimizing for multi-turn rewards. The transition from static single-turn problem-solving to dynamic multistep reasoning is essential for agentic benchmarks such as interactive text and embodied simulations (TextWorld [Côté et al., 2018], ALFWorld [Shridhar et al., 2021], etc.), real-world software programming (OSWorld [Xie et al., 2024], SWE-gym [Pan et al., 2025], etc.), and abstract reasoning in novel situations (ARC-AGI [Chollet et al., 2025]). However, existing multi-turn RL implementations vary widely: some refer to tool-augmented single queries as multi-turn [Zeng et al., 2025], while many rely on model-based assumptions [Wang et al., 2025]. This fragmentation has led to incomparable results across papers and confusion about what constitutes true multi-turn learning versus pseudo-multi-turn adaptations of single-turn methods. Motivated by the lack of standardization of multi-turn RL approaches, our work focuses on providing a unified formulation of multi-turn RL and documenting the critical design decisions that determine success or failure in interactive environments.

This paper aims to facilitate research efforts on the open research question: *What factors are practically important in making multi-turn RL for LLM agent learning work*. First, we formulate multi-turn agentic RL with dense reward structure and token-level credit assignment. Next, we provide a systematic analysis revealing that success in multi-turn agentic RL requires careful co-design across all three pillars illustrated in Figure 1. For the **environment**, we investigate scaling with environment complexity and generalization across different environments. For the **policy**, we investigate how model prior affects continual multi-turn RL training and analyze the interplay between multi-turn imitation learning and multi-turn RL. We further compare both biased (PPO) and unbiased (RLOO) policy gradient RL algorithms to isolate benefits from from algorithmic heuristics.

Submitted to MTI-LLM @ 39th Conference on Neural Information Processing Systems (2025). Do not distribute.

For the **reward**, we experiment with varying densities of per-turn rewards to understand their impact on learning dynamics.

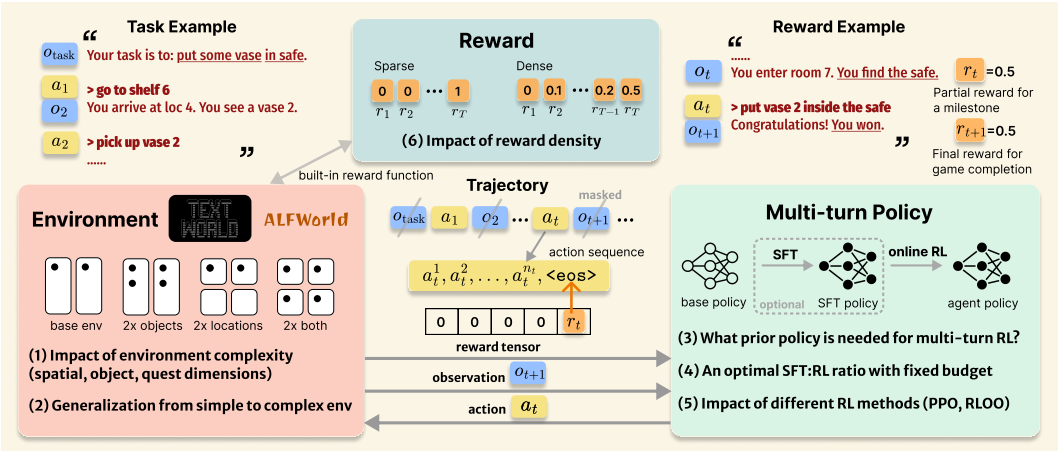

Figure 1: Illustration of multi-turn agentic RL and the key research questions.

Our key findings demonstrate that: 1) Multi-turn RL performance scales with environment complexity in terms of world size, interactable objects, and exploration steps; 2) Agents trained on simpler environments showing promising generalization to complex ones; 3) Model priors from even minimal demonstration data accelerate convergence, but multi-turn RL training is needed for generalization; 4) With a fixed budget, there is an optimal SFT:RL ratio that balances task accuracy and generalization; 5) Both PPO and RLOO achieve stable learning in multi-turn RL, validating that the performance gains stem from our multi-turn formulation rather than the heuristics benefits; 6) Dense turn-level rewards accelerate multi-turn RL training compared to sparse alternatives, but are sensitive to algorithm choice.

