FINE-GRAINED CREDIT ASSIGNMENT FOR MULTI-TURN CRITIC-FREE LLM REINFORCEMENT LEARNING WITH SEQUENTIAL EQUILIBRIUM OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Critic-free group-based methods have shown strong potential for single-turn large language model (LLM) reinforcement learning (RL), owing to their reduced memory requirements and stable optimization. However, complex tasks often require multi-turn interactions, where the sparsity and delayed nature of verifiable rewards pose a fundamental challenge for credit assignment. Although recent methods alleviate this issue by introducing additional step-level groups or related mechanisms, trajectory-level grouping remains an essential component of advantage estimation and thus continues to exert a substantial influence on optimization. This motivates us to ask whether credit assignment can be further improved from a complementary perspective by directly refining the computation of trajectory-level advantages. To this end, we propose **S**equential **E**quilibrium **P**olicy **O**ptimization (SEPO), a general credit assignment algorithm that can be flexibly integrated with a variety of existing group-based methods. SEPO introduces a novel multi-agent perspective that formulates multi-turn interaction as a fully cooperative sequential game. It then derives a solution to the subgame perfect Nash equilibrium through best-response optimization and aligns this solution procedure with multi-turn LLM RL, yielding more accurate trajectory-level credit assignment. Extensive experiments on ALFWorld, WebShop and QA tasks with Qwen2.5-1.5B/3B/7B-Instruct, and Qwen3-4B demonstrate that SEPO consistently improves over GRPO, GiGPO, and DAPO while holding the amount of data used for actor updates unchanged, up to 11 percentage points, validating the effectiveness and generality of its credit assignment mechanism.

1 INTRODUCTION

Large language models (LLMs) (Singh et al., 2025; Yang et al., 2025; Guo et al., 2025; Zeng et al., 2026) have demonstrated remarkable capabilities across a wide range of language understanding, reasoning, and decision-making tasks, and are increasingly serving as general-purpose agents in real-world applications (Ferrag et al., 2026). However, many complex tasks of growing practical importance, such as healthcare (Chen et al., 2026), mobile and web navigation (Liu et al., 2025a; Prabhu et al., 2026), scientific discovery (Ren et al., 2025), and coding (Dong et al., 2025), require LLMs to engage in multi-turn interactions with users, environments, or external tools, demanding strong long-horizon planning and decision-making capabilities. These requirements make reinforcement learning (RL) post-training increasingly important for further improving LLM capabilities.

Among existing RL algorithms for LLM post-training, critic-free group-based methods have emerged as a particularly promising paradigm due to their lower GPU memory requirements and improved training stability (Zhang et al., 2025). Representative approaches, including ReMax (Li et al., 2023), GRPO (Shao et al., 2024), and its subsequent variants such as Dr.GRPO (Liu et al., 2025b) and DAPO (Yu et al., 2026), have demonstrated strong performance on reasoning tasks and become an important foundation for scalable LLM RL. However, these methods are primarily designed for single-turn settings, when extended to multi-turn interactions, sparse and delayed verifiable rewards make it difficult to accurately assign credit across different turns. Recent methods such as GiGPO (Feng et al., 2026) and HGPO (He et al., 2026) address this challenge by introducing additional step-level groups to estimate step-level advantages, leading to improved long-horizon

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.

decision-making. Nevertheless, the original trajectory-level advantage inherited from GRPO remains a critical component of the overall advantage estimation and thus has a non-negligible impact on policy optimization. This observation raises a natural question: ***Can we directly refine trajectory-level advantage to improve the accuracy of credit assignment from a complementary perspective?***

To address this challenge, we propose **S**equential **E**quilibrium **P**olicy **O**ptimization (SEPO), a general credit assignment algorithm that can be seamlessly integrated with various existing group-based methods for multi-turn LLM RL. SEPO introduces a novel perspective by reformulating multi-turn interactions as a multi-agent cooperative problem and modeling them as fully cooperative sequential games. It then proposes a parallel best-response optimization procedure to solve the subgame perfect Nash equilibrium (SPNE) of the sequential game. Finally, SEPO incorporates this equilibrium-solving framework into multi-turn LLM RL optimization, assigning each groups trajectory-level advantage to its designated turn or chunk, thereby enabling more fine-grained credit assignment.

Extensive experiments on ALFWorld, WebShop, and seven search-augmented QA benchmarks, using Qwen2.5-1.5B/3B/7B-Instruct and Qwen3-4B, demonstrate the effectiveness and generality of SEPO. With the actor-update data budget held fixed, SEPO improves overall ALFWorld success rates and WebShop scores across all nine combinations of baseline methods and backbones. In particular, integrating SEPO with DAPO yields success-rate gains of 6-11 percentage points across the two environments. Improvements with GiGPO and DAPO on both single-hop and multi-hop QA further support its applicability to tool-augmented reasoning. Our analysis additionally identifies a trade-off between credit assignment granularity and the availability of informative samples, providing insight into the stronger gains observed when SEPO is combined with dynamic sampling.

2 Related Work

RL with Verifiable Rewards for LLMs. As the foundation capabilities of LLMs continue to improve, post-training has increasingly shifted from RL from Human Feedback (RLHF) toward RL with Verifiable Rewards (RLVR) to enhance reasoning and decision-making capabilities with objective and automatically verifiable feedback. In particular, critic-free methods have attracted growing attention by avoiding the additional cost and instability associated with value-function learning. ReMax (Li et al., 2023) uses greedy decoding rewards as a baseline, while RLOO (Ahmadian et al., 2024) constructs leave-one-out baselines from multiple sampled responses. GRPO (Shao et al., 2024) further estimates advantages using group-relative rewards, establishing a widely adopted framework for critic-free RLVR. Building on this paradigm, Dr.GRPO (Liu et al., 2025b) and CPPO (Lin et al., 2026) refine the optimization objective to alleviate undesirable biases and improve training behavior, while DAPO (Yu et al., 2026) further enhances training stability through improved clipping, sampling, and loss designs. Together, these methods have established critic-free group-based optimization as a strong paradigm for LLM RLVR.

RL for multi-turn LLM Agents. Beyond single-turn reasoning, recent studies have increasingly extended RL to multi-turn LLM agents that interact with environments and external tools over long horizons. Representative works such as WebRL (Qi et al., 2025), Search-R1 (Jin et al., 2025), RAGEN (Wang et al., 2025), and WebAgent-R1 (Wei et al., 2025b) demonstrate that outcome-based RL can effectively improve capabilities in web navigation, search, and interactive decision making. However, directly applying trajectory-level optimization to multi-turn interactions leads to a fundamental credit assignment challenge, as sparse and delayed outcome rewards cannot distinguish the contributions of individual turns. To address this issue, methods like MT-PPO (Wei et al., 2025a) construct dense turn-level rewards by estimating intermediate progress toward the final outcome. More closely related to our work, GiGPO (Feng et al., 2026) extends group-based RL with step-level relative advantages by grouping actions from shared anchor states, and HGPO (He et al., 2026) further introduces history-aware hierarchical grouping to mitigate context inconsistency in step-wise advantage estimation. Despite these advances, existing methods primarily obtain fine-grained credit through additional intermediate signals or local comparison groups, leaving trajectory-level relative advantages largely shared across turns. Our work instead focuses on how such trajectory-level credit should be explicitly attributed to individual turns according to their influence on the final outcome. More related work can be found in Appendix B.

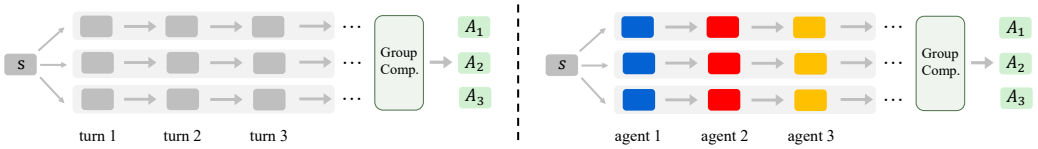


Figure 1: Illustration of multi-agent modeling of multi-turn interactions.

3 PRELIMINARIES

Problem setup. We consider a general multi-turn decision-making setting, where an LLM agent interacts with an environment to accomplish a task specified by an instruction $x \sim p(X)$. The interaction proceeds for T turns. At turn t, the agent receives the current state $s_t \in \mathcal{S}$ and produces a textual action $a_t \in \mathcal{V}^{\leq n}$, where $\mathcal{V}$ is the token vocabulary and n denotes the maximum response length. After executing a_t, the environment provides a scalar reward $r_t \in \mathbb{R}$ and transitions to the next state s_{t+1}. An episode can therefore be represented by a trajectory $\tau = \{(s_t, a_t, r_t)\}_{t=1}^{T}$. The agent is parameterized by an LLM policy $\pi_\theta(a_t \mid s_t, x)$, which generates each action conditioned on the task instruction and the current interaction state. In practical multi-turn tasks, we consider a sparse-reward setting, where a task-level reward indicating success or failure is only available upon task completion, while intermediate steps receive either format-based rewards or zero reward.

Group-based RL. In group-based RL, given an initial task instruction x, the LLM samples a group of N trajectories $\{\tau_1, \ldots, \tau_N\}$ conditioned on x and obtains their corresponding cumulative rewards $\{R(\tau_1), \ldots, R(\tau_N)\}$. The advantages are then computed from the reward statistics within the group according to a specific group-based estimator. For example, GRPO normalizes the cumulative reward of each trajectory using the mean and standard deviation of the group rewards, and uses the resulting normalized rewards as the advantages for policy optimization.

