DCGPO: DISTRIBUTIONAL CRITIC GUIDED POLICY OPTIMIZATION FOR MULTI-TURN LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Agentic Reinforcement Learning (Agentic RL) for large language model agents faces challenges in credit assignment and exploration efficiency. Stochastic generation does not necessarily yield effective environment exploration: repeated trajectories and limited state coverage can reduce the effective use of interaction budgets. We propose Distributional Critic Guided Policy Optimization (DCGPO), an actor–critic framework that combines turn-level distributional value learning with distribution-guided exploration. The critic supports turn-level advantage estimation and provides action-conditioned return means and dispersion to guide training-time sampling. By ranking policy-generated candidate turns before execution, DCGPO directs environment interactions toward promising behaviors while allowing exploration beyond mean-greedy selection. Our fixed-policy analysis shows that categorical critic learning preserves the form and statistical order of the scalar turn-level error bound for mean-induced advantage estimation. Experiments on embodied and web-based interactive tasks across multiple model scales demonstrate the effectiveness of DCGPO, while component ablations and visitation analyses examine its effects on task performance and exploration.

1 INTRODUCTION

Recent research on large language models (LLMs) has achieved remarkable success in static, single-turn reasoning tasks such as mathematical reasoning or code generation (OpenAI, 2024; Hui et al., 2024; Guo et al., 2025). However, real-world applications often involve far more complex and open-ended scenarios, where an agent must not only produce correct answers but also actively gather information and execute actions through multi-turn interactions with dynamic environments. Agentic Reinforcement Learning (Agentic RL) (Singh et al., 2025; Zhang et al., 2025a; Dong et al., 2026b) emerges as a direct response to this demand: by training LLMs as decision-making agents that interact with external environments, it turns passive task solving into active, experience-driven exploration (Wen et al., 2024; Dong et al., 2026a; Qian et al., 2026).

Recent advances in Agentic RL have primarily focused on two pivotal challenges: sparse reward and exploration efficiency. For the former, several approaches reformulate agent responses as macro-actions within turn-level MDPs to enable turn-wise credit assignment (Zhou et al., 2024; Li et al., 2026; Xi et al., 2026), or compute local advantage values through structural grouping and adaptive weighting (Feng et al., 2026; He et al., 2026; Peng et al., 2026). For the latter, recent studies leverage policy entropy to guide proactive branching of the sampling policy (Dong et al., 2026b) or use uncertainty-guided reweighting strategies to encourage active exploration (Wang et al., 2026). Despite these advances, efficient exploration remains a challenge in multi-turn LLM agents. Stochastic sampling and entropy-based mechanisms promote diversity in generation, but this diversity does not necessarily translate into effective exploration of the environment. Agents may repeatedly follow similar trajectories or revisit previously encountered states, spending limited interaction budgets on redundant experience. As we examine in Section 3, these limitations motivate exploration mechanisms that account for the downstream task consequences of generated actions.

To address these challenges, we propose Distributional Critic Guided Policy Optimization (DCGPO), which combines turn-level distributional value learning with Distributional-Guided Exploration. The distributional critic models the downstream returns of complete interaction turns and supports turn-level advantage estimation. Beyond value estimation, the predicted return distri-

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.

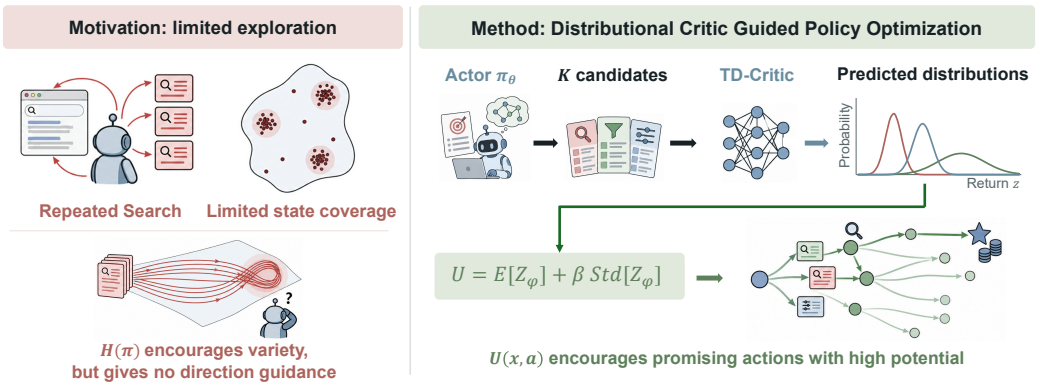


Figure 1: Comparison between PPO with entropy regularization and DCGPO. (Left) Increasing policy entropy diversifies local action generation but does not evaluate the downstream consequences of sampled actions, resulting in low state coverage and repeated trajectories in our preliminary study. (Right) DCGPO separates candidate generation from selection and ranks policy-generated turns by combining expected downstream return with predicted return dispersion.

butions provide action-conditioned means and dispersion for guiding training-time sampling. The policy generates candidate turns, and the exploration mechanism uses these statistics to select which candidate to execute, directing limited environment interactions toward promising behaviors while allowing exploration beyond mean-greedy selection. Our fixed-policy analysis further shows that categorical critic learning preserves the form and statistical order of the scalar turn-level bound for mean-induced advantage estimation.

We conduct experiments on ALFWorld and WebShop (Shridhar et al., 2021; Yao et al., 2022) across multiple model scales. The results demonstrate the effectiveness of DCGPO, while ablations and visitation analyses examine how distributional critic learning and dispersion-guided sampling contribute to task performance and exploration. Notably, some recent RL methods for LLMs (Liu et al., 2026; Zhou et al., 2026) also model return distributions, but they operate on token-level MDPs over single-turn RL with verifiable rewards (RLVR), using the distributions for value estimation and tail-aware advantages. In contrast, DCGPO focuses on multi-turn agent–environment interaction and uses action-conditioned return statistics to guide training-time sampling. Our contributions are summarized as follows:

- First, we identify and characterize an exploration bottleneck in LLM agents: in our controlled study, stochastic generation and the tested entropy regularization do not sufficiently prevent repeated trajectories or expand state coverage, limiting the effective use of environment interactions.
- Second, we propose DCGPO, which combines a turn-level distributional critic with mean–dispersion candidate selection. The critic supports advantage estimation, while its action-conditioned return statistics guide training-time sampling and the allocation of environment interactions.
- Third, we evaluate DCGPO across multiple models and interactive tasks and analyze the contributions of its components and their effects on state visitation. We further establish a fixed-policy guarantee preserving the form and statistical order of the scalar turn-level advantage-estimation bound under categorical critic learning.