We establish that multi-turn RL with LLMs is not merely an extension of single-turn optimization but requires fundamental redesign of environment, policy, and reward. We will release the multi-turn agentic RL framework built upon veRL [Sheng et al., 2025] with all agentic environments included in the paper: TextWorld [Côté et al., 2018], ALFWorld [Shridhar et al., 2021], etc. The framework provides a minimal interface that requires only a `step` function, allowing easy integration of new environments and agents. This paper provides both empirical analysis and practical guidelines for developing the next generation of agentic AI systems that can operate effectively in real-world interactive environments.

2 Related Work

While single-turn RL methods for LLMs including PPO [Schulman et al., 2017], RLOO [Ahmadian et al., 2024], GRPO [Shao et al., 2024], and DAPO [Yu et al., 2025] have been extensively optimized for immediate response quality, adapting them to multi-turn agentic scenarios remains non-trivial. These methods assume rewards directly follow individual actions, but multi-turn environments only reveal outcomes after extended interaction sequences, breaking the action-reward coupling that single-turn methods rely upon. Existing efforts on multi-turn RL have made limited progress on these challenges. Some approaches construct multi-turn scenarios by interleaving tool-use or reasoning steps for single-turn QA pairs [Zeng et al., 2025, Dong et al., 2025]. Others working on true interactive environments either rely on sparse terminal rewards without turn-level learning signals [Wang et al., 2025], or assign turn-level advantages uniformly across sequence tokens without fine-grained credit assignment [Zhou et al., 2025]. More importantly, there lacks a comprehensive understanding of how the three fundamental pillars of RL – environment, policy, and reward – jointly determine performance in multi-turn interactive environments. This paper provides a systematic analysis on how the fundamental pillars of RL impact multi-turn RL training respectively and concludes insights on how to practically train multi-turn RL in different interactive agentic environments. Throughout the paper, we dedicate essential related works in individual sections.

3 Multi-turn Agentic Reinforcement Learning

We formulate multi-turn agentic tasks as a Partially Observable Markov Decision Process (POMDP) problem, defined as a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O}, \gamma)$. Taking the Textworld task [Côté et al., 2018] as an example, an agent takes the action a_t (`go south`) sampled from the action space $\mathcal{A}$ and receives a text observation o_t (`You are in front of a garden`) from the observation space Ω. o_t is a partial description of the true state s_t in the hidden state space $\mathcal{S}$ which contains the complete state world model. We assume that the state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ is deterministic. Upon taking an action, the agent also receives a scalar reward $r_t = \mathcal{R}(s_t, a_t)$. The agent's objective is to learn a policy that maximizes the expected discounted sum of rewards $\mathbb{E}[\sum_t \gamma^t \cdot r_t]$.

We denote the trajectory history consisting of a task prompt u, action and state sequences by $h_t = (u, s_0, a_0, s_1, a_1, \cdots, s_t)$[1]. An LLM agent with policy π_θ samples an action sequence $a_t \sim \pi_\theta(\cdot|h_t)$ based on the trajectory history. a_t is a token sequence in natural language: $(a_t^1, a_t^2, ..., a_t^{n_t}, a_t^{eos})$, with each token a_t^i generated as $\pi_\theta(\cdot|h_t, a_t^{<i})$. Agentic environments execute language commands only upon completion, naturally defining the reward structure at the command boundaries, marked by `<eos>` tokens. Therefore, we assign scalar reward r_t at a_t^{eos}, and the reward for each action token is formulated as: $r_t^i = \begin{cases} r_t & \text{if } a_t^i = \texttt{<eos>} \\ 0 & \text{otherwise} \end{cases}$. We make sure only action tokens contribute to the loss by masking out all state tokens.