4 METHOD

4.1 MULTI-AGENT MODELING OF MULTI-TURN INTERACTIONS

Credit assignment under sparse rewards is a central challenge in multi-turn LLM post-training. Interestingly, a closely related problem has long been studied in multi-agent games and RL: when multiple agents cooperate to accomplish a shared objective, how should the contribution of each individual agent to the final outcome be quantified? This parallel motivates us to ask: **whether ideas from multi-agent scenarios can be transferred to multi-turn LLM RL, providing a more principled way to assign credit across different turns.**

Surprisingly, we observe that multi-turn interactions are also structurally analogous to multi-agent systems. If each turn is viewed as an individual agent making a decision, the multi-turn process can be naturally formulated as a parameter-sharing multi-agent system, where all agents share the same LLM policy while being distinguished by their turn IDs encoded in the prompt, as illustrated in Figure 1. Formally, under this multi-agent perspective, we model multi-turn interactions as a fully cooperative sequential game:

Definition 4.1 (Fully cooperative sequential game)**.** In a fully cooperative sequential game, agents act in a fixed order $1 \to 2 \to \cdots \to T$. Before making a decision, agent i observes the history of all preceding actions $h_i = (a_1, \ldots, a_{i-1}) \in \mathcal{H}_i$, $\mathcal{H}_i$ denotes history space for agent i, where the environment state, if needed, can be included in h_i. Its policy is a history-dependent mapping

$$\pi_i : \mathcal{H}_i \to \mathcal{A}_i(h_i), \tag{1}$$

and the joint policy is $\pi = (\pi_1, \ldots, \pi_T)$. Executing π induces a terminal trajectory $z = (a_1, \ldots, a_T)$, for which all agents share the same utility $R(z)$. The team value and action-value functions are defined recursively as

$$V_{T+1}^{\pi}(z) = R(z), \quad V_i^{\pi}(h_i) = V_{i+1}^{\pi}(h_i, \pi_i(h_i)), \quad \text{and} \quad Q_i^{\pi}(h_i, a_i) = V_{i+1}^{\pi}(h_i, a_i). \tag{2}$$

4.2 Fine-grained Credit Assignment with Parallel Best-response Update

Based on the formulation above, policy optimization can be viewed as solving for a game equilibrium. In sequential games, the solution comes to the subgame perfect equilibrium (SPE), a refinement of Nash equilibrium (NE) that requires equilibrium behavior in subgame under every history:

Definition 4.2 (SPE (Fudenberg & Levine, 1983))**.** For a fully cooperative sequential game, define the optimal subgame value as $V^*_{T+1}(z) = R(z)$, and, for $i = T, T-1, \ldots, 1$, $V^*_i(h_i) = \max_{a_i} V^*_{i+1}(h_i, a_i)$. A policy $\pi^* = (\pi^*_1, \ldots, \pi^*_T)$ is a SPE if, for every agent i and every history h_i,

$$\pi^*_i(h_i) \in \arg\max_{a_i} V^*_{i+1}(h_i, a_i). \tag{3}$$

For a fully cooperative sequential game, a NE can be naturally approached through best-response updates, where $\mathrm{BR}^\pi_i(h_i) = \arg\max_{a_i \in A_i(h_i)} Q^\pi_i(h_i, a_i)$. A natural question is whether the same update principle can also recover a SPE. The following theorem provides an affirmative answer:

Definition 4.3 (Parallel best-response update)**.** Given a policy π^t, at iteration t, let $B_t \subseteq \{1, \ldots, T\}$ be a non-empty set of agents selected for simultaneous updates. For each $i \in B_t$ and selected $h_i \in \mathcal{H}_i$, define the local improvement gap as

$$\Delta^{\pi^t}_i(h_i) = \max_{a_i \in \mathcal{A}_i(h_i)} Q^{\pi^t}_i(h_i, a_i) - Q^{\pi^t}_i\left(h_i, \pi^t_i(h_i)\right). \tag{4}$$

Parallel best-response update optimizes the policy as

$$\pi^{t+1}_i(h_i) \in \begin{cases} \mathrm{BR}^{\pi^t}_i(h_i), & \Delta^{\pi^t}_i(h_i) > 0, \\ \pi^t_i(h_i), & \Delta^{\pi^t}_i(h_i) \le 0, \end{cases} \qquad i \in B_t, \tag{5}$$

while $\pi^{t+1}_j = \pi^t_j,\ j \notin B_t$.

Theorem 4.4. *Consider a fully cooperative sequential game satisfying the following conditions: (i) policy decisions are independently parameterized across agents and histories; (ii) all history and action spaces are finite; and (iii) the selection schedule is fair, i.e., every agent-history pair (i, h_i) is selected infinitely often. Starting from any initial policy π^0, the parallel best-response updates stabilize after finitely many strict improvements at a policy π^∞ satisfying*

$$\pi^\infty_i(h_i) \in \arg\max_{a_i} V^{\pi^\infty}_{i+1}(h_i, a_i), \quad \forall i, \forall h_i. \tag{6}$$

Consequently, π^∞ is a SPE and attains the team-optimal value in every subgame.

Remark. While Theorem 4.4 establishes the SPE convergence of parallel best-response updates, their benefit for credit assignment stems from a key property of best-response optimization: when optimizing one agent, all other agents are held fixed. Consequently, changes in the team objective can be attributed more directly to the updated agent, substantially reducing interference from other agents and enabling a finer-grained estimate of individual contribution. Viewed from this perspective, the underlying principle is closely related to classical multi-agent credit assignment methods such as Shapley values (Winter, 2002) and counterfactual value estimation (Foerster et al., 2018). Although parameter sharing in LLMs departs from the agent- and history-wise independence assumed in Theorem 4.4, we hypothesize that their large parameter capacity, together with conditioning on distinct prompts and interaction histories, can partially mitigate the practical effects of this coupling.

4.3 Align Sequential Equilibrium Optimization with Group-based Methods

The previous section introduced a method for solving fully cooperative sequential games to achieve more effective credit assignment. However, the formulation is developed for conventional game settings and is not fully aligned with the structure of group-based multi-turn LLM RL. Therefore, we further adapt the method to bridge this gap.

The key to our method is the best-response update, which selects an agent i and a history h_i for optimization while keeping the behaviors of the remaining agents fixed. To emulate this process in group-based multi-turn LLM RL, we modify the group rollout procedure, as illustrated in Figure 2. Specifically, updating agent i requires first conditioning on a valid history h_i. This implies that

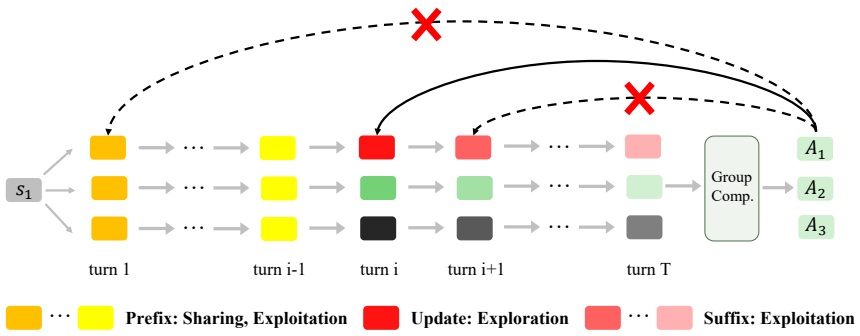


Figure 2: Illustration of rollout process.

the corresponding group should be constructed from h_i, rather than independently from the initial state x. To achieve this, we introduce **prefix sharing**, where the decisions of agents $1, \ldots, i-1$ are shared across all rollouts within the group, ensuring an identical interaction prefix and thus a common history h_i for updating agent i. Although Theorem 4.4 requires fair sampling over histories h_i to guarantee convergence, we generate shared prefixes using the base method's validation temperature, concentrating local comparisons on histories encountered under its evaluation policy. This choice determines the distribution of training histories, and its effect is examined through the prefix-temperature ablation in Section 5.5.

For the agent being updated, although it is infeasible to enumerate the entire action space as required by an exact best-response computation, we can approximate this process by **exploration**, sampling N candidate actions $\{a_{i,1}, \ldots, a_{i,N}\}$, where N is the group size. For each candidate action, we estimate its action value $Q_i^\pi(h_i, a_{i,j})$ through sampling. Specifically, after taking $a_{i,j}$, we let the subsequent agents $i+1, \ldots, T$ act **exploitatively**, which corresponds to a single-trajectory Monte Carlo estimate of $Q_i^\pi(h_i, a_{i,j})$. Given the sampling procedure above, the mean group return estimates the average action value of the exploratory candidate distribution of the history h_i. Denote this history-dependent baseline by $b_i(h_i)$, subtracting this preserves the best-response objective:

$$\arg\max_{a_i} [Q_i^\pi(h_i, a_i) - b_i(h_i)] = \arg\max_{a_i} Q_i^\pi(h_i, a_i). \tag{7}$$

We therefore use the group-relative return $A^E(\tau_j) = \frac{R(\tau_j) - \text{mean}\left(\{R(\tau_\ell)\}_{\ell=1}^N\right)}{F_{\text{norm}}\left(\{R(\tau_\ell)\}_{\ell=1}^N\right)}$ as a sampled local comparison signal, where τ_j denotes the j-th trajectory in the group, and F_{norm} is an action-independent factor. For a fixed sampled group with a positive normalization factor, centering and normalization preserve the ordering of sampled returns. The resulting advantages provide a sampled approximation to the action comparison underlying best-response selection, encouraging candidates with above-average returns. Assigning these advantages only to the selected turn focuses the training objective on the decision whose alternatives are being compared. Let H_i denote the random history at turn i, A_i the current action, and Ω_i the subsequent rollout randomness, such that $R = R(H_i, A_i, \Omega_i)$. Define $\tilde{V}_i^\pi(H_i) = \mathbb{E}[R \mid H_i]$ and $\tilde{Q}_i^\pi(H_i, A_i) = \mathbb{E}[R \mid H_i, A_i]$. We have

Proposition 4.5 (Prefix-conditioned credit variance)**.** *The trajectory-level return variance satisfies*

$$\mathcal{V}_i^{\text{traj}} := \text{Var}(R \mid x) = \underbrace{\text{Var}_{H_i|x}[\tilde{V}_i^\pi(H_i)]}_{\text{prefix history variation}} + \underbrace{\mathbb{E}_{H_i|x}[\text{Var}(R \mid H_i)]}_{\mathcal{V}_i^{\text{SEPO}}}. \tag{8}$$

Therefore, $\mathcal{V}_i^{\text{SEPO}} \leq \mathcal{V}_i^{\text{traj}}$*, with strict inequality whenever* $\tilde{V}_i^\pi(H_i)$ *varies across histories, and*

$$\text{Var}(R \mid H_i) = \underbrace{\text{Var}_{A_i|H_i}[\tilde{Q}_i^\pi(H_i, A_i)]}_{\text{action-value signal}} + \underbrace{\mathbb{E}_{A_i|H_i}[\text{Var}(R \mid H_i, A_i)]}_{\text{suffix noise}}. \tag{9}$$

Thus, **prefix sharing** removes the variation induced by different prefix histories while preserving the action-value variation required for best-response optimization; **suffix exploitation** reduces the stochasticity of subsequent decisions, yielding a more focused credit signal for the selected turn.

Then, under SEPO, each rollout group is used to update only one turn (i.e., one agent). Therefore, matching the amount of training data used for policy updates in conventional methods would require

approximately T times as many rollout groups, resulting in substantial rollout overhead. To better balance computational efficiency and credit-assignment granularity, we introduce **turn chunking**. Instead of treating every individual turn as an agent, we group several consecutive turns into a single agent and partition the entire multi-turn trajectory into K turn chunks. Each rollout group then targets one of these K chunks for optimization, reducing the required rollout expansion from T-fold to K-fold while retaining finer-grained credit assignment than trajectory-level optimization.