2 Preliminaries

Turn-Level MDP Following recent studies (Zhou et al., 2024; Li et al., 2026), we model the entire dynamic process as a turn-level Markov Decision Process $M = \{\mathcal{S}, \mathcal{A}, \mathcal{P}, r\}$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ the action space, $\mathcal{P}$ the state transition function, and $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function. Throughout this work, we assume a deterministic environment, a setting common

in RLVR (Lambert et al., 2024), where the transition function $\mathcal{P}$ maps each state-action pair to a unique successor state, i.e., $\mathcal{P}(\cdot \mid s, a)$ is a point mass for every (s, a).

Further, the action $a_t \in \mathcal{A}$ and state $s_t \in \mathcal{S}$ are token sequences over vocabulary V. At each time step t, the agent's policy is governed by an LLM π_θ parameterized by θ. Conditioned on the interaction history $h_t = (s_0, a_0, \ldots, s_t)$ and task prompt s_p, the model autoregressively samples an action as $a_t^{1:L} \sim \pi_\theta(\cdot \mid h_t, s_p)$, where a_t^i denotes the i-th token of the output sequence a_t, and L is the length of a_t. For simplicity, we abbreviate (h_t, s_p) as $x_t \in \mathcal{X}$, where $\mathcal{X}$ is the observation space. The complete trajectory over the episode is denoted by $\tau = \{(x_t, a_t, r_t)\}_{t=0}^{T-1}$, where T is the length of this episode.

Agentic Reinforcement Learning Once the multi-turn interactions are formulated as turn-level MDPs $\mathcal{M}$, we can naturally apply RL methods to train the agent by maximizing the expected cumulative return $R(\tau)$ over the whole episode, i.e., $J(\theta) = \mathbb{E}_{\mathcal{M}, \tau \sim \pi_\theta}[R(\tau)]$. To get the optimal policy π^* in the turn-level MDP, a straightforward method is to use turn-level PPO (Li et al., 2026), where the critic estimates the value for each turn and the advantage is shared by all tokens within the same turn. For a turn $a_t = a_t^{1:L_t}$ of length L_t, we define the per-token importance ratio $\rho_t^i(\theta) := \frac{\pi_\theta(a_t^i | x_t, a_t^{<i})}{\pi_{\theta_{\text{old}}}(a_t^i | x_t, a_t^{<i})}$ for $i = 1, \ldots, L_t$, which measures the change in the policy probability of the i-th token in turn t. Accordingly, the turn-level advantage and the actor update objective are formulated as:

$$\hat{A}_t = \sum_{l=0}^{T-t-1} (\gamma\lambda)^l \delta_{t+l}, \quad \text{where } \delta_t = r_t + \gamma V_\phi(x_{t+1}) - V_\phi(x_t), \tag{1}$$

$$\mathcal{L}_{\text{ACTOR}}(\theta) = -\mathbb{E}_{\tau \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{\sum_{t=0}^{T-1} L_t} \sum_{t=0}^{T-1} \sum_{i=1}^{L_t} \min \left(\rho_t^i(\theta) \hat{A}_t,\ \text{clip}(\rho_t^i(\theta), 1-\epsilon, 1+\epsilon) \hat{A}_t \right) \right], \tag{2}$$

where V_ϕ denotes the critic model parameterized by ϕ, $\hat{A}_t$ the advantage for turn t, T the total number of turns, L_t the length of turn a_t, and $\rho_t^i(\theta)$ the token-level probability ratio. γ, λ, and ϵ are the discount factor, GAE parameter, and clipping range, respectively.

3 A Preliminary Study of Exploration in LLM Agents

Stochastic generation and entropy regularization encourage diversity in model outputs, but do they also ensure effective exploration of the environment? To examine this question, we conduct a preliminary study on WebShop (Yao et al., 2022) using Qwen3-4B. We compare ***PPO***, with a token-level entropy coefficient of 0.001, and ***PPO w/o entropy***, which differs only in setting this coefficient to 0. We examine generation entropy, state visitation, and task success separately to understand how generation-side exploration relates to agent–environment interaction.

Experimental Setup and Metrics Both configurations use the same environment, prompts, and task sampling setup. We analyze the first 150 training updates over three seeds, with 16 task groups and 8 rollouts per group, totaling 128 episodes per update. State-visitation statistics are collected during training, while the actor is evaluated directly on a fixed set of 128 validation tasks every five updates. Figure 2 shows the training curves. We measure the new-state rate NR, episode revisit rate RR, and validation success rate SR:

$$NR = \frac{N_{\text{new}}}{N_{\text{interact}}}, \qquad RR = \frac{N_{\text{revisit}}}{N_{\text{interact}}}, \qquad SR = \frac{N_{\text{success}}}{N_{\text{episode}}}, \tag{3}$$

where N_{new} counts states first encountered relative to the cumulative training visitation set, N_{revisit} counts interactions returning to a state already visited within the same episode, and N_{interact} counts training environment interactions in the update. State keys combine task identity with the observable page, search field, and clickable items, excluding dialogue history. Initial states enter the novelty count, but resets are not counted as interactions. Thus, NR measures observable-state novelty. For SR, N_{success} and N_{episode} count successful and total validation episodes. We also report token entropy, averaged over all valid generated tokens in an update:

$$H_{\text{token}} = -\frac{1}{\sum_{j=1}^{B} L_j} \sum_{j=1}^{B} \sum_{i=1}^{L_j} \sum_{v \in V} \pi_\theta(v \mid x_j, a_j^{<i}) \log \pi_\theta(v \mid x_j, a_j^{<i}), \tag{4}$$