Here is a concrete example: the input to the LLM during the rollout stage using a chat template is:

```
<|im_start|>user
Welcome! Your task is: {task prompt}. state: {state 0} your action:<|im_end|>
<|im_start|>assistant
{action 0}<|im_end|>
...
<|im_start|>user
state: {state t} your action:<|im_end|>
<|im_start|>assistant
```

The LLM of policy π_θ generates the output `{action t}<|im_end|>`. The environment handles state transition and reward computation: `next_state, reward, done = env.step(state, action)`. The reward for each turn is assigned to the `<|im_end|>` token. The action and next state are then appended to the chat history under the template.

4 Background and Experimental Setup

Our experiments systematically investigate how the three fundamental pillars of RL impact in multi-turn agentic RL. For **Environment** (Section 5), we examine how scaling environmental complexity along spatial and object dimensions impacts learning (Section 5.1), and test whether agents trained on simple tasks can generalize to more complex environments (Section 5.2). For **Policy** (Section 6), we analyze how model priors from demonstration data influence RL convergence and identify optimal ratios of imitation learning to RL data under budget constraints (Section 6.1). We contrast biased (PPO) with unbiased (RLOO) algorithms to isolate benefits from algorithmic design versus multi-turn formulation (Section 6.2). For **Reward** (Section 7), we investigate how reward density – the frequency of feedback signals during trajectories – affects learning dynamics and final performance (Section 7.1). These results establish that multi-turn RL requires careful co-design across all three components rather than simple extensions of single-turn methods.

Tasks and Environments. We evaluate our multi-turn RL framework on two text-based interactive benchmarks that require sequential decision-making over extended horizons: TextWorld [Côté et al., 2018] and ALFWorld [Shridhar et al., 2021]. For online RL training, we integrate the TextWorld and ALFWorld backends directly into our codebase as environments that interact with LLM agents during rollout via standard `step` and `reset` functions. Unlike traditional RL settings where agents receive both observations and lists of admissible actions at each step, which effectively reduces the task to action selection. Instead, we adopt a more challenging setup where our agents must generate executable natural language commands based solely on environmental observations, without action hints. Here, we specify the tasks we use from the two benchmarks:

[1]We substitute observation o for state s for simplicity. The agent has no access to the true state of the game.

- **TextWorld**: A text-based game environment where agents navigate rooms, manipulate objects, and solve quests through natural language commands. We procedurally generate tasks with controlled complexity along three dimensions: world size (w), number of objects (o), and quest length (q). For example, "w2-o3-q4" denotes a task with 2 rooms, 3 objects, and a 4-step quest. Each task is generated with a unique seed to ensure diversity.
- **ALFWorld**: embodied household environment built on the TextWorld engine, requiring agents to complete multi-step tasks through text-based interaction. We use the text-only variant with tasks spanning six categories[2]. We train on the "train" split and evaluate on both "valid_seen" and "valid_unseen" splits to assess generalization.

Models and Training. We experiment with Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct[3] as base models (abbreviated as Qwen-1.5B and Qwen-7B), training with PPO [Schulman et al., 2017] and RLOO [Ahmadian et al., 2024] algorithms implemented in the veRL framework. For PPO, by default, we use an actor learning rate of 5e-7, a critic learning rate of 5e-6, a clip ratio of 0.2, a discount factor γ of 1.0, a KL penalty coefficient of 0.001, and a zero entropy regularization. We set both rollout and PPO mini-batch sizes to 256. For RLOO, by default, we use an actor learning rate of 1e-6, a KL penalty coefficient of 0.001, and the same batch sizes. Maximum iteration steps and token length limits are adjusted based on task complexity. During rollout generation, we use a temperature of 0.7 to balance exploration and exploitation.

Evaluation. We evaluate agents on held-out test sets, reporting task success rate as the primary metric – the percentage of episodes where agents complete objectives within the allocated exploration budget. All experiments run for 150 epochs unless convergence criteria trigger early stopping or otherwise specified.