Finally, the group advantage A^E can serve as a drop-in replacement for the original trajectory-level advantage in different group-based RL frameworks: be directly used as the advantage for policy optimization (GRPO); combined with an additional step-level advantage A^S (GiGPO et. al.); or incorporated with additional sampling strategies such as dynamic sampling (DAPO). Suppose all turns are partitioned into K chunks, with boundaries $\{t_1, \ldots, t_K, t_{K+1}\}$, the resulting advantage is then optimized using the PPO objective:

$$\mathcal{J}(\theta) = \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}_{\substack{x\sim p(X) \\ \{\tau_j\}_{j=1}^{N}}} \left[\frac{K}{NT} \sum_{j=1}^{N} \sum_{i=t_k}^{t_{k+1}-1} \min\Big(\rho_\theta(a_{i,j})\, A(a_{i,j}) \,,\, \mathrm{clip}(\rho_\theta(a_{i,j}) \,, 1\pm\epsilon)\, A(a_{i,j}) \Big) \right] - \beta D_{\mathrm{KL}}\left(\pi_\theta(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)\right), \tag{10}$$

where $\rho_\theta(a_{i,j}) = \frac{\pi_\theta(a_{i,j}|h_{i,j})}{\pi_{\theta_{\mathrm{old}}}(a_{i,j}|h_{i,j})}$ is the importance sampling ratio, β controls the strength of the KL penalty encouraging proximity to a reference policy π_{ref}.

5 Experiments

In this section, we present empirical evaluations of SEPO across a variety of base models and tasks, aiming to answer the following questions: (1) Can SEPO consistently improve the performance of group-based RL methods (Section 5.2)? (2) What explains the performance gains of SEPO (Section 5.3)? (3) How does the number of chunks K affect the performance of SEPO (Section 5.4)? (4) How do individual components contribute to the performance of SEPO (Section 5.5)?

5.1 Experiment Setup

Benchmarks. We evaluate SEPO on three types of multi-turn tasks: household task execution, web-based shopping, and search-augmented QA. ALFWorld (Shridhar et al., 2020) requires agents to accomplish household goals through textual observations and actions. It covers six task types: pick and place (Pick), examine in light (Look), clean and place (Clean), heat and place (Heat), cool and place (Cool), and pick two and place (Pick2). WebShop (Yao et al., 2022) simulates online shopping, where agents search for products, navigate webpages, select options, and complete purchases according to natural language instructions. Terminal rewards reflect how well the purchased item satisfies the specified product type, attributes, options, and price constraints. For search-augmented QA, agents interleave reasoning with search calls to gather evidence and produce answers (Jin et al., 2025). We use three single-hop datasets-NQ (Kwiatkowski et al., 2019), TriviaQA (Joshi et al., 2017), and PopQA (Mallen et al., 2023)-and four multi-hop datasets-HotpotQA (Yang et al., 2018), 2Wiki (Ho et al., 2020), MuSiQue (Trivedi et al., 2022), and Bamboogle (Press et al., 2023).

Baselines. To study credit assignment in critic-free group-based RL, we select three widely used and competitive baselines. GRPO (Shao et al., 2024) estimates advantages from groups of trajectories, while GiGPO (Feng et al., 2026) augments GRPO with step-level advantages for finer-grained credit assignment. DAPO (Yu et al., 2026) incorporates techniques such as dynamic sampling to increase the proportion of samples with non-zero advantages in each update, providing more informative training signals under sparse rewards. We integrate SEPO with each baseline, yielding GRPO+SEPO, GiGPO+SEPO, and DAPO+SEPO, to assess whether our approach consistently improves credit assignment across different group-based RL methods.

Settings. We use Qwen2.5-1.5B/3B/7B-Instruct and Qwen3-4B as backbone models. We train for 150 steps on ALFWorld and WebShop and 200 steps on QA tasks, following prior work (Feng et al., 2026). All WebShop and ALFWorld results are averaged over three random seeds to account for

Table 1: Performance on ALFWorld and WebShop. Shaded rows denote methods enhanced with SEPO. Blue bold values indicate the better result within each baseline–SEPO pair. Some baseline results for Qwen2.5-7B-Instruct are taken from Feng et al. (2026).

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Qwen2.5-1.5B-Instruct									
GRPO	0.85	0.43	**0.80**	0.60	0.58	0.48	0.66	0.78	0.63
GRPO+SEPO	**0.89**	**0.77**	0.68	**0.81**	**0.73**	**0.62**	**0.75**	**0.81**	**0.71**
GiGPO	0.93	0.56	**0.97**	0.94	**0.92**	0.69	0.89	0.80	0.65
GiGPO+SEPO	**0.95**	**0.86**	0.88	**1.00**	**0.92**	**0.84**	**0.91**	**0.84**	**0.71**
DAPO	0.92	0.71	0.88	**0.97**	0.82	0.65	0.82	0.83	0.73
DAPO+SEPO	**0.98**	**0.94**	**0.93**	**0.97**	**0.93**	**0.84**	**0.93**	**0.93**	**0.84**
Qwen3-4B									
GRPO	**0.94**	0.69	0.79	0.58	0.68	**0.72**	0.77	0.85	0.76
GRPO+SEPO	0.84	**0.75**	**0.93**	**0.82**	**0.76**	0.67	**0.80**	**0.88**	**0.77**
GiGPO	**1.00**	**0.91**	0.94	**1.00**	0.83	0.77	0.91	0.84	**0.72**
GiGPO+SEPO	0.96	0.87	**1.00**	0.92	**0.93**	**0.84**	**0.93**	**0.86**	0.71
DAPO	0.99	0.87	0.83	0.95	0.85	0.73	0.87	0.92	0.86
DAPO+SEPO	**1.00**	**1.00**	**0.98**	**0.98**	**1.00**	**0.95**	**0.98**	**0.97**	**0.92**
Qwen2.5-7B-Instruct									
GRPO	0.91	0.66	0.89	0.75	**0.73**	**0.65**	0.78	0.79	0.66
GRPO+SEPO	**0.96**	**0.70**	**0.98**	**0.80**	0.68	**0.65**	**0.81**	**0.81**	**0.72**
GiGPO	**0.98**	0.83	**0.99**	0.84	**0.89**	0.79	0.91	0.84	0.73
GiGPO+SEPO	0.94	**0.90**	**0.99**	**0.97**	0.88	**0.92**	**0.94**	**0.89**	**0.77**
DAPO	**0.99**	0.86	0.95	0.83	0.81	**0.90**	0.90	0.91	0.80
DAPO+SEPO	**0.99**	**0.96**	**0.98**	**0.92**	**1.00**	**0.90**	**0.96**	**0.97**	**0.90**

Table 2: Performance on search-augmented QA benchmarks. We report results as percentages to make small performance differences more apparent. For these experiments, we report results from a single run, as both the environment and the validation procedure are deterministic.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Qwen2.5-3B-Instruct								
GiGPO	42.0	59.5	42.4	36.9	37.0	12.6	64.1	42.1
GiGPO+SEPO	**43.9**	**60.6**	**45.8**	**37.2**	**37.5**	**13.4**	**65.3**	**43.4**
DAPO	47.5	63.2	47.9	44.0	43.3	**18.2**	67.7	47.4
DAPO+SEPO	**48.1**	**63.6**	**48.6**	**45.8**	**45.2**	**18.2**	**69.4**	**48.4**

variability across runs. We partition each trajectory into $K = 4$ turn chunks, treating each chunk as an agent. All other hyperparameters are kept identical to those of the corresponding baselines.

5.2 Main Comparison

ALFWorld and WebShop. We first evaluate SEPO on ALFWorld and WebShop across multiple backbones and baselines. As shown in Table 1, SEPO improves the overall ALFWorld success rate and WebShop score across all nine baseline–backbone combinations, with ALFWorld gains ranging from 2 to 11 percentage points. WebShop success rates also improve in eight of the nine settings. The training curves in Figure 3(a,b) further show that these benefits emerge during training and persist across much of the optimization process. The largest gains are observed when combining SEPO with DAPO. Across the six backbone–environment combinations, DAPO+SEPO improves

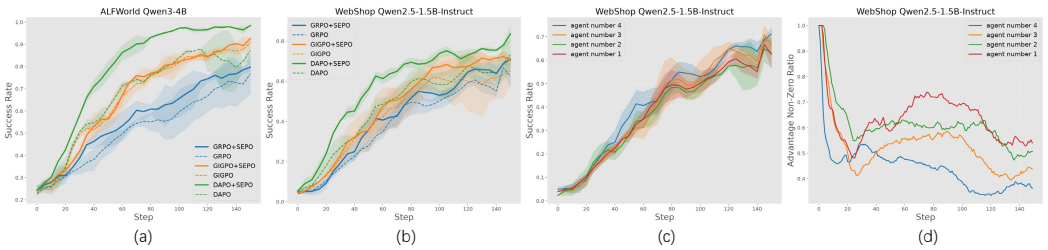


Figure 3: (a), (b) Training curves of Qwen3-4B on ALFWorld and Qwen2.5-1.5B-Instruct on WebShop. (c), (d) Success rate and non-zero advantage rate of different agent number K under GRPO. For (a), (b), and (c), results are averaged over 3 seeds and smoothed by window size of 3. For (d), results are smoothed by window size of 20.

success rates by 6–11 percentage points and achieves at least 90% success in five settings. We hypothesize that this stronger benefit reflects an interaction between more focused credit assignment and dynamic sampling. While SEPO reduces return variability unrelated to the selected chunk, it may also produce more groups with identical returns and thus zero relative advantages. Without dynamic sampling, this can reduce the number of informative samples per update, even when the total update batch size remains unchanged. We further examine this explanation in the subsequent experiments. Although the magnitude of improvement varies across settings, the aggregate gains support the applicability of SEPO across different backbones and group-based RL methods. In particular, improvements over GiGPO and DAPO suggest that refining trajectory-level credit assignment complements step-level advantage estimation and dynamic sampling.

Search-augmented QA. We further evaluate SEPO on search-augmented QA tasks involving explicit tool calls. As shown in Table 2, the results are consistent with our earlier findings: integrating SEPO with GiGPO or DAPO generally improves performance across both single-hop and multi-hop benchmarks, further supporting the generality of its credit assignment mechanism.

5.3 Simple Case Study

To better understand the credit assignment challenges faced by conventional group-based methods in multi-turn settings, we compare GRPO and SEPO on a sampled QA trajectory in Figure 4. The final answer is semantically correct but marked incorrect because it uses a numeral rather than the English wording. GRPO assigns the resulting negative advantage to every turn, although the first search is appropriate and the second is somewhat redundant without compromising answer correctness. SEPO instead distinguishes their contributions. The first turn receives zero advantage because its candidate searches all yield successful continuations under suffix validation. The second receives positive advantage because some other alternative searches lead to failure. Only the final turn receives negative advantage due to the answer-format mismatch. This case illustrates how focusing credit on relevant turns can avoid propagating a terminal verification failure to otherwise reasonable decisions.