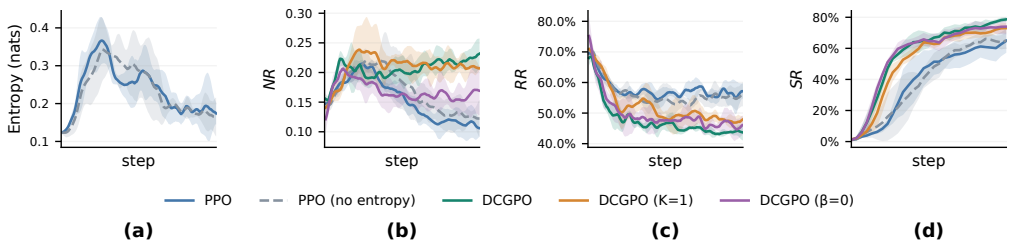


Figure 2: Preliminary study on WebShop with Qwen3-4B, comparing PPO with entropy coefficient 0.001 against PPO w/o entropy. (a) Token entropy. (b) New-state rate (NR). (c) Episode revisit rate (RR). (d) Success rate (SR).

where B is the number of collected turn-level responses, L_j is the number of valid tokens in response a_j, and x_j is its interaction context. This statistic measures conditional token entropy at visited contexts and generated prefixes, rather than entropy over environment actions.

Observations and Motivation. As shown in Figure 2, PPO improves task success while continuing to revisit states frequently: 57.4% of training interactions return to previously visited states within the same episode. Adding the entropy bonus yields slightly higher mean token entropy, but neither a higher new-state rate nor a lower revisit rate. These observations suggest that encouraging generation-side randomness alone does not ensure more effective environment exploration.

The key insight is a mismatch between the level at which diversity is encouraged and the level at which exploration matters: token entropy rewards uncertainty in generation without distinguishing responses by the states or returns they may lead to. Different responses can induce the same environment action, while visiting a new state need not advance the task. Generation therefore supplies possible behaviors, but does not by itself assess which candidate is worth executing. This motivates separating candidate generation from selection based on downstream task consequences, so that limited environment interactions can be directed toward promising behaviors.

4 Method

DCGPO combines a turn-level distributional critic with Distributional-Guided Exploration. The critic supports advantage estimation and provides action-conditioned return statistics that guide which policy-generated turns are executed during training. Figure 1 summarizes the framework.

4.1 Turn-Level Distributional Critic

The standard PPO critic collapses the return into a scalar point estimate, which overlooks variability in downstream outcomes. To retain this information, we introduce a distributional model that predicts the return distribution conditioned on context x and candidate action a. We adopt the same formulation for state-value estimation $V(x)$.

Formally, let $Z^\pi(x) \in \mathcal{Z}$ and $Z^\pi(x,a) \in \mathcal{Z}$ be the random variables representing the cumulative discounted return at context x and after taking action a at context x under policy π, respectively, where $\mathcal{Z}$ denotes the space of return distributions. Their expectations induce the turn-level value function and Q-function:

$$V^\pi(x) = \mathbb{E}[Z^\pi(x)], \qquad Q^\pi(x,a) = \mathbb{E}[Z^\pi(x,a)]. \tag{5}$$

The two functions are both governed by the distributional Bellman operator $\mathcal{T}^\pi$:

$$\mathcal{T}^\pi Z(x,a) \overset{D}{=} r + \gamma\, Z(x',a'), \tag{6}$$

where $x' \sim \mathcal{P}(\cdot|x,a)$, $a' \sim \pi(\cdot|x')$, and $\overset{D}{=}$ denotes equality in distribution. For learning, we parameterize the two distributions as $Z_\psi(x)$ and $Z_\phi(x,a)$ with parameters ψ and ϕ, respectively. Both models are trained to match the projected distributional Bellman target $\Phi\mathcal{T}^\pi Z_\phi(x,a)$, where

Φ projects the target back onto the parametric support, which gives two separate update rules:

$$\begin{aligned}\mathcal{L}_V(\psi) &= \mathbb{E}_{(x,a)\sim\mathcal{D}}\left[d\left(\Phi\mathcal{T}^\pi Z_\phi(x,a),\ Z_\psi(x)\right)\right],\\ \mathcal{L}_Q(\phi) &= \mathbb{E}_{(x,a)\sim\mathcal{D}}\left[d\left(\Phi\mathcal{T}^\pi Z_\phi(x,a),\ Z_\phi(x,a)\right)\right].\end{aligned} \tag{7}$$

where $d(\cdot,\cdot)$ denotes a discrepancy between two distributions. Here π denotes a generic fixed policy being evaluated, and the action conditional in $\mathcal{D}$ is $a \sim \pi(\cdot \mid x)$, so the state-value target averages over that policy's actions. The following result establishes a turn-level error bound for mean-induced advantage estimation under categorical critic learning. The analysis assumes bounded returns and mean-preserving projection without boundary clipping, exact projected distributional Bellman completeness, bounded data-coverage ratios, and independent samples with exact empirical cross-entropy minimization over a fixed finite function class; these conditions are specified in Assumptions 1–4.

Theorem 1 (Turn-Level Advantage Bound Preservation; Informal) *Fix a policy π. Under Assumptions 1–4, let $\bar{Z}$ be the output of categorical Distributional Fitted Policy Evaluation and $\bar{Q} = \mathbb{E}[\bar{Z}]$. For any $\delta \in (0,1)$, with probability at least $1-\delta$,*

$$\mathbb{E}_{(x,a)\sim d^\pi}\left[\left(\bar{Q}(x,a) - \mathbb{E}_{a'\sim\pi(x)}\bar{Q}(x,a') - A^\pi(x,a)\right)^2\right] \le \frac{4}{1-\gamma}\left(\varepsilon_{\text{stat}} + 2R\sqrt{\varepsilon_{\text{stat}}}\right), \tag{8}$$

where d^π is the discounted state-action occupancy measure, A^π is the true turn-level advantage, R bounds the return support, and $\varepsilon_{\text{stat}}$ is a statistical upper bound on the mean critic's squared Bellman residual, defined in Appendix D.

Categorical critic learning thus preserves the form and statistical order of the scalar turn-level bound, with the categorical function-class complexity entering $\varepsilon_{\text{stat}}$. Under the common return-scale and sample-budget conditions of Corollary 1, the turn-level reference upper bound is at most $1/(\gamma\sqrt{L_{\max}})$ times the token-level reference upper bound, where $L_{\max}$ denotes the number of tokens per turn after padding and $\gamma\sqrt{L_{\max}} > 1$. The residual-budget conditions and full proof are provided in Appendix D.