5 Environment

The environment fundamentally determines the challenges that an agent must overcome. Unlike single-turn tasks where complexity is primarily measured by reasoning difficulty, multi-turn environments introduce dimensions such as spatial navigation, object manipulation, and extended planning horizons. We focus on two core research questions that directly investigate practical multi-turn deployment. First, we ask: *how environment complexity affects the efficiency of multi-turn RL training*, which helps determine exploration budget and model size requirements for tasks with varied complexities. Second, we want to understand: *whether agents learn generalizable abilities or simply memorize task-specific behaviors*. This addresses whether expensive multi-turn training can transfer across environments, a key consideration for scalable agentic systems.

Setup. We procedurally generate TextWorld environments with varied complexity. Starting from our base configuration of task w2-o3-q4 (2 rooms, 3 objects, 4-step quest), we create controlled variations: w8-o3-q4 isolates spatial complexity by increasing the world size, w2-o12-q4 isolates object complexity by increasing the number of interactable objects, and w8-o12-q4 combines both. Additionally, we create task w4-o6-q8 which linearly scales all dimensions. We train both Qwen-1.5B and Qwen-7B models from scratch using PPO with consistent hyperparameters across all experiments. Training uses 5,000 episodes while evaluation is performed on 100 held-out test episodes with different seeds.

Tasks w/ varying env complexity	Qwen-1.5B	Qwen-1.5B (PPO)
w2-o3-q4 (base env)	0.17	$0.88_{\uparrow 0.71}$
w8-o3-q4 (4x rooms)	0.07	$0.68_{\uparrow 0.61}$
w2-o12-q4 (4x objects)	0.08	$0.54_{\uparrow 0.46}$
w8-o12-q4 (4x objects & rooms)	0.03	$0.51_{\uparrow 0.48}$

Table 1: Multi-turn PPO performance on TextWorld tasks with varying environment complexities. The maximum steps per game is 16 for all tasks.

[2]Pick & Place, Examine in Light, Clean & Place, Heat & Place, Cool & Place, and Pick Two & Place.

[3]`https://huggingface.co/spaces/Qwen/Qwen2.5`

5.1 How does multi-turn RL performance scale with environment complexity?

To understand how individual environment factors impact multi-turn RL, we systematically vary spatial and object complexity while holding other variables constant. As shown in Table 1, base models struggle dramatically as environment complexity increases, with performance dropping from 17% to just 3% when both spatial and object dimensions are scaled. More importantly, **multi-turn RL gains less improvements as the environment complexity increases** – while PPO achieves a 88% improvement on the base environment, this drops to only 51% on the most complex setting. In particular, **object complexity proves to be more challenging than spatial complexity**. This suggests that learning to manipulate and track multiple objects in turns presents fundamentally harder challenges than spatial exploration.

Beyond varying individual complexity dimensions, we examine how proportionally scaling all environment parameters affects multi-turn RL training. As shown in Table 2, **doubling all dimensions creates a dramatically harder problem that goes beyond linear scaling**. The base Qwen-1.5B model's performance collapses from 15% to 1%, indicating that the search space expands significantly. While multi-turn PPO on 1.5B model achieves substantial performance (58% gain), the final 59% success falls well short of the 80% achieved on w2-o3-q4. In addition, the performance also scales with model size – the 7B model reaches 72% success on w4-o6-q8, suggesting that larger models better handle the increased state space of complex environments. We can also see the potential of the 1.5B model in navigating difficult environments considering the huge performance gain (65% on w2-o3-q4 and 58% on w4-o6-q8).

Tasks	Qwen-1.5B	Qwen-1.5B (PPO)	Qwen-7B	Qwen-7B (PPO)
w2-o3-q4	0.15	$0.8_{\uparrow 0.65}$	0.65	$0.98_{\uparrow 0.33}$
w4-o6-q8	0.01	$0.59_{\uparrow 0.58}$	0.28	$0.72_{\uparrow 0.44}$

Table 2: Multi-turn PPO performance on TextWorld tasks with linearly scaled difficulty. The maximum steps per game is 12 for w2-o3-q4 task and 24 for w4-o6-q8 task.