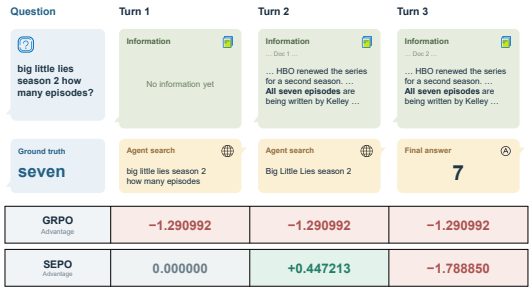


Figure 4: A Case study on QA task.

5.4 Chunk Number Analysis

Since SEPO models consecutive turns as agent chunks, the number of chunks K is a central hyperparameter. Thus, we investigate its effect in Figure 3(c). As shown, larger values of K generally yield higher success rates, with $K = 3, 4$ outperforming $K = 1, 2$, although the incremental gains from increasing K remain limited. To investigate this behavior, we further visualize the non-zero

advantage ratio in Figure 3(d), which clearly decreases as K increases. We interpret this pattern as a trade-off between credit assignment granularity and the availability of informative samples. Increasing K shortens the chunk optimized within each rollout, leaving more turns subject to prefix sharing or suffix exploitation. This can reduce within-group return diversity and increase the frequency of groups with identical returns, whose relative advantages are zero. The resulting reduction in informative samples may partially offset the benefits of finer-grained credit assignment. These observations support our earlier explanation for the stronger gains with DAPO, whose dynamic sampling can mitigate this effect, and are consistent with the variance-reduction rationale in Proposition 4.5. Choosing K therefore requires balancing credit assignment granularity, rollout cost, and the proportion of informative samples.

5.5 Ablation Studies

Finally, we conduct ablation studies and additional analyses to examine the effectiveness of SEPO. To assess the contributions of individual components, we consider four variants: **w/o prefix validation** explores prefix histories while sharing them within each group; **w/o prefix sharing** independently samples prefixes using exploitation; **w/o suffix validation** explores suffix turns; and **only suffix validation** independently explores prefix turns within each group, retaining exploitation only in suffix turns. We further examine two practical variants. **update suffix** adds otherwise unused suffix turns to policy updates and increases the mini-batch size to keep the total number of gradient updates approximately unchanged, testing whether additional training data improves performance under a comparable update budget. **warm up** uses standard GRPO for the first 75 training steps and incorporates SEPO for the remaining 75 steps, evaluating its effectiveness as a late-stage refinement mechanism with reduced rollout overhead.

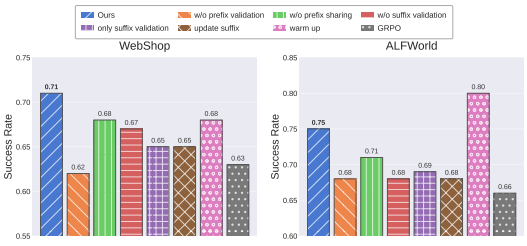


Figure 5: Ablation and more analysis studies with GRPO on WebShop and ALFWorld.

As shown in Figure 5, **w/o prefix validation** exhibits the largest performance degradation, suggesting that exploring prefix histories can reduce learning efficiency even when histories are shared within each group. Both **w/o prefix sharing** and **w/o suffix validation** also degrade performance, supporting the benefit of limiting variability outside the updated chunk. The small impact of removing prefix sharing may reflect the similar histories already produced by low-temperature sampling. The larger degradation under **only suffix validation**, which independently explores prefixes, is consistent with this interpretation. Furthermore, **update suffix** provides no clear improvement under an approximately fixed gradient-update budget, suggesting that incorporating additional turns outside the selected chunk does not necessarily improve optimization. Finally, compared with applying SEPO throughout training, **warm up** slightly reduces WebShop performance and improves ALFWorld performance. These results suggest that SEPO can effectively refine policies during later training stages, retaining its credit assignment benefits while limiting additional rollout overhead.

6 Conclusion

In this paper, we presented SEPO, a general credit assignment algorithm for critic-free group-based multi-turn LLM RL. By modeling turn chunks as agents in a fully cooperative sequential game, SEPO translates best-response optimization into a rollout procedure that assigns trajectory-level credit to selected chunks, reducing interference from return variability induced by turns outside the updated chunk, enabling finer-grained credit assignment. Experiments on ALFWorld, WebShop, and search-augmented QA show improvements across multiple backbones and group-based methods, supporting its effectiveness and generality. A limitation of SEPO is the additional rollout time required to construct chunk-specific groups while maintaining a comparable actor-update data budget. Future work could explore adaptive turn chunking and rollout reuse to reduce this overhead, and extend the framework to longer-horizon tasks with more diverse tool interactions.

AI USE STATEMENT

In this work, we used generative AI to assist with translation, language editing, and the preparation of tables and presentation slides. These tools were not used to generate the underlying experimental data. We take responsibility for the accuracy and integrity of the final content of this work.

REPRODUCIBILITY STATEMENT

We provide complete and executable code in the supplementary materials. Therefore, the reported results can be readily reproduced.

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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 12248–12267, 2024.

Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Advances in Neural Information Processing Systems*, 32, 2019.

Lang Cao, Hui Ruan, Yongqian Li, Peng Chao, Wu Ning, Haonan Song, Renhong Chen, and Yitong Li. Treeadv: Tree-structured advantage redistribution for group-based rl. *arXiv preprint arXiv:2601.03703*, 2026.

Sully F Chen, Anton Alyakin, Andreas Seas, Eunice Yang, Joanne J Choi, Jin Vivian Lee, Amelia L Chen, Pranav I Warman, Rochelle T Bitolas, Robert J Steele, et al. Llm-assisted systematic review of large language models in clinical medicine. *Nature medicine*, 32(3):1152–1159, 2026.

Yihong Dong, Xue Jiang, Jiaru Qian, Tian Wang, Kechi Zhang, Zhi Jin, and Ge Li. A survey on code generation with llm-based agents. *arXiv preprint arXiv:2508.00083*, 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.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *IEEE Access*, 2026.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Drew Fudenberg and David Levine. Subgame-perfect equilibria of finite-and infinite-horizon games. *Journal of Economic Theory*, 31(2):251–268, 1983.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, et al. Hindsight credit assignment. *Advances in neural information processing systems*, 32, 2019.

Shuo He, Lang Feng, Xin Cheng, Lei Feng, Bo An, et al. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *International Conference on Learning Representations*, volume 2026, pp. 27572–27593, 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.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. Treerl: Llm reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, 2025.

Tianyi Hu, Qingxu Fu, Yanxi Chen, Zhaoyang Liu, and Bolin Ding. Seeupo: Sequence-level agentic-rl with convergence guarantees. *arXiv preprint arXiv:2602.06554*, 2026.

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.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Refining credit assignment in rl training of llms. *arXiv preprint arXiv:2410.01679*, 2024.

Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. Trust region policy optimisation in multi-agent reinforcement learning. *arXiv preprint arXiv:2109.11251*, 2021.

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.

Yugu Li, Jimmy Cao, Jianglin Qiao, and Siyi Hu. Rtpo: Reverse-turn policy optimization for stabilizing agentic rl training. *arXiv preprint arXiv:2608.18682*, 2026.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *Advances in Neural Information Processing Systems*, 38:61043–61068, 2026.

Guangyi Liu, Pengxiang Zhao, Yaozhen Liang, Liang Liu, Yaxuan Guo, Han Xiao, Weifeng Lin, Yuxiang Chai, Yue Han, Shuai Ren, et al. Llm-powered gui agents in phone automation: Surveying progress and prospects. *arXiv preprint arXiv:2504.19838*, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025b.

Hao Ma, Tianyi Hu, Zhiqiang Pu, Boyin Liu, Xiaolin Ai, Yanyan Liang, and Min Chen. Coevolving with the other you: Fine-tuning llm with sequential cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:15497–15525, 2024.

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.

Viraj Prabhu, Yutong Dai, Matthew Fernandez, Krithika Ramakrishnan, Jing Gu, Yanqi Luo, Caiming Xiong, Junnan Li, Zeyuan Chen, Ran Xu, et al. Walt: Web agents that learn tools. In *International Conference on Learning Representations*, volume 2026, pp. 55589–55607, 2026.

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.

Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Jiadai Sun, Xinyue Yang, Yu Yang, Shuntian Yao, Wei Xu, et al. Webrl: Training llm web agents via self-evolving online curriculum reinforcement learning. In *International Conference on Learning Representations*, volume 2025, pp. 79791–79821, 2025.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

Shuo Ren, Can Xie, Pu Jian, Zhenjiang Ren, Chunlin Leng, and Jiajun Zhang. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047*, 2025.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025.

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.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

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.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, 2024.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, et al. Reinforcing multi-turn reasoning in llm agents via turn-level reward design. *arXiv preprint arXiv:2505.11821*, 2025a.

Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, et al. Webagent-r1: Training web agents via end-to-end multi-turn reinforcement learning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7920–7939, 2025b.

Eyal Winter. The shapley value. *Handbook of game theory with economic applications*, 3:2025–2054, 2002.

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, 2026.

Yifan Zang, Jinmin He, Kai Li, Haobo Fu, Qiang Fu, and Junliang Xing. Sequential cooperative multi-agent reinforcement learning. In *Proceedings of the 2023 international conference on autonomous agents and multiagent systems*, pp. 485–493, 2023.

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, Bo Qian, Peng Chen, and Jie Jiang. At^{2}po: Agentic turn-based policy optimization via tree search. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 24120–24143, 2026.

A Mathematical Proofs

A.1 Proof of Theorem 4.4

Proof. We work with the deterministic policies in Definition 4.1. Fix an iteration t and write $\pi = \pi^t$ and $\pi' = \pi^{t+1}$. Since policy decisions are independently parameterized across agents and histories, updating the selected agent-history pairs leaves all unselected decisions unchanged. At each selected pair, Definition 4.3 either replaces the current action by an exact best response with a strictly positive improvement gap or retains the current action. Consequently, for every i and $h_i \in \mathcal{H}_i$,

$$Q_i^{\pi}\big(h_i, \pi_i'(h_i)\big) \geq Q_i^{\pi}\big(h_i, \pi_i(h_i)\big) = V_i^{\pi}(h_i), \tag{11}$$

and the inequality is strict whenever $\pi_i'(h_i) \neq \pi_i(h_i)$. All action values in this comparison are evaluated under the same pre-update policy π.