4.2 Distributional-Guided Exploration

The exploration bottleneck in Section 3 motivates separating candidate generation from selection: stochastic sampling supplies diverse turns, while selection considers their downstream task outcomes. Selecting solely by Q-value favors the highest expected return, but does not explicitly account for how concentrated or variable the continuation returns are. A return distribution describes both aspects. In the deterministic environments we study, variation in true returns arises from subsequent policy choices that lead to different outcomes, as illustrated in Figure 3.

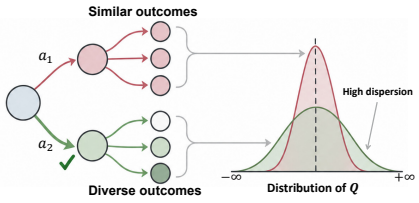


Figure 3: Illustration of Distributional-Guided Exploration

Some candidates may lead to useful solution paths that the current policy has not yet learned to follow reliably, yielding variable downstream returns and modest expected values. Mean-greedy selection may therefore give them few opportunities for execution. We combine expected return with a dispersion bonus to encourage further trials of such candidates, providing experience for learning promising continuations while keeping selection tied to task performance.

Concretely, at context x_t, the policy π_θ generates K candidate turn-level actions $\{a_{t,k}\}_{k=1}^K$. For each candidate action, we compute its expected value $Q_\phi(x_t,a_{t,k}) = \mathbb{E}[Z_\phi(x_t,a_{t,k})]$ and predicted return dispersion $\sigma(x_t,a_{t,k}) = \text{Std}(Z_\phi(x_t,a_{t,k}))$. Candidate actions are then ranked using the mean–dispersion exploration utility:

$$U(x_t,a_{t,k}) = Q_\phi(x_t,a_{t,k}) + \beta\,\sigma(x_t,a_{t,k}), \tag{9}$$

where $\beta > 0$ controls the preference for return dispersion and K controls the breadth of the candidate set. The agent selects the candidate maximizing this utility:

$$a_t = \arg\max_{a\in\{a_{t,k}\}_{k=1}^K} U(x_t,a). \tag{10}$$

Algorithm 1 Distributional-Guided Action Selection (one turn)

Require: Context x, policy π_θ, distributional critic Z_ϕ, exploration weight β, number of candidates K.

1: Sample K candidate turns $\{a^{(k)}\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid x)$.
2: **for** $k = 1$ **to** K **do**
3: Compute mean $\mu^{(k)} = Q_\phi(x, a^{(k)}) = \mathbb{E}[Z_\phi(x, a^{(k)})]$.
4: Compute standard deviation $\sigma^{(k)} = \mathrm{Std}(Z_\phi(x, a^{(k)}))$.
5: Set exploration utility $U^{(k)} = \mu^{(k)} + \beta\, \sigma^{(k)}$.
6: **end for**
7: Select $k^* \leftarrow \arg\max_k U^{(k)}$ (first index in a tie); execute $a^* \leftarrow a^{(k^*)}$, observe reward r and next context x'.
8: **return** a^*, r, x'

At iteration n, the frozen actor and selection critic induce a behavior policy μ_n. Candidate selection thus changes how environment interactions are allocated among policy-generated turns. Return dispersion serves as a sampling preference, whose effects on exploration and task performance are evaluated empirically. The collected trajectories train the critic to evaluate continuation under μ_n and update the base policy across turns; the updated policy generates the next iteration's candidates. Evaluation uses the base policy without selection. Algorithm 1 summarizes the selection procedure.

4.3 Practical Implementation

Distributional Critic. In practice, the value model and the Q-model above are realized by a *single* categorical network and share the same parameters ($\psi = \phi$). They differ only in their input, with the value model reading the context x alone and the Q-model reading it concatenated with the candidate a. Both models are supported on a fixed set of atoms $z_i = V_{\text{MIN}} + i\,\Delta z$ with $i = 0, \ldots, M-1$ and $\Delta z := \frac{V_{\text{MAX}} - V_{\text{MIN}}}{M-1}$, where $V_{\text{MIN}}, V_{\text{MAX}} \in \mathbb{R}$ are the lower and upper bounds of the return support and $M \in \mathbb{N}$ is the number of atoms. The network outputs M logits, and the two categorical distributions are obtained by a softmax over these logits,

$$Z_\psi(x) = z_i \;\text{ w.p. }\; p_i(x) = \frac{\exp(\psi_i(x))}{\sum_{j=0}^{M-1} \exp(\psi_j(x))}, \tag{11}$$

and $Z_\phi(x, a)$ is defined likewise with ϕ in place of ψ. The dispersion term used in equation 9 is computed directly from the categorical distribution,

$$\sigma(x, a) = \sqrt{\sum_{i=0}^{M-1} p_i(x, a)\, z_i^2 - Q_\phi(x, a)^2}, \qquad \text{where} \quad Q_\phi(x, a) = \sum_{i=0}^{M-1} p_i(x, a)\, z_i. \tag{12}$$

At each iteration n, we freeze the candidate-generating policy and the selection critic during rollout collection. The critic is trained to approximate $Z^{\mu_n}(x, a)$, the return distribution after executing candidate a and continuing with this selection procedure, while the state-only input estimates the corresponding value distribution $Z^{\mu_n}(x)$. We construct projected distributional λ-return targets by recursively mixing the next-state value distribution with the next-turn target distribution. The recursion and update details are given in Appendix C.