#Exploration steps	Qwen-1.5B	Qwen-1.5B (PPO)
6 (1.5× optimal)	0.05	$0.55_{\uparrow 0.5}$
8 (2× optimal)	0.09	$0.73_{\uparrow 0.64}$
12 (3× optimal)	0.15	$0.8_{\uparrow 0.65}$
16 (4× optimal)	0.17	$0.88_{\uparrow 0.71}$

Table 3: Multi-turn PPO performance on TextWorld w2-o3-q4 task with different exploration sizes.

Tasks	w2-o12-q4	w8-o3-q4	w8-o12-q4	w4-o6-q8
w2-o3-q4	$0.4_{\uparrow 0.32}$	$0.51_{\uparrow 0.44}$	$0.27_{\uparrow 0.24}$	$0.12_{\uparrow 0.11}$
w8-o3-q4	$0.5_{\uparrow 0.42}$	$0.68_{\uparrow 0.61}$	$0.51_{\uparrow 0.48}$	$0.21_{\uparrow 0.2}$
w2-o12-q4	$0.54_{\uparrow 0.46}$	$0.27_{\uparrow 0.2}$	$0.27_{\uparrow 0.24}$	$0.13_{\uparrow 0.12}$
w2-o12-q4 + w8-o3-q4	$0.41_{\uparrow 0.33}$	$0.52_{\uparrow 0.45}$	$0.34_{\uparrow 0.31}$	$0.17_{\uparrow 0.16}$

Table 4: Multi-turn PPO performance on cross-environment generalization across TextWorld tasks. All models are trained with 5000 episodes per epoch. For the mixed-task setting (w2-o12-q4 + w8-o3-q4), the model uses a 50/50 mixture per epoch (2500 episodes from each). The total RL data budget is held constant across all training conditions for a fair comparison.

Lastly, we investigate how the exploration size (the maximum number of steps agents can take during rollout) affects learning and final performance. For the w2-o3-q4 task with an optimal solution length of 4 steps, we vary the exploration size from 6 to 16 steps. Table 3 shows that **performance gains saturate beyond 8 exploration steps**. Constraining agents to 6 steps (1.5× optimal) limits PPO performance to 55% success rate. Increasing to 8 steps (2× optimal) yields 73% success, while further increasing to 12 and 16 steps produces only marginal gains. These results indicate that while insufficient exploration steps severely limit learning, excessive steps beyond 2× optimal provide negligible benefits for TextWorld tasks.

5.2 How does multi-turn RL generalize to environments with different complexities?

To investigate whether multi-turn RL learns transferable skills, we evaluate cross-environment generalization. In addition to single-task models trained on w2-o3-q4, w8-o3-q4, and w2-o12-q4, we train a mixed-task model on a 50/50 combination of w2-o12-q4 and w8-o3-q4 under the same amount of RL data. Table 4 reveals that **agents trained on simpler environments show substantial generalization to more complex environments**, evidenced by the model trained in w2-o3-q4 that improves performance in every higher complexity environment. The transfer of agent ability is especially strong from w8-o3-q4 (higher spatial complexity), which achieves the largest average improvements across targets; notably, it improves w8-o12-q4 by 48% – matching the 48% gain from training directly on w8-o12-q4. These results suggest that multi-turn RL acquires reusable skills, such as spatial exploration and object manipulation, that transfer across environment complexity.

6 Policy

The choice of RL optimization algorithm and model initialization critically determines multi-turn RL performance. First, for practical considerations of multi-turn RL which is understudied, we may or may not have access to human demonstration data for supervised fine-tuning (SFT) under the exact same task domain. A natural question to ask is: *what prior model policy is needed in order to gain sufficient performance from multi-turn RL?* This addresses the practical question of whether expensive human demonstrations are necessary or if agents can learn effectively from scratch. Furthermore, suppose we have a fixed budget for data collection, we try to answer *whether there is an optimal SFT versus RL data training ratio that gives us the highest performance gain*. Second, we are interested in understanding *whether RL optimization choices significantly impact multi-turn RL training*. We pick both a heuristic policy gradient method (PPO [Schulman et al., 2017]) and an unbiased method (RLOO [Ahmadian et al., 2024]) as the algorithms. This isolates the contributions of our multi-turn formulation from specific optimization heuristics, which, as evidenced in [Oertell et al., 2025], are essential when making claims about algorithmic improvements.