We first show by backward induction that $V_i^{\pi'}(h_i) \geq V_i^{\pi}(h_i)$ at every history. At a terminal history z, we have $V_{T+1}^{\pi'}(z) = R(z) = V_{T+1}^{\pi}(z)$. Suppose the inequality holds at every history in $\mathcal{H}_{i+1}$. Then, for any $h_i \in \mathcal{H}_i$,

$$\begin{aligned} V_i^{\pi'}(h_i) &= V_{i+1}^{\pi'}\big(h_i, \pi_i'(h_i)\big) \\ &\geq V_{i+1}^{\pi}\big(h_i, \pi_i'(h_i)\big) \\ &= Q_i^{\pi}\big(h_i, \pi_i'(h_i)\big) \\ &\geq Q_i^{\pi}\big(h_i, \pi_i(h_i)\big) = V_i^{\pi}(h_i). \end{aligned} \tag{12}$$

The first inequality follows from the induction hypothesis, and the second from the update rule. This proves the claim for all stages. Moreover, $V_i^{\pi'}(h_i) > V_i^{\pi}(h_i)$ whenever the action at (i, h_i) changes.

Now define the potential

$$\Psi(\pi) = \sum_{i=1}^{T} \sum_{h_i \in \mathcal{H}_i} V_i^{\pi}(h_i), \tag{13}$$

where the sum includes all histories, including those not reached under π. The preceding inequalities imply $\Psi(\pi^{t+1}) \geq \Psi(\pi^t)$ at every iteration, with strict inequality whenever $\pi^{t+1} \neq \pi^t$. Because the horizon, history spaces, and action spaces are finite, there are only finitely many deterministic joint policies. Thus, Ψ takes only finitely many values, and the joint policy can change only finitely many times. The rule that retains the current action whenever there is no strict improvement excludes changes between equally valued actions. It follows that there exist a finite iteration t_0 and a policy π^∞ such that $\pi^t = \pi^\infty$ for all $t \geq t_0$.

Suppose that π^∞ fails the stated optimality condition at some pair (i, h_i). Then

$$\max_{a_i \in \mathcal{A}_i(h_i)} Q_i^{\pi^\infty}(h_i, a_i) > Q_i^{\pi^\infty}\big(h_i, \pi_i^\infty(h_i)\big). \tag{14}$$

By fairness, this pair is selected at some iteration $t \geq t_0$. Since $\pi^t = \pi^\infty$, its improvement gap is strictly positive, so the update must change its action. This contradicts stabilization. Therefore,

$$\pi_i^\infty(h_i) \in \operatorname*{arg\,max}_{a_i \in \mathcal{A}_i(h_i)} Q_i^{\pi^\infty}(h_i, a_i) = \operatorname*{arg\,max}_{a_i \in \mathcal{A}_i(h_i)} V_{i+1}^{\pi^\infty}(h_i, a_i) \tag{15}$$

for every agent and every history.

Finally, we show that this condition implies optimality in every subgame. At terminal histories, $V_{T+1}^{\pi^\infty}(z) = R(z) = V_{T+1}^*(z)$. Assume that $V_{i+1}^{\pi^\infty}(h_{i+1}) = V_{i+1}^*(h_{i+1})$ for every $h_{i+1} \in \mathcal{H}_{i+1}$. Then, for every $h_i \in \mathcal{H}_i$,

$$\begin{aligned} V_i^{\pi^\infty}(h_i) &= \max_{a_i \in \mathcal{A}_i(h_i)} V_{i+1}^{\pi^\infty}(h_i, a_i) \\ &= \max_{a_i \in \mathcal{A}_i(h_i)} V_{i+1}^*(h_i, a_i) = V_i^*(h_i). \end{aligned} \tag{16}$$

Backward induction therefore gives $V_i^{\pi^\infty}(h_i) = V_i^*(h_i)$ for all agents and histories. The same induction also yields

$$\pi_i^\infty(h_i) \in \operatorname*{arg\,max}_{a_i \in \mathcal{A}_i(h_i)} V_{i+1}^*(h_i, a_i), \qquad \forall i,\ \forall h_i \in \mathcal{H}_i. \tag{17}$$

Hence π^∞ satisfies Definition 4.2: it is a SPE and attains the team-optimal value in every subgame. □

A.2 Proof of Proposition 4.5

Proof. Fix a task instruction x and the rollout distribution used to define $\widetilde{V}_i^\pi$ and $\widetilde{Q}_i^\pi$, and assume $\mathbb{E}[R^2 \mid x] < \infty$. For brevity, write $\mu_x = \mathbb{E}[R \mid x]$, $U = \widetilde{V}_i^\pi(H_i) = \mathbb{E}[R \mid H_i]$, and $W = \widetilde{Q}_i^\pi(H_i, A_i) = \mathbb{E}[R \mid H_i, A_i]$.

By the definition of conditional expectation and the tower property,

$$\mathbb{E}[R - U \mid H_i] = 0, \qquad \mathbb{E}[U \mid x] = \mu_x. \tag{18}$$

Since U is a function of H_i, the two terms in $R - \mu_x = (R - U) + (U - \mu_x)$ are orthogonal under the conditional distribution given x:

$$\begin{aligned} \mathbb{E}\big[(R - U)(U - \mu_x) \mid x\big] &= \mathbb{E}\big[(U - \mu_x)\mathbb{E}[R - U \mid H_i] \mid x\big] \\ &= 0. \end{aligned} \tag{19}$$

Expanding the square and using the tower property therefore gives

$$\begin{aligned} \mathrm{Var}(R \mid x) &= \mathbb{E}\big[(R - U)^2 \mid x\big] + \mathbb{E}\big[(U - \mu_x)^2 \mid x\big] \\ &= \mathbb{E}_{H_i|x}\big[\mathrm{Var}(R \mid H_i)\big] + \mathrm{Var}_{H_i|x}\big[\widetilde{V}_i^\pi(H_i)\big]. \end{aligned} \tag{20}$$

This proves the first decomposition. In particular,

$$\mathcal{V}_i^{\mathrm{traj}} - \mathcal{V}_i^{\mathrm{SEPO}} = \mathrm{Var}_{H_i|x}\big[\widetilde{V}_i^\pi(H_i)\big] \geq 0. \tag{21}$$

Thus $\mathcal{V}_i^{\mathrm{SEPO}} \leq \mathcal{V}_i^{\mathrm{traj}}$, with strict inequality if and only if $\widetilde{V}_i^\pi(H_i)$ is not almost surely constant under the distribution of $H_i \mid x$.

To prove the second decomposition, apply the tower property once more:

$$\mathbb{E}[W \mid H_i] = \mathbb{E}\big[\mathbb{E}[R \mid H_i, A_i] \mid H_i\big] = U, \qquad \mathbb{E}[R - W \mid H_i, A_i] = 0. \tag{22}$$

Now decompose $R - U = (R - W) + (W - U)$. Because $W - U$ is a function of (H_i, A_i), its conditional cross term with $R - W$ vanishes:

$$\begin{aligned} \mathbb{E}\big[(R - W)(W - U) \mid H_i\big] &= \mathbb{E}\big[(W - U)\mathbb{E}[R - W \mid H_i, A_i] \mid H_i\big] \\ &= 0. \end{aligned} \tag{23}$$

Consequently, almost surely in H_i,

$$\begin{aligned} \mathrm{Var}(R \mid H_i) &= \mathbb{E}\big[(R - W)^2 \mid H_i\big] + \mathbb{E}\big[(W - U)^2 \mid H_i\big] \\ &= \mathbb{E}_{A_i|H_i}\big[\mathrm{Var}(R \mid H_i, A_i)\big] + \mathrm{Var}_{A_i|H_i}\big[\widetilde{Q}_i^{\pi}(H_i, A_i)\big]. \end{aligned} \tag{24}$$

This is the second decomposition: the conditional return variance separates into variation in the conditional action values and the remaining variance of subsequent rollouts given the history and current action. For a group sharing a fixed history $H_i = h_i$, the population return variance within that group is $\mathrm{Var}(R \mid H_i = h_i)$, provided its actions and continuations follow the same conditional rollout distribution. Averaging over shared histories drawn from the same marginal distribution of $H_i \mid x$ gives exactly $\mathcal{V}_i^{\mathrm{SEPO}}$. Hence prefix sharing removes the between-history component from the expected within-group return variance, while retaining the conditional action-value component in the second decomposition. □

B More Related Work

Temporal credit assignment and process supervision. RUDDER (Arjona-Medina et al., 2019) redistributes delayed rewards through learned return decomposition, while Hindsight Credit Assignment (Harutyunyan et al., 2019) evaluates past actions using their likelihood of producing observed outcomes. Process supervision provides another source of fine-grained feedback: *Let's Verify Step by Step* (Lightman et al., 2024) trains reward models with human-labeled reasoning steps, and Math-Shepherd (Wang et al., 2024) automatically constructs process supervision for verification and RL. *Rewarding Progress* (Setlur et al., 2025) further develops process advantage verifiers that measure improvements in future success probability under a prover policy. These approaches motivate distinguishing a decision's contribution from the quality of its resulting state. SEPO obtains local credit from comparisons of sampled task returns under shared histories, without requiring an additional learned return-decomposition or process reward model.

Structured rollouts and turn-level policy optimization. VinePPO (Kazemnejad et al., 2024) estimates intermediate values through Monte Carlo continuations from reasoning prefixes. TreeRL (Hou et al., 2025) uses on-policy tree search to derive intermediate supervision from outcomes, while TreeAdv (Cao et al., 2026) redistributes complete-rollout advantages to internal tree segments. For interactive agents, AT^{2}PO (Zong et al., 2026) combines turn-level tree expansion and credit estimation with turn-level importance weighting and clipping. These methods obtain finer-grained credit through structured rollout comparisons. SEPO constructs groups around selected turns or chunks and assigns their trajectory-level advantages to the corresponding decisions.

Cooperative credit assignment and sequential optimization. COMA (Foerster et al., 2018) uses a centralized critic and a counterfactual baseline that varies one agent's action while fixing the others. QMIX (Rashid et al., 2020) represents joint action values through a monotonic combination of individual values, and SeCA (Zang et al., 2023) assigns credit through sequential advantage functions with an adaptive evaluation order. For policy optimization, Kuba et al. (2021) derive a multi-agent advantage decomposition and sequential update scheme underlying HATRPO and HAPPO. CORY (Ma et al., 2024) applies sequential cooperation to LLM fine-tuning through separate pioneer and observer policies, with the observer conditioning on the pioneer's response and periodic role exchange during training. These works provide precedents for cooperative attribution and ordered optimization. SEPO assigns agent roles to the temporally ordered turns or chunks of an interaction and uses local best-response reasoning to motivate history-conditioned rollout groups for critic-free credit assignment.