Policy Update. To learn from selection-guided interactions, each iteration optimizes a turn-level surrogate on trajectories collected by the frozen behavior policy μ_n. Following GSPO (Zheng et al., 2025), we define the length-normalized probability ratio for an entire turn a_t as

$$\rho_t(\theta) = \left(\frac{\pi_\theta(a_t \mid x_t)}{\pi_{\theta_{\text{old}}}(a_t \mid x_t)} \right)^{1/L_t}, \tag{13}$$

where $\pi_\theta(a_t \mid x_t) = \prod_{i=1}^{L_t} \pi_\theta(a_t^i \mid x_t, a_t^{<i})$. The raw advantage $\hat{A}_t = g_t^\lambda - V_\psi(x_t)$ estimates $A^{\mu_n}(x_t, a_t)$ by comparing the turn's estimated return with the state baseline under μ_n. After the processing described in Appendix C, these advantages are held fixed as weights during actor optimization. Let $\mathcal{B}$ be a minibatch of valid turns pooled from the rollout batch $\mathcal{D}_n$, and let $\widetilde{A}_t$ denote

the processed advantage. Omitting the reference-policy KL regularizer for brevity, the actor loss is

$$\mathcal{L}_{\mathrm{ACTOR}}(\theta) = -\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}} \min\Big\{\rho_t(\theta)\widetilde{A}_t,\ \mathrm{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\widetilde{A}_t\Big\}. \tag{14}$$

At a fixed context and for a fixed learning signal, candidate selection induces a bounded local reweighting of the actor surrogate, with $0 \le c_n \le K$ (Appendix C.5). We further examine its practical effect through a softmax-selection ablation with exact candidate-index importance weights applied outside clipping (Appendix C.6). This correction does not improve final validation success at any tested selection temperature (Appendix F.6). These results support the empirical viability of the uncorrected surrogate in the evaluated setting. The full training algorithm appears in Appendix B.

5 Experiments

In this section, we evaluate DCGPO on LLM agent tasks and aim to answer the following questions: (1) Does DCGPO improve task performance over prior online RL methods for LLMs? (2) How do distributional critic learning and dispersion-guided selection contribute to these improvements? (3) Does distribution-guided sampling improve exploration efficiency by reducing redundant revisits and broadening state coverage?

5.1 Experimental Settings

Environments. We train the LLM agents on two challenging benchmarks: WebShop (Yao et al., 2022) and ALFWorld (Shridhar et al., 2021). WebShop is a simulated e-commerce environment in which the agent must navigate multiple types of web pages to find, configure, and purchase a product matching a natural-language instruction. We use its 1,000-product configuration with synthetic goals, which requires compositional language understanding, query reformulation, and strategic exploration. ALFWorld is a parallel interactive environment that aligns abstract text-based simulations with embodied robotic tasks. Each episode supplies a textual objective that the agent must fulfill through iterative interactions with the environment, which comprises 3,827 instances across six types of everyday household activities: Pick & Place, Examine in Light, Clean & Place, Heat & Place, Cool & Place, and Pick Two & Place (abbreviated as Pick, Look, Clean, Heat, Cool, and Pick2, respectively).

Baselines. We compare DCGPO against two families of competitive baselines. The first comprises Actor-Critic methods, PPO (Schulman et al., 2017) and TurnPPO, which require training a separate critic model to provide advantage estimates. TurnPPO is a turn-level PPO variant that maintains a scalar turn-level critic, whose estimated advantage is shared by all tokens within the same response, and adopts the same sequence-level importance-ratio correction as GSPO (Zheng et al., 2025). The second comprises group-based and critic-free methods, namely GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), and GiGPO (Feng et al., 2026), which forgo the critic and instead estimate advantages through group-relative comparisons over the sampled trajectories. Further details are provided in Appendix E.

Training details. We train DCGPO and all baselines on four base models: Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, Qwen3-1.7B, and Qwen3-4B (Yang et al., 2025). In the main text we report the results for Qwen2.5-7B-Instruct and Qwen3-4B, and defer the complete results across all four models to Appendix F. The exploration weight coefficient β is set to 1 across experiments. Hyperparameter configurations and prompts for all baseline RL methods are kept the same as GiGPO (Feng et al., 2026). Full training settings and hyperparameter details are provided in Appendix E.

Evaluation protocol. For final performance comparisons, we evaluate the actor checkpoint at training step 200 for every method. Each policy generates one response per turn, with candidate selection and critic-based ranking disabled during evaluation. We report success rates on the seen and unseen splits of ALFWorld, and both average task score and success rate on WebShop. Full evaluation settings, task splits, and metric definitions are provided in Appendix E.3.

Table 1: Performance on ALFWorld and WebShop. Results are reported as mean ± standard deviation across 3 training seeds, following the evaluation protocol in Appendix E.3. For ALFWorld, we report the success rate (%) of each subtask and the overall result on the seen split, together with the overall result on the unseen split. For WebShop, we report the average score and the success rate (%). Bold entries denote the best result per metric for each base model.

Method	ALFWorld								WebShop	
	Seen							Unseen	Score	Succ.
	Pick	Look	Clean	Heat	Cool	Pick2	All	All		
Base: Qwen2.5-7B-Instruct										
PPO	93.0± 2.4	69.2± 6.8	92.2± 2.9	79.9± 4.8	72.9± 8.9	75.0± 5.6	82.5± 2.4	76.4± 3.4	86.59± 0.90	76.40± 2.16
TurnPPO	90.5± 3.4	70.1± 7.4	82.3± 10.1	67.4± 14.2	67.1± 0.8	71.3± 13.5	76.9± 5.6	75.5± 6.8	79.49± 14.21	70.80± 11.98
GRPO	91.1± 0.6	66.7± 11.2	85.2± 7.5	74.3± 3.2	67.6± 4.3	62.0± 11.6	76.6± 2.6	72.0± 4.5	83.13± 1.97	73.80± 1.56
GSPO	97.5± 1.5	69.2± 5.1	90.1± 10.7	83.3± 5.5	75.6± 0.8	78.2± 7.0	84.6± 3.9	81.8± 2.4	86.31± 3.80	74.27± 4.80
GiGPO	99.1± 1.7	87.2± 6.8	**97.5± 2.1**	84.0± 8.7	88.9± 1.5	93.1± 2.4	93.1± 0.9	90.1± 3.0	87.84± 3.22	77.60± 2.31
Ours	**100.0± 0.0**	**89.7± 6.8**	97.1± 2.6	**95.8± 2.1**	**93.3± 2.3**	**94.9± 6.6**	**96.0± 1.7**	**91.4± 3.0**	**92.40± 1.56**	**83.60± 1.70**
Base: Qwen3-4B										
PPO	87.9± 4.8	62.4± 13.2	68.7± 4.3	65.3± 1.2	68.0± 11.6	41.7± 4.8	67.8± 0.4	58.7± 8.3	82.14± 3.63	69.53± 4.15
TurnPPO	90.2± 1.5	70.9± 8.2	60.5± 3.3	75.7± 8.7	64.0± 12.0	60.2± 5.3	71.2± 3.7	67.1± 2.7	88.85± 2.31	77.60± 3.30
GRPO	94.0± 3.6	65.8± 3.9	70.4± 9.8	59.7± 12.0	63.6± 4.3	51.4± 8.7	70.2± 4.3	61.9± 8.2	81.92± 2.47	67.73± 2.60
GSPO	88.9± 4.4	67.5± 3.9	72.8± 4.5	76.4± 6.7	67.6± 5.6	63.0± 2.9	74.1± 3.9	66.2± 4.8	85.69± 5.14	73.80± 4.68
GiGPO	95.9± 2.0	82.1± 13.3	94.7± 3.1	85.4± 9.6	81.8± 3.4	75.5± 2.1	87.1± 4.2	83.7± 2.5	88.21± 1.47	75.87± 1.53
Ours	**100.0± 0.0**	**89.7± 6.8**	**96.3± 0.0**	**92.4± 6.0**	**87.1± 5.6**	**90.7± 2.1**	**93.6± 0.0**	**89.6± 1.4**	**89.68± 0.95**	**80.60± 1.31**