Setup. TextWorld provides a gold solution to each procedurally generated game. We use these gold solutions as our human demonstration data for supervised fine-tuning, representing the optimal multi-turn trajectories for each environment. The SFT data follows a turn-based chat format, the default template used in most instruction-following scenarios. To reduce overfitting, we train for exactly one epoch on all SFT data, ensuring each demonstration is seen only once during training. All SFT data is generated with different random seeds than the RL training data to prevent data leakage.

We reuse the TextWorld tasks w2-o3-q4 and w4-o6-q8 from Section 5, and ALFWorld (text-based version), where models are trained on 3553 training episodes and evaluated on 134 "valid_unseen" episodes. We train both Qwen-1.5B and Qwen-7B models using PPO and RLOO respectively with consistent hyperparameters across all experiments. We performed hyperparameter sweeping on both algorithms and present results under optimal configurations which have been listed in Section 4.

Multi-turn PPO Formulation. For optimization algorithms with advantage estimation, such as Proximal Policy Optimization (PPO) [Schulman et al., 2017], we adopt token-level credit assignment. We compute token-level values and apply to TD error as $\delta_t^i = r_t^i + \gamma V(h_t^{i+1}) - V(h_t^i)$ where h_t^i is the history up to and including token a_t^i. Then we estimate the advantage for each token using GAE: $\hat{A}_t^i = \sum_{l=0}^{L-i} (\gamma\lambda)^l \delta_t^{i+l}$, where L is the horizon (number of tokens until episode ends). Even though only `<eos>` tokens receive rewards, all preceding tokens get non-zero advantages through value bootstrapping. Therefore, the Clipped Surrogate Objective for all tokens in the trajectory can be written as:

$$L^{CLIP}(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{T}\sum_{i=1}^{n_t+1} \min\left(r_t^i(\theta)\hat{A}_t^i, \text{clip}(r_t^i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^i\right)\right]$$

where the probability ratio for each token is: $r_t^i(\theta) = \dfrac{\pi_\theta(a_t^i|h_t, a_t^{<i})}{\pi_{\theta_{old}}(a_t^i|h_t, a_t^{<i})}$.

6.1 How does prior model policy influence multi-turn RL training?

We distinguish between two training phases: model priors refer to the initial policy obtained through SFT, while continual training refers to the subsequent multi-turn online PPO training. This two-stage approach mirrors the real-world deployment, where agents first learn from human demonstrations before being deployed for online learning. To investigate how imitation learning affect multi-turn RL performance, we train SFT priors on gold solutions from the w2-o3-q4 TextWorld task, then continue training with multi-turn RL.

Multi-turn RL with good imitation learning priors achieves comparable performance with dramatically fewer RL episodes. As shown in Table 5, an SFT prior trained on 60 demonstrations followed by 400 RL episodes achieves 85% success on w2-o3-q4, nearly matching the 88% performance of pure RL training with 5000 episodes. This represents a significant reduction in RL training data while maintaining competitive performance.

To address practical deployment scenarios, we further investigate the optimal allocation of data resources between SFT and RL. Assuming that SFT data costs 10 times more than RL episodes (reflecting the higher human effort required), we analyze performance under a fixed budget of 1000 cost units across different SFT/RL ratios. Table 5 reveals that pure SFT (100 demonstrations, 0 RL episodes) achieves excellent performance (95%) on the training domain w2-o3-q4 but shows limited generalization to the more complex w4-o6-q8 environment (55%). **The optimal configuration ratio uses 60 demonstrations with 400 RL episodes**, achieving 85% success on w2-o3-q4 and 59% on w4-o6-q8, **which balances task-specific performance with generalization robustness.** The key insight here is that SFT data provides crucial behavioral priors, but RL training is essential for robustness. And in real-world scenarios, demonstration data are usually noisy, which would require more RL data resources.