Algorithm 1 SEPO

Require: Initial policy π_θ, task distribution $p(X)$, group size N, maximum turns T, chunk number K ($1 \leq K \leq T$), exploitation temperature τ_{exploit}, exploration temperature τ_{explore}

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$; $\mathcal{D} \leftarrow \varnothing$; sample task $x \sim p(X)$
  for $i = 1, \ldots, K$ **do**
    Initialize a group $\mathcal{G}_i$ of N rollouts for x with identical environments, using $\pi_{\theta_{\text{old}}}$
    for $t = 1, \ldots, T$ while at least one rollout is active **do**
      $c_t \leftarrow 1 + \lfloor (t-1)K/T \rfloor$ ▷ Chunk index
      if $c_t < i$ **then** ▷ Prefix
        Extend one shared prefix at τ_{exploit}; copy its history and environment state to all rollouts
      else if $c_t = i$ **then** ▷ Update
        Independently sample and execute actions for active rollouts at τ_{explore}
      else ▷ Suffix
        Independently sample and execute actions for active rollouts at τ_{exploit}
      end if
    end for
    Compute full-trajectory returns and the base method's advantages A from $\mathcal{G}_i$
    Retain only update-turn training samples from $\mathcal{G}_i$ and append them to $\mathcal{D}$
  end for
  Update π_θ on $\mathcal{D}$ using the objective in Equation (10)
end for

Concurrent work on turn-level policy optimization. Concurrent with our work, SeeUPO (Hu et al., 2026) and RTPO (Li et al., 2026) adopt related decompositions of multi-turn interactions: SeeUPO models individual turns as virtual agents, while RTPO formulates a turn-boundary MDP. SeeUPO primarily changes how group-relative advantages guide reverse-order policy updates. In its GRAE implementation, positive importance weights preserve the sign of a shared trajectory-level advantage across turns. SEPO instead constructs local rollout groups to estimate credit for selected chunks. When evaluating a reference trajectory at turn-level granularity, these local comparisons can yield advantages of opposite signs for different turns, allowing decisions to be reinforced or penalized according to their estimated value relative to local alternatives. RTPO also compares sibling rollouts to estimate advantages, but couples this estimation with reverse-order updates and refreshed downstream continuations; SEPO constructs its chunk-specific groups under a common pre-update policy without requiring this update schedule. The related formulations and empirical successes of these methods support the usefulness of the multi-agent perspective for multi-turn learning. Their different emphases also suggest potential complementarity: future work could combine SEPO's chunk-level advantage estimation with SeeUPO-style reverse-order updates within each chunk, accounting for policy changes during sequential updates to improve credit assignment.

C Pseudo Code

Algorithm 1 presents the pseudo code for SEPO, highlighting its modifications to the rollout procedure relative to the base method. Other components, such as advantage estimation and DAPO's dynamic sampling, follow the base method and are omitted for brevity.

D Experiment Details

D.1 Training Details

Hyperparameters. To facilitate reproducibility, our implementation is built on the verl-agent framework[1] (Feng et al., 2026). For each environment, we retain all hyperparameter settings of the corresponding base algorithm when integrating SEPO. The only additional hyperparameter introduced by SEPO is the chunk number K. The exploration and exploitation temperatures follow the training and validation settings of the base algorithm, respectively.

[1] https://github.com/langfengQ/verl-agent

Computing details. All experiments are conducted on NVIDIA A800 GPUs. We train Qwen2.5-1.5B with two to four GPUs: two GPUs suffice to meet the memory requirements, while four GPUs can accelerate training. Qwen2.5-3B and Qwen3-4B are trained with four GPUs, and Qwen2.5-7B with four to eight GPUs. For Qwen2.5-7B, reducing the micro-batch size enables training on four GPUs within the available memory budget. To illustrate the computational overhead, on WebShop with four A800 GPUs, standard GRPO takes approximately 179 seconds per training iteration, whereas SEPO with $K = 4$ takes approximately 394 seconds. This additional cost arises from the four rollout groups required to optimize the four chunks. Possible ways to reduce the overall overhead include applying SEPO only during later training stages, as in our warm-up variant, increasing rollout parallelism when resources permit, and optimizing the implementation of prefix sharing. For example, shared prefixes can first be generated in an auxiliary batch with a group size of one and then replicated across the full rollout group, avoiding redundant prefix generation. Similar to DAPO, our work prioritizes performance at convergence over wall-clock efficiency. Accordingly, SEPO is particularly suited to settings where sufficient computational resources are available to accommodate the additional rollout cost.

Prompts. Our work focuses on algorithmic improvements, and all prompts are kept identical to those used in Feng et al. (2026).

D.2 Case Study Details

This section describes the implementation details of the case study in Section 5.3. To analyze the advantages assigned to individual turns by GRPO and SEPO, we first sample a reference trajectory, which is the trajectory presented in the case study. To obtain its GRPO advantage, we sample a group of five trajectories and replace the first trajectory with the reference trajectory. We then compute the group-relative advantage of the reference trajectory, which GRPO assigns to all of its turns. To obtain the SEPO advantage for each turn, we perform counterfactual sampling. Specifically, for a target turn i, we construct a group of five trajectories whose prefixes, comprising turns $1, \ldots, i-1$, are fixed to the corresponding content of the reference trajectory. At turn i, the first trajectory reuses the reference response, while the remaining four trajectories sample new responses using the exploration temperature. All five trajectories are then completed using the exploitation temperature for subsequent turns. Based on the resulting trajectory returns, we compute the group-relative advantage of the first trajectory and assign it to turn i of the reference trajectory. Repeating this procedure for every turn yields the corresponding SEPO advantages. This procedure closely follows the rollout design of SEPO, with each target turn treated as the update segment.

D.3 Performance on Vision-Language Models

Our previous experiments focus on language-only models. To further assess the generality of SEPO, we extend our evaluation to the vision-language model (VLM) Qwen2.5-VL-3B-Instruct on Sokoban [6×6]. As shown in Table 3, integrating SEPO with GiGPO also improves performance in this setting, consistent with our findings on language-only models.

Table 3: Results on Sokoban [6×6].

Method	Sokoban [6×6]
GiGPO	$0.77_{\pm 0.03}$
GiGPO+SEPO	$\mathbf{0.82}_{\pm 0.03}$

D.4 Comparison with AT^{2}PO

AT^{2}PO (Zong et al., 2026) is another strong recent baseline that employs tree-based rollouts. We therefore include an additional comparison in Table 4. We set the number of initial nodes to 4, the number of branching nodes to 1, and the number of branching rounds to 2. Under our accounting of update data, this yields an effective batch size of $4 + 2 \times (1/2) = 5$, matching the update data budget of the other methods. We further combine SEPO with ATPO, a variant of AT^{2}PO without tree rollouts, to evaluate the effectiveness of our credit assignment mechanism. AT^{2}PO is a strong baseline, achieving higher average performance than GiGPO+SEPO. Nevertheless, ATPO+SEPO outperforms AT^{2}PO on five of the seven benchmarks, improving the average score from 43.7 to 45.3. These results support the effectiveness of the credit assignment provided by SEPO and its applicability across different group-based methods. Overall, the two approaches offer different strengths:

Table 4: Comparison with AT^{2}PO on QA benchmarks.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ	TriviaQA	PopQA	HotpotQA	2Wiki	MuSiQue	Bamboogle	
Qwen3-4B								
AT^{2}PO	45.4	**61.6**	44.7	37.5	37.8	13.1	**65.7**	43.7
ATPO+SEPO	**46.4**	61.3	**47.1**	**40.7**	**41.0**	**16.7**	64.1	**45.3**

Table 5: Comparison with larger-batch baselines. Subscripts indicate standard deviations.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Qwen2.5-1.5B-Instruct									
GRPO (Larger Batch)	$0.86_{\pm 0.04}$	$0.71_{\pm 0.10}$	$\mathbf{0.81}_{\pm 0.05}$	$0.74_{\pm 0.05}$	$0.81_{\pm 0.09}$	$\mathbf{0.65}_{\pm 0.18}$	$0.77_{\pm 0.04}$	$\mathbf{0.83}_{\pm 0.01}$	$\mathbf{0.71}_{\pm 0.04}$
GRPO (Larger Batch Same Data)	$0.72_{\pm 0.07}$	$0.58_{\pm 0.11}$	$0.71_{\pm 0.16}$	$0.66_{\pm 0.07}$	$0.75_{\pm 0.03}$	$0.59_{\pm 0.04}$	$0.68_{\pm 0.01}$	$0.78_{\pm 0.03}$	$0.64_{\pm 0.04}$
GRPO+SEPO	$\mathbf{0.89}_{\pm 0.00}$	$0.77_{\pm 0.21}$	$0.68_{\pm 0.16}$	$0.81_{\pm 0.08}$	$0.73_{\pm 0.04}$	$0.62_{\pm 0.14}$	$0.75_{\pm 0.04}$	$0.81_{\pm 0.03}$	$\mathbf{0.71}_{\pm 0.03}$
GRPO+SEPO (Warm-up)	$0.87_{\pm 0.05}$	$\mathbf{0.82}_{\pm 0.07}$	$0.80_{\pm 0.04}$	$\mathbf{0.82}_{\pm 0.14}$	$\mathbf{0.86}_{\pm 0.07}$	$0.61_{\pm 0.17}$	$\mathbf{0.80}_{\pm 0.05}$	$0.82_{\pm 0.02}$	$0.68_{\pm 0.03}$
DAPO (Larger Batch)	$\mathbf{0.99}_{\pm 0.01}$	$0.88_{\pm 0.04}$	$0.86_{\pm 0.04}$	$\mathbf{0.97}_{\pm 0.05}$	$0.87_{\pm 0.07}$	$0.74_{\pm 0.11}$	$0.88_{\pm 0.04}$	$0.89_{\pm 0.01}$	$0.80_{\pm 0.01}$
DAPO+SEPO	$0.98_{\pm 0.02}$	$\mathbf{0.94}_{\pm 0.05}$	$\mathbf{0.93}_{\pm 0.03}$	$\mathbf{0.97}_{\pm 0.05}$	$\mathbf{0.93}_{\pm 0.01}$	$\mathbf{0.84}_{\pm 0.06}$	$\mathbf{0.93}_{\pm 0.01}$	$\mathbf{0.93}_{\pm 0.02}$	$\mathbf{0.84}_{\pm 0.04}$

AT^{2}PO delivers competitive performance under limited rollout budgets, while SEPO has the potential to achieve higher final performance when larger rollout budgets are available.