5.2 PERFORMANCE

As for **ALFWorld**, as demonstrated in Table 1, DCGPO attains the best overall success rate on both the seen and unseen splits of ALFWorld for both base models. The ordering among the baselines is itself instructive. PPO and GRPO perform markedly worse, because both rely on token-level credit assignment, which is misaligned with the turn-level granularity at which agent decisions are actually made. GSPO fixes this issue by adopting a sequence-level importance ratio, and GiGPO improves further through its finer turn-level credit assignment, becoming the strongest baseline. The performance gap remains substantial on long-horizon subtasks, where DCGPO's advantage is most evident. On Pick2 with Qwen3-4B, DCGPO reaches 90.7%, compared with 41.7% for PPO and 75.5% for GiGPO. DCGPO uses predicted return means and dispersion to guide training-time candidate selection. These gains are measured with the trained base policy acting independently, showing that the benefit persists without evaluation-time selection. On **WebShop**, DCGPO again achieves the best score and success rate on both base models. On Qwen2.5-7B-Instruct, DCGPO attains 83.60% success versus 77.60% for GiGPO, and Qwen3-4B shows the same pattern (80.60% vs. 75.87% success).

The same trend extends to the smaller models, as shown in Tables 4 and 5 in Appendix F. Qwen3-1.7B is by far the weakest base policy on ALFWorld, with PPO reaching an overall success rate of only 48.0% on the seen split, yet DCGPO still attains 87.9%. Weaker models exhibit the most severe behavioral collapse, repeatedly retracing a small set of trajectories under stochastic sampling, which leaves the largest room for exploration-guided improvement. The complete results over all four base models are reported in Appendix F, and successful and failed cases are provided in Appendix G.

5.3 ABLATION STUDIES

Ablation of components. To assess the contribution of each component, we compare four configurations on WebShop: (1) TurnPPO with a scalar critic; (2) DCGPO with only the distributional critic and no candidate selection ($K = 1$); (3) a greedy variant that selects among candidates by predicted mean return ($K = 3, \beta = 0$); (4) the full method ($K = 3, \beta = 1$). As illustrated in Figure 4, success rates improve progressively across these configurations for both base models. Replacing the scalar critic with the distributional critic increases success from 70.80% to 78.80% on Qwen2.5-7B-Instruct and from 77.60% to 77.87% on Qwen3-4B, indicating that the benefit of critic replacement alone varies across models. Greedy candidate selection further raises success, while adding the dispersion term achieves the highest success rates of 83.60% and 80.60%. These results support the contribution of both distributional critic learning and candidate selection, and show that incorporating return dispersion provides an additional benefit beyond mean-only selection.

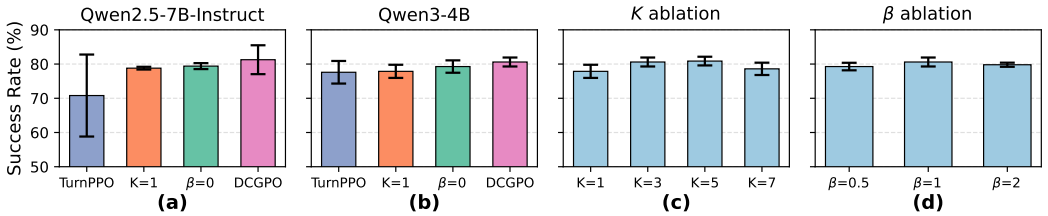


Figure 4: Ablation study on Webshop. All panels report success rate (%), with bars showing the mean and error bars the standard deviation over 3 seeds. (a) Component ablation on Qwen2.5-7B-Instruct and (b) on Qwen3-4B. (c) Effect of the K hyperparameter on Qwen3-4B. (d) Effect of β on Qwen3-4B.

Ablation of candidate budget K and exploration weight β. Figure 4(c)–(d) reports the ablations on WebShop with Qwen3-4B. Success improves from 77.87% at $K = 1$ to 80.60% at $K = 3$, with little further gain at $K = 5$ and a decline at $K = 7$. Training time is $1.13\times$ that of $K = 1$ at $K = 3$, compared with $1.31\times$ at $K = 5$. For β, mean success peaks at 80.60% with $\beta = 1$, compared with 79.27% at $\beta = 0.5$ and 79.80% at $\beta = 2$. Balancing performance and training cost, we adopt $K = 3$ and $\beta = 1$.

5.4 Distributional-Guided Selection Improves Exploration

DCGPO improves visitation and task performance. Section 3 identifies a gap between generation-side randomness and effective environment exploration. We revisit this issue on Qwen3-4B in WebShop by adding DCGPO and its selection ablations to Figure 2(b)–(d), using the same protocol and metrics. Relative to PPO and PPO w/o entropy, DCGPO roughly doubles the new-state rate and reduces episode revisits from 57.4% to 43.7%, while attaining higher SR. Thus, DCGPO addresses the observed limitation by coupling more novel state visitation and fewer revisits with improved task performance.