#SFT data	**#RL data**	SFT	SFT (test on w4-o6-q8)	SFT+PPO	SFT+PPO (test on w4-o6-q8)
/	5000	0.17 (base)	0.01 (base)	0.88*	0.12*
0	1000	/	/	0.54	0.11
20	800	0.59	0.15	0.62	0.15
40	600	0.75	0.51	0.72	0.44
60	400	0.71	0.53	0.85	0.59
80	200	0.94	0.29	0.95	0.35
100	0	0.95	0.55	/	/

Table 5: Multi-turn learning performance across different SFT/RL data allocations under a fixed budget. Models are trained on w2-o3-q4 and evaluated on both w2-o3-q4 and w4-o6-q8 for generalization purpose. We first train model priors through SFT, then continues training through multi-turn PPO. Results marked with * are extracted from previous experiments for comparison.

Lastly, to investigate whether multi-turn RL can benefit from demonstration data collected in different but related domains, we experiment with cross-domain model priors. We train SFT models on 3553 ALFWorld demonstrations and then apply multi-turn PPO on TextWorld w2-o3-q4, and vice versa, training on 3000 TextWorld demonstrations before applying PPO to ALFWorld tasks. We find that **cross-domain priors lead to rapid policy collapse during multi-turn RL training.** A possible reason is that demonstration data creates behavioral biases that conflict with the target environment's action-outcome relationships, making the policy unstable during multi-turn RL.

6.2 How do RL algorithms impact multi-turn RL training?

Understanding whether performance gains stem from our multi-turn formulation or specific algorithmic choices is crucial for establishing the generalizability of our approach. We compare PPO (a heuristic policy gradient method with value function bootstrapping) against RLOO (an unbiased policy gradient estimator) to isolate the contributions of our token-level credit assignment from algorithmic design decisions [Oertell et al., 2025].

Both PPO and RLOO achieve substantial improvements over base models, demonstrating that performance gains stem from our multi-turn formulation rather than PPO-specific heuristics. As shown in Table 6, PPO achieves 88% success on w2-o3-q4 compared to RLOO's 51% success.

This gap becomes bigger on w4-o6-q8, where PPO reaches 59% success while RLOO fails completely with 0% success for the 1.5B model. We conclude that **PPO outperforms RLOO in multi-turn settings, with performance gaps increasing for complex environments.** Model size also affects the performance gap: with 7B parameters, both algorithms perform similarly on simple tasks (97% vs 98%), but PPO maintains advantages on complex tasks (72% vs 47%). These results demonstrate that the performance gains are not due to heuristics from PPO, evidenced by RLOO's consistent improvements across tasks.

Task / Model	Base model	RLOO	PPO
w2-o3-q4 / Qwen-1.5B	0.15	$0.51_{\uparrow 0.36}$	$0.88_{\uparrow 0.73}$
w4-o6-q8 / Qwen-1.5B	0.01	0.0	$0.59_{\uparrow 0.58}$
w2-o3-q4 / Qwen-7B	0.65	$0.97_{\uparrow 0.32}$	$0.98_{\uparrow 0.33}$
w4-o6-q8 / Qwen-7B	0.28	$0.47_{\uparrow 0.21}$	$0.72_{\uparrow 0.44}$

Table 6: Comparison of PPO and RLOO on multi-turn TextWorld tasks across model sizes.

7 Reward

Multi-turn environments typically provide sparse feedback upon task completion. This sparsity creates challenges for credit assignment across extended sequences, potentially leading to slow convergence or training instability. However, some environments provide built-in dense reward signals where agents receive partial rewards at each milestone reached on the solution trajectory. We investigate how reward density, the frequency of feedback signals during trajectories, affects multi-turn RL performance and whether dense rewards can improve learning efficiency.