D.5 Comparison Under Same Rollout Count

Our previous comparisons primarily focused on the improvements in policy convergence performance provided by SEPO, using the same amount of data for policy updates. In this section, we additionally compare against GRPO/DAPO (Larger Batch), which use four times the default rollout batch size. Their mini-batch sizes are also increased by a factor of four to preserve the number of gradient update steps. These baselines therefore match the total rollout data volume of the corresponding SEPO variants with $K = 4$, while using four times as much data for policy updates. We further include GRPO (Larger Batch Same Data), which matches the rollout count of SEPO and applies the same rule for selecting update turns, thereby also matching the amount of data used for policy updates. As shown in Table 5, GRPO (Larger Batch) achieves slightly better aggregate performance than GRPO+SEPO, although the gains remain modest despite the substantially larger amount of update data. On ALFWorld, the warm-up variant outperforms GRPO (Larger Batch) while using only 62.5% of its rollout budget and 25% of its update data. On WebShop, the warm-up variant performs slightly worse, with relatively small gaps in both score and success rate. These results suggest that SEPO can remain competitive under limited rollout budgets and, with a suitable training schedule, achieve better performance with less data. For DAPO, the Larger Batch baseline remains below DAPO+SEPO on the aggregate metrics, despite using the same rollout budget and more update data. This supports the effectiveness of SEPO beyond increasing the amount of training data. Furthermore, GRPO (Larger Batch Same Data) performs similarly to standard GRPO overall and falls behind GRPO+SEPO, despite matching both its rollout count and update data volume. This comparison indicates that the gains from SEPO cannot be explained solely by collecting more trajectories and updating on only a subset of their turns. Instead, the results support the contribution of its rollout design to more precise credit assignment. Nevertheless, we consider SEPO particularly suitable for settings with ample rollout budgets or for the later stages of RL training, where more precise credit assignment can help improve final policy performance.

D.6 Standard Deviation of SEPO

To further assess the stability of SEPO, we report the standard deviations of its main experimental results across three random seeds in Table 6. The standard deviations are relatively small in most settings, suggesting consistent performance across random seeds. Comparing these results with the corresponding mean performance reported in the main text also suggests that stronger performance is generally accompanied by lower variability.

Table 6: Standard deviations across three random seeds for SEPO combined with different base algorithms on ALFWorld and WebShop.

Method	ALFWorld							WebShop	
	Pick	Look	Clean	Heat	Cool	Pick2	All	Score	Succ.
Qwen2.5-1.5B-Instruct									
GRPO+SEPO	0.00	0.21	0.16	0.08	0.04	0.14	0.04	0.03	0.03
GiGPO+SEPO	0.03	0.17	0.20	0.00	0.02	0.14	0.03	0.02	0.03
DAPO+SEPO	0.02	0.05	0.03	0.05	0.01	0.06	0.01	0.02	0.04
Qwen3-4B									
GRPO+SEPO	0.08	0.23	0.03	0.11	0.10	0.04	0.02	0.03	0.03
GiGPO+SEPO	0.06	0.06	0.00	0.11	0.01	0.16	0.01	0.00	0.02
DAPO+SEPO	0.00	0.00	0.03	0.04	0.00	0.00	0.00	0.02	0.03
Qwen2.5-7B-Instruct									
GRPO+SEPO	0.02	0.14	0.03	0.04	0.12	0.09	0.01	0.05	0.03
GiGPO+SEPO	0.05	0.04	0.02	0.04	0.00	0.06	0.01	0.01	0.04
DAPO+SEPO	0.02	0.06	0.03	0.03	0.00	0.04	0.00	0.01	0.02

D.7 Reasoning Behaviors

In this section, we present complete multi-turn rollout trajectories of DAPO+SEPO across the three environments to illustrate the model's reasoning behavior.

D.7.1 ALFWorld

Environment (Step 1)

-= Welcome to TextWorld, ALFRED! =-
You are in the middle of a room. Looking quickly around you, you see a cabinet 10, a cabinet 9, a cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a cabinet 1, a coffeemachine 1, a countertop 1, a diningtable 1, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.
Your task is to: put a clean fork in countertop.

Agent (Step 1)

<think> To complete the task of putting a clean fork on the countertop, I first need to locate a fork. Forks are commonly found in cabinets, drawers, or on dining tables. Since the task does not specify a particular location, I will start with the most logical places to look, which are the cabinets and the dining table. I will go to the dining table first to check if there is a fork there. If not, I will proceed to the cabinets. </think>
<action>go to diningtable 1</action>

Environment (Step 2)

You arrive at diningtable 1. On the diningtable 1, you see a bowl 2, a bowl 1, a butterknife 1, a fork 4, a fork 3, a fork 2, a fork 1, a knife 1, a pan 1, a pen 1, a pencil 2, a pencil 1, a peppershaker 1, a plate 1, a pot 1, a potato 1, a soapbottle 1, a spatula 1, a spoon 3, a spoon 2, a spoon 1, a tomato 1, and a winebottle 1.

Agent (Step 2)

<think> I need to find a clean fork and place it on the countertop. The fork is currently on the diningtable, so I should examine the diningtable to find it. The fork is one of the

Agent (Step 2, continued)

items listed, so I can take it directly from the diningtable. I'll start by taking the first fork available, which is fork 1. </think>
<action>take fork 1 from diningtable 1</action>

Environment (Step 3)

You pick up the fork 1 from the diningtable 1.

Agent (Step 3)

<think> I have just picked up fork 1 from the diningtable 1. The next step is to clean the fork before placing it on the countertop. The most appropriate action from the list provided would be to go to the sinkbasin 1 to clean the fork. After cleaning, I can place the fork on the countertop. </think>
<action>go to sinkbasin 1</action>

Environment (Step 4)

You arrive at sinkbasin 1. On the sinkbasin 1, you see nothing.

Agent (Step 4)

<think> I have fork 1 in my inventory and need to clean it. The sinkbasin 1 is available, which seems to be the most appropriate place to clean the fork. The action 'clean fork 1 with sinkbasin 1' is available, which would allow me to clean the fork. After cleaning, I can put the fork on the countertop. </think>
<action>clean fork 1 with sinkbasin 1</action>

Environment (Step 5)

You clean the fork 1 using the sinkbasin 1.

Agent (Step 5)

<think> I have just cleaned the fork 1 using sinkbasin 1. The next step is to put the fork on the countertop. The action 'go to countertop 1' is available, which would allow me to place the fork there. Therefore, I will go to the countertop to put the fork there. </think>
<action>go to countertop 1</action>

Environment (Step 6)

You arrive at countertop 1. On the countertop 1, you see a bread 1, a knife 2, a lettuce 1, and a peppershaker 3.

Agent (Step 6)

<think> I have just cleaned fork 1 and need to put it on the countertop. The action 'move fork 1 to countertop 1' is available, which would accomplish this task. Since the fork is likely still in my inventory or the last place I interacted with it (sinkbasin 1), I should first examine fork 1 to confirm its location before performing the action. </think>

Agent (Step 6, continued)

<action>examine fork 1</action>

Environment (Step 7)

This is a clean fork 1.

Agent (Step 7)

<think> I have a clean fork 1 that needs to be put on the countertop 1. The action 'move fork 1 to countertop 1' is available, which would accomplish this task. Since the fork is already clean and I am currently at countertop 1, I can directly place the fork there. </think>
<action>move fork 1 to countertop 1</action>

D.7.2 WEBSHOP

Environment (Step 1)

Your task is to: Find me machine wash men's dress shirts with cotton spandex, classic fit, short sleeve with color: patina green, and size: x-large, and price lower than 60.00 dollars. 'Search'.

Agent (Step 1)

<think> The task is to find machine wash men's dress shirts with specific attributes such as cotton spandex, classic fit, short sleeve, color, and size. The search should also include a price lower than $60.00.
From the current observation, the action to be performed is to use the 'search[<your query>]' option. I need to construct a query that best matches the required attributes. </think>
<action>search[men's shirt cotton spandex classic fit short sleeve patina green X-large price < 60.00]</action>

Environment (Step 2)

'Back to Search' [SEP] 'Page 1 (Total results: 50)' [SEP] 'Next >' [SEP] 'B07HRFSNL4' [SEP] 'Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt' [SEP] '$16.05 to $40.98' [SEP] 'B09QQP3356' [SEP] 'HAUKLIE Men's Sports Waffle Ribbed Polo Shirts Summer Short Sleeve Cotton Muscle Quarter-Zip Henley T-Shirt Tunics Tops' [SEP] '$10.99' [SEP] 'B09R9YCM6R' [SEP] 'XXBR Summer T-shirts for Mens, Soldier Short Sleeve 3D Street Vintage Printed Shirt Slim Fit Muscle Casual Tee Tops' [SEP] '$8.98 to $11.99' [SEP] 'B085WQKRRJ' [SEP] 'CLANMILUMS Men's Classic Comfort Soft Regular Fit Short Sleeve Henley T-Shirt Tee' [SEP] '$17.99 to $19.99' [SEP] 'B09NNMV9LN' [SEP] 'JSPOYOU Mens Short Sleeve Crewneck 3D Graphic Tunic Shirts Big & Tall Tie Dye Summer Top Basic Designed Classic Cotton Shirt' [SEP] '$3.99 to $10.99' [SEP] 'B09KLQLLT2' [SEP] 'Long Sleeve Superhero T Shirt Tank Top Mens Compression Shirt Men Workout Fitness Gym Shirt' [SEP] '$19.99' [SEP] 'B09HGFV91W' [SEP] 'ZHDD Tops for Mens, Men's Knight Punk Vintage Style Long Sleeve Asymmetric Hem Cowl Neck Teen Boys Gothic Pullover' [SEP] '$4.98 to $7.99' [SEP] 'B07N7TDKXQ' [SEP] 'SOCKS'NBULK Mens Cotton Crew Neck Short Sleeve T-Shirts Mix Colors Bulk' [SEP] '$38.79 to $172.8' [SEP] 'B09S3BN15C' [SEP] 'Mens Linen Shirt,Men's Striped Shirts Casual Short Sleeve Button Down Shirts Regular Fit Hawaiian Shirts Beach Tees Tops' [SEP] '$3.78 to $11.38' [SEP] 'B07F2G93BJ' [SEP] 'Amazon Essentials Men's Slim-fit Long-Sleeve Solid Pocket Oxford Shirt' [SEP] '$18.5'.