Candidate selection improves the learned policy. Compared with the $K = 1$ variant, full DCGPO achieves a higher NR, fewer episode revisits, and a 6.7 percentage-point gain in SR. Since validation disables candidate selection, this gain reflects a stronger learned actor, consistent with the motivation that return-guided selection can help generate useful training experience. Fixing $K = 3$, the $\beta = 0$ variant selects candidates solely by predicted mean return. Adding the dispersion term increases the NR from 16.9% to 23.4% and SR by 4.9 percentage points in the analysis window. RR also decreases. These results indicate that dispersion-guided selection sustains state novelty beyond mean-greedy selection, with an accompanying policy benefit in this setting. Together, the comparisons support using downstream return information to allocate training interactions: candidate selection improves the actor, and return dispersion provides an additional exploration preference beyond expected return alone.

6 Conclusions

In this work, we examine exploration inefficiency in multi-turn LLM agents, where stochastic generation can leave repeated trajectories and limited state coverage, reducing the effective use of environment interactions. We propose DCGPO, which combines turn-level distributional value learning with distribution-guided exploration. The critic supports advantage estimation and provides action-conditioned return means and dispersion to guide training-time sampling beyond greedy mean-value selection. Our fixed-policy analysis shows that categorical critic learning preserves the form and statistical order of the scalar turn-level error bound for mean-induced advantage estimation. Experiments demonstrate the effectiveness of this approach, with component ablations and visitation analyses examining its contributions to task performance and exploration. These findings support using downstream return information to guide training interactions in multi-turn LLM agents.

AI USE STATEMENT

In this work, we used generative AI tools to aid and polish the writing of this paper and for the retrieval and discovery of related work. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, support qualitative and thematic data analysis, interpret results, generate synthetic data sets, assist with translation, or clean and reformat datasets. We have reviewed all AI-assisted work, and all AI-generated content was verified by three authors. The authors have proofread the final manuscript. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work focuses on improving the training and exploration efficiency of reinforcement learning for large language model agents. It does not involve human subjects, human participants, or personally identifiable information. All experiments are conducted on publicly available simulated environments and benchmarks, and this work does not release or collect any new datasets. To the best of our knowledge, the proposed method and its experimental study raise no ethical concerns regarding privacy, fairness, safety, or potential misuse beyond those already inherent to the underlying large language models.

REPRODUCIBILITY STATEMENT

The code implementing DCGPO is provided in the supplementary material. The pseudocode for DCGPO is presented in Algorithm 2. Implementation details, including environment setup, evaluation metrics, and hyperparameters, are provided in Appendix E.

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 (Volume 1: Long Papers)*, pp. 12248–12267, 2024.

Marc G Bellemare, Will Dabney, and Rémi Munos. A distributional perspective on reinforcement learning. In *International conference on machine learning*, pp. 449–458. Pmlr, 2017.

Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

Junying Chen, Zhenyang Cai, Ke Ji, Xidong Wang, Wanlong Liu, Rongsheng Wang, Jianye Hou, and Benyou Wang. Huatuogpt-o1, towards medical complex reasoning with llms. *arXiv preprint arXiv:2412.18925*, 2024.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. *Advances in neural information processing systems*, 30, 2017.

Ning Dai, Zheng Wu, Renjie Zheng, Ziyun Wei, Wenlei Shi, Xing Jin, Guanlin Liu, Chen Dun, Liang Huang, and Lin Yan. Process supervision-guided policy optimization for code generation. *arXiv preprint arXiv:2410.17621*, 2024.

Guanting Dong, Junting Lu, Junjie Huang, Wanjun Zhong, Longxiang Liu, Shijue Huang, Zhenyu Li, Yang Zhao, Xiaoshuai Song, Xiaoxi Li, et al. Agent-world: Scaling real-world environment synthesis for evolving general agent intelligence. *arXiv preprint arXiv:2604.18292*, 2026a.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. In *International Conference on Learning Representations*, volume 2026, pp. 16981–17017, 2026b.

Shihan Dou, Yan Liu, Haoxiang Jia, Enyu Zhou, Limao Xiong, Junjie Shan, Caishuang Huang, Xiao Wang, Xiaoran Fan, Zhiheng Xi, et al. Stepcoder: improving code generation with reinforcement learning from compiler feedback. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4571–4585, 2024.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

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

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

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.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv preprint arXiv:2501.03262*, 1(4), 2025.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

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.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

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

Jinyi Liu, Yiboyun Chen, Hongyao Tang, Yi Ma, Shuyue Hu, Yang Chen, Fei Ni, Qiaosheng Zhang, Lei Bai, Yan Zheng, et al. Beyond scalar critics: A distributional perspective on reinforcement learning with verifiable rewards for llms. In *The 1st Workshop on Scaling Post-training for LLMs*, 2026.

Ziyu Liu, Zeyi Sun, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Haodong Duan, Dahua Lin, and Jiaqi Wang. Visual-rft: Visual reinforcement fine-tuning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2034–2044. IEEE, 2025.

Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

OpenAI. Learning to reason with llms. `https://openai.com/index/learning-to-reason-with-llms`, September 2024. Accessed: 2026-09-03.

Jiangweizhi Peng, Yuanxin Liu, Ruida Zhou, Charles Fleming, Zhaoran Wang, Alfredo Garcia, and Mingyi Hong. Hiper: Hierarchical reinforcement learning with explicit credit assignment for large language model agents. *arXiv preprint arXiv:2602.16165*, 2026.

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

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

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

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

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

Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv preprint arXiv:2504.07615*, 2025.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning Text and Embodied Environments for Interactive Learning. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2010.03768.

Joykirat Singh, Raghav Magazine, Yash Pandya, and Akshay Nambi. Agentic reasoning and tool integration for llms via reinforcement learning. *arXiv preprint arXiv:2505.01441*, 2025.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

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

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.

Muning Wen, Ziyu Wan, Jun Wang, Weinan Zhang, and Ying Wen. Reinforcing llm agents via policy optimization with action decomposition. *Advances in Neural Information Processing Systems*, 37: 103774–103805, 2024.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, et al. Agentprm: Process reward models for llm agents via step-wise promise and progress. In *Proceedings of the ACM Web Conference 2026*, pp. 4184–4195, 2026.

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

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.

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

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*, 2025b.