Setup. We experiment with different reward density schemes on TextWorld tasks using both PPO and RLOO algorithms. Our reward density configurations leverage TextWorld's built-in reward functions with varying densities: (1) sparse rewards, provided only upon successful task completion, and (2) dense rewards, provided at intermediate steps throughout the trajectory. We quantify reward density as the average number of steps between reward signals and larger values indicate sparser rewards. We denote the reward density as the average number of steps per given reward. The larger the value, the sparse the reward. We evaluate the tw-simple tasks[4] from TextWorld, training on 3,000 episodes generated with different random seeds to ensure diversity.

Reward density	Qwen-7B (PPO)	Qwen-7B (RLOO)
Sparse (10.22)	$0.41_{\uparrow 0.12}$	$0.35_{\uparrow 0.06}$
Dense 1 (2.89)	$0.29_{\uparrow 0.0}$	$0.55_{\uparrow 0.26}$
Dense 2 (1.17)	$0.58_{\uparrow 0.29}$	$0.55_{\uparrow 0.26}$

Table 7: Performance comparison across reward density schemes on tw-simple task using Qwen-7B. Numbers in parentheses are reward density defined as #steps per reward.

7.1 What reward signals are needed for multi-turn RL training?

Dense rewards significantly improve multi-turn RL performance, with optimal density varying by algorithm. As shown in Table 7, reward density has different effects on PPO and RLOO. PPO benefits most from the most dense rewards (Dense 2), achieving 58% success compared to 41% with sparse rewards. RLOO shows robust performance across dense reward schemes, achieving 55% success with both Dense 1 and Dense 2 configurations. This consistency suggests that RLOO's unbiased gradient estimates are less sensitive to reward density.

The key takeaway here is that the choice of reward density should align with the selected optimization algorithm. Dense rewards may enable faster convergence in multi-turn RL, evidenced by the substantial performance gains observed across both algorithms. However, the effectiveness of dense rewards depends not only on frequency but also on the quality and consistency of the reward signal design, where poorly designed intermediate rewards may provide misleading guidance that impairs rather than improves learning.

[4] `https://textworld.readthedocs.io/en/stable/textworld.challenges.simple.html`

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL https://arxiv.org/abs/2402.14740.

Francois Chollet, Mike Knoop, Gregory Kamradt, and Bryan Landers. Arc prize 2024: Technical report, 2025. URL https://arxiv.org/abs/2412.04604.

Marc-Alexandre Côté, Akos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, et al. Textworld: A learning environment for text-based games. In *Workshop on Computer Games*, pages 41–75. Springer, 2018.

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. Agentic reinforced policy optimization, 2025. URL https://arxiv.org/abs/2507.19849.

Owen Oertell, Wenhao Zhan, Gokul Swamy, Zhiwei Steven Wu, Kiante Brantley, Jason Lee, and Wen Sun. Heuristics considered harmful: Rl with random rewards should not make llms reason, 2025. URL https://fuchsia-arch-d8e.notion.site/Heuristics-Considered-Harmful-RL-With-Random-Rewards-Should-Not-Make-LLMs-Reason-21ba29497c418

Jiayi Pan, Xingyao Wang, Graham Neubig, Navdeep Jaitly, Heng Ji, Alane Suhr, and Yizhe Zhang. Training software engineering agents and verifiers with swe-gym. In *Proceedings of the 42nd International Conference on Machine Learning (ICML 2025)*, 2025. URL https://arxiv.org/abs/2412.21139. arXiv:2412.21139, accepted at ICML 2025.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning, 2021. URL https://arxiv.org/abs/2010.03768.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning, 2025. URL https://arxiv.org/abs/2504.20073.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Siliang Zeng, Quan Wei, William Brown, Oana Frunza, Yuriy Nevmyvaka, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level credit assignment, 2025. URL https://arxiv.org/abs/2505.11821.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks, 2025. URL https://arxiv.org/abs/2503.15478.