Agent (Step 2)

<think> From the current observation, the most relevant product to the task is the "Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt" (B07HRFSNL4) because it matches the majority of the criteria: men's shirt, cotton spandex, classic fit, short sleeve, and has a price range of $16.05 to $40.98. However, the color is not specified in the observation, and the color requirement is patina green. Since the color is not mentioned, it's best to click on the product to see if it matches the color requirement or to find a similar product.
</think>
<action>click[b07hrfsnl4]</action>

Environment (Step 3)

'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] 'x-small' [SEP] 'small' [SEP] 'medium' [SEP] 'large' [SEP] 'large tall' [SEP] 'x-large' [SEP] 'x-large tall' [SEP] 'xx-large' [SEP] 'xx-large tall' [SEP] '3x-large tall' [SEP] '4x-large tall' [SEP] '5x-large tall' [SEP] '1x' [SEP] '2x' [SEP] '3x' [SEP] '4x' [SEP] '5x' [SEP] '6x' [SEP] 'color' [SEP] 'navy' [SEP] 'noon blue' [SEP] 'grey heather' [SEP] 'nautica red' [SEP] 'bright white' [SEP] 'true black' [SEP] 'pale coral' [SEP] 'cradle pink' [SEP] 'bright aqua' [SEP] 'mint spring' [SEP] 'monaco blue' [SEP] 'charcoal heather (dark)' [SEP] 'barolo' [SEP] 'tidal green' [SEP] 'charcoal heather (light)' [SEP] 'estate blue' [SEP] 'patina green' [SEP] 'black' [SEP] 'bright aqua blue' [SEP] 'bright cobalt' [SEP] 'coral sands' [SEP] 'deep anchor heather' [SEP] 'hawaiian ocean' [SEP] 'melon berry' [SEP] 'pine forest green' [SEP] 'shipwreck burgundy' [SEP] 'white' [SEP] 'zest' [SEP] 'deep atlantic' [SEP] 'delft' [SEP] 'pale coral red' [SEP] 'Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt' [SEP] 'Price: $16.05 to $40.98' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'.

Agent (Step 3)

<think> The goal is to find machine wash men's dress shirts with cotton spandex, classic fit, short sleeve with the color patina green and size x-large, and price lower than $60.00. From the current observation, I am currently viewing a list of products with different sizes and colors. The product "Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt" has the color patina green, but it does not have the size x-large and the price range is from $16.05 to $40.98, which is not within the required range.
Given the current observation, the best action is to click on the color "patina green" to see if there are any products in the desired color and size. </think>
<action>click[patina green]</action>

Environment (Step 4)

'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] 'x-small' [SEP] 'small' [SEP] 'medium' [SEP] 'large' [SEP] 'large tall' [SEP] 'x-large' [SEP] 'x-large tall' [SEP] 'xx-large' [SEP] 'xx-large tall' [SEP] '3x-large tall' [SEP] '4x-large tall' [SEP] '5x-large tall' [SEP] '1x' [SEP] '2x' [SEP] '3x' [SEP] '4x' [SEP] '5x' [SEP] '6x' [SEP] 'color' [SEP] 'navy' [SEP] 'noon blue' [SEP] 'grey heather' [SEP] 'nautica red' [SEP] 'bright white' [SEP] 'true black' [SEP] 'pale coral' [SEP] 'cradle pink' [SEP] 'bright aqua' [SEP] 'mint spring' [SEP] 'monaco blue' [SEP] 'charcoal heather (dark)' [SEP] 'barolo' [SEP] 'tidal green' [SEP] 'charcoal heather (light)' [SEP] 'estate blue' [SEP] 'patina green' [SEP] 'black' [SEP] 'bright aqua blue' [SEP] 'bright cobalt' [SEP] 'coral sands' [SEP] 'deep anchor heather' [SEP] 'hawaiian ocean' [SEP] 'melon berry' [SEP] 'pine forest green' [SEP] 'shipwreck burgundy' [SEP] 'white' [SEP] 'zest' [SEP] 'deep atlantic' [SEP] 'delft' [SEP] 'pale coral red' [SEP] 'Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt' [SEP] 'Price: $16.05 to $40.98' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'.

Agent (Step 4)

<think> The goal is to find machine wash men's dress shirts with cotton spandex, classic fit, short sleeve, in patina green color, size X-large, and price lower than $60.00.
From the current observation, I have already clicked on "patina green" and now I am on the size options. The next logical step is to click on "x-large" to see if there are any X-large sizes available in the desired color. </think>
<action>click[x-large]</action>

Environment (Step 5)

'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] 'x-small' [SEP] 'small' [SEP] 'medium' [SEP] 'large' [SEP] 'large tall' [SEP] 'x-large' [SEP] 'x-large tall' [SEP] 'xx-large' [SEP] 'xx-large tall' [SEP] '3x-large tall' [SEP] '4x-large tall' [SEP] '5x-large tall' [SEP] '1x' [SEP] '2x' [SEP] '3x' [SEP] '4x' [SEP] '5x' [SEP] '6x' [SEP] 'color' [SEP] 'navy' [SEP] 'noon blue' [SEP] 'grey heather' [SEP] 'nautica red' [SEP] 'bright white' [SEP] 'true black' [SEP] 'pale coral' [SEP] 'cradle pink' [SEP] 'bright aqua' [SEP] 'mint spring' [SEP] 'monaco blue' [SEP] 'charcoal heather (dark)' [SEP] 'barolo' [SEP] 'tidal green' [SEP] 'charcoal heather (light)' [SEP] 'estate blue' [SEP] 'patina green' [SEP] 'black' [SEP] 'bright aqua blue' [SEP] 'bright cobalt' [SEP] 'coral sands' [SEP] 'deep anchor heather' [SEP] 'hawaiian ocean' [SEP] 'melon berry' [SEP] 'pine forest green' [SEP] 'shipwreck burgundy' [SEP] 'white' [SEP] 'zest' [SEP] 'deep atlantic' [SEP] 'delft' [SEP] 'pale coral red' [SEP] 'Nautica Men's Solid Crew Neck Short-Sleeve Pocket T-Shirt' [SEP] 'Price: $16.05 to $40.98' [SEP] 'Rating: N.A.' [SEP] 'Description' [SEP] 'Features' [SEP] 'Reviews' [SEP] 'Buy Now'.

Agent (Step 5)

<think> The goal is to find machine wash men's dress shirts with cotton spandex, classic fit, short sleeve with the color patina green and size x-large, and price lower than $60.00.
From the current observation, the previous actions have already clicked on "patina green" and "x-large". However, the search results do not match the required criteria. The next logical step is to click on "Buy Now" to see if there are any products that match the current criteria.
If "Buy Now" does not show any products, I will click "Back to Search" to refine the search criteria. </think>
<action>click[buy now]</action>

D.7.3 SEARCH

Environment (Step 1)

Your question: who was the captain of the mayflower when it took the pilgrims to the new world?

Agent (Step 1)

<think> I do not have knowledge about the specific captain of the Mayflower when the pilgrims arrived in the New World. I need to conduct a search to find the correct answer. </think>
<search>who was the captain of the mayflower when it took the pilgrims to the new world</search>

Environment (Step 2)

Doc 1: Mayflower
Mayflower The Mayflower was an English ship that transported the first English Puritans, known today as the Pilgrims, from Plymouth, England, to the New World in 1620. There were 102 passengers, and the crew is estimated to have been about 30, but the exact number is unknown. The ship has become a cultural icon in the history of the United States. The Pilgrims signed the Mayflower Compact prior to leaving the ship and establishing Plymouth Colony, a document which established a rudimentary form of democracy with each member contributing to the welfare of the community. There was a second ship
Doc 2: Mayflower
to build a replica of the ship at Harwich, England. Mayflower The Mayflower was an English ship that transported the first English Puritans, known today as the Pilgrims, from Plymouth, England, to the New World in 1620. There were 102 passengers, and the crew is estimated to have been about 30, but the exact number is unknown. The ship has become a cultural icon in the history of the United States. The Pilgrims signed the Mayflower Compact prior to leaving the ship and establishing Plymouth Colony, a document which established a rudimentary form of democracy with each member contributing to
Doc 3: "Plymouth Colony"
had been designated by the Merchant Adventurers to act as shipboard governor during the trans-Atlantic trip; and Stephen Hopkins, a veteran of a failed colonial venture that may have inspired Shakespeare's ""The Tempest"". The group who later became the Leiden Leaders after the merging of ships included John Carver, William Bradford, Edward Winslow, William Brewster, and Isaac Allerton. The ""Mayflower"" departed Plymouth, England on September 6, 1620 with 102 passengers and about 30 crew members in the small, 106 foot-long ship. The seas were not severe during the first month in the Atlantic but, by the second month, the ship

Agent (Step 2)

<think> From the information provided, we know that the Mayflower transported the Pilgrims to the New World in 1620. To determine the captain of the Mayflower during this voyage, we need to search the internet for the specific captain of the Mayflower during the 1620 voyage. </think>
<search>captain of mayflower in 1620</search>

Environment (Step 3)

Doc 1: Mayflower
Mayflower The Mayflower was an English ship that transported the first English Puritans, known today as the Pilgrims, from Plymouth, England, to the New World in 1620. There were 102 passengers, and the crew is estimated to have been about 30, but the exact number is unknown. The ship has become a cultural icon in the history of the United States. The Pilgrims signed the Mayflower Compact prior to leaving the ship and establishing Plymouth Colony, a document which established a rudimentary form of democracy with each member contributing to the welfare of the community. There was a second ship
Doc 2: "Isaac Allerton"
Isaac Allerton Isaac Allerton Sr. (c.1586 – 1658/9), and his family, were passengers in 1620 on the historic voyage of the ship ""Mayflower"". Allerton was a signatory to the Mayflower Compact. In Plymouth Colony he was active in colony governmental affairs and business and later in trans-Atlantic trading. Problems with the latter regarding colony expenditures caused him to be censured by the colony government and ousted from the colony. He later became a well-to-do businessman elsewhere and in his later years resided in Connecticut. Based on a deposition given in 1639, Allerton was born in England about 1586–88, although clues
Doc 3: "Christopher Jones (Mayflower captain)"
Christopher Jones (Mayflower captain) Capt. Christopher Jones Jr. (c. 1570 – about 5 March 1622) was the Captain of the 1620 voyage of the Pilgrim ship ""Mayflower"". Christopher Jones is believed to have been born in Harwich, Essex about 1570, although baptismal records are blank for his parish church for the period of time between April 1565

Environment (Step 3, continued)

and June 1571. He was the son of Christopher Jones, Sr. and his wife Sybil. The senior Jones was also a mariner and ship owner who died in 1578, leaving to his young son, bearing his name, his interest in the ship ""Marie

Agent (Step 3)

<think> From the information provided, the captain of the Mayflower during the 1620 voyage that transported the Pilgrims to the New World was Christopher Jones. The query "who was the captain of the mayflower when it took the pilgrims to the new world" can be answered with Christopher Jones. </think>
<answer>Christopher Jones</answer>