Sheng Zhang, Qianchu Liu, Guanghui Qin, Tristan Naumann, and Hoifung Poon. Med-rlvr: Emerging medical reasoning from a 3b base model via reinforcement learning. *arXiv preprint arXiv:2502.19655*, 2025c.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62178–62209. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/zhou24t.html.

Zhijian Zhou, Long Li, Xuan Zhang, Zongkai Liu, Yanting Miao, Yuchen Liu, Deshu Chen, Ke Li, Xing Sun, Ruoxi Jiang, et al. Disppo: Quantile-based distributional reinforcement learning for large language models. In *Forty-third International Conference on Machine Learning*, 2026.

A Related Work

Reinforcement learning for LLMs. RL has emerged as a key paradigm for teaching models to reason, plan, and interact with dynamic environments (Zhang et al., 2025b). The literature can be broadly divided into two methodological threads. The first focuses on alignment, where RL from human feedback (RLHF) (Christiano et al., 2017) tunes pre-trained models to follow instructions and reflect human preferences, typically optimized with Proximal Policy Optimization (PPO) (Schulman et al., 2017). Preference-based alternatives such as direct preference optimization (DPO) (Rafailov et al., 2023), KTO (Ethayarajh et al., 2024), and SimPO (Meng et al., 2024) further simplify alignment by optimizing a preference loss without an explicit reward model. The second focuses on reasoning, where RL with verifiable rewards (RLVR) (Lambert et al., 2024) incentivizes long chain-of-thought behaviors through rule-based accuracy or unit-test signals, a direction popularized by DeepSeek-R1 (Guo et al., 2025). This thread is largely driven by a family of critic-free algorithms, RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), GSPO (Zheng et al., 2025), DAPO (Yu et al., 2026), and REINFORCE++ (Hu, 2025), which replace the value function of PPO with advantages estimated from a group of sampled responses, thereby reducing memory overhead and enabling large-scale RL training. These techniques have been successfully deployed in a variety of application domains, including competitive programming and software engineering (Yu et al., 2026; Dai et al., 2024), agentic tasks such as web browsing and tool use (Team et al., 2025; Qian et al., 2026), multimodal reasoning over images and videos (Shen et al., 2025; Liu et al., 2025), robotic manipulation (Kim et al., 2024; Black et al., 2024), and medical question answering (Chen et al., 2024; Zhang et al., 2025c). Despite these advances, current methods still face fundamental challenges in reward design, training stability, and scalability (Wang et al., 2025), especially when moving beyond tasks with objective verifiers to open-ended or subjective domains. A distributional perspective has recently emerged for LLM post-training: DistRLVR (Liu et al., 2026) and DisPPO (Zhou et al., 2026) learn token-level return distributions to sharpen value estimates and tail-aware advantages under terminal verifiable rewards, confirming the value of distributional modeling. However, both remain confined to token-level MDPs over single-turn reasoning tasks and neither addresses the multi-turn, sparse-reward setting of Agentic RL nor uses predicted return dispersion to select among candidate turns, which is the focus of our work.

RL algorithms for agentic tasks. While directly applying single-turn RL algorithms such as GRPO (Shao et al., 2024) to multi-turn agentic tasks reveals notable limitations (Wang et al., 2025), recent approaches broadly fall into three categories. The first category concerns turn-based credit assignment. In this vein, ArCHer (Zhou et al., 2024) and Turn PPO (Li et al., 2026) conceptualize agent responses as macro-actions within turn-level MDP formulations, whereas AgentPRM (Xi et al., 2026) supplants costly Monte Carlo step-labeling with temporal-difference (TD) learning and GAE for turn-level process rewards, thereby solidifying the turn as the fundamental unit of credit attribution. The second category comprises group-based credit estimation, which refines per-step advantages through structural grouping or hierarchical architectures. Specifically, GiGPO (Feng et al., 2026) and HGPO (He et al., 2026) leverage anchor-state or context-consistent grouping mechanisms augmented with adaptive weighting, while HiPER (Peng et al., 2026) jointly optimizes subgoal planning and low-level execution via hierarchical advantage estimation, particularly addressing the challenges inherent in sparse-reward settings. The third category pertains to uncertainty-driven exploration strategies. ARPO (Dong et al., 2026b) introduces entropy-based adaptive rollouts to facilitate step-level exploration during ambiguous tool-calling procedures, and T²PO (Wang et al., 2026) concurrently monitors uncertainty at both the token and turn levels to dynamically orchestrate intervention triggers or resampling, thus substantially bolstering training stability and exploration efficiency. In parallel, Agentic RL has been applied to a wide range of domain-specific tasks like research agents (Jin et al., 2025), code generation and software engineering (Dai et al., 2024; Dou et al., 2024). However, these designs are often tied to specific contexts and are therefore outside this paper's scope.

B ALGORITHM

Algorithm 2 Distributional Critic Guided Policy Optimization (DCGPO)

Require: Policy π_θ; distributional critic Z_ϕ shared by the value head $V_\phi(x)$ and the Q-head $Q_\phi(x, a)$; exploration weight β; number of candidates K; discount factor γ; λ; clipping range ϵ; atom support $\{z_i\}_{i=0}^{M-1}$.

Initialize θ, ϕ.
for iteration $n = 1$ **to** N **do**
  Freeze snapshots $\theta_{\text{old}} \leftarrow \theta$, $\phi_{\text{old}} \leftarrow \phi$; initialize an empty rollout batch $\mathcal{D}_n$.
  Collect $\mathcal{D}_n$ under μ_n using Algorithm 1 with the frozen snapshots and first-index tie-breaking.
  Record rewards, endpoint status, and old base-policy log-probabilities $\log \pi_{\theta_{\text{old}}}(a_t \mid x_t)$.
  for each trajectory in $\mathcal{D}_n$ **do**
    Compute $\bar{p}_t$ backward using Equation equation 19 and frozen state-only predictions from $Z_{\phi_{\text{old}}}$; bootstrap nonterminal cutoffs as in Appendix C.
    Extract target means: $g_t^\lambda \leftarrow \sum_i z_i \bar{p}_{t,i}$.
    Compute behavior-return advantage: $\hat{A}_t \leftarrow g_t^\lambda - V_{\phi_{\text{old}}}(x_t)$.
  end for
  Broadcast each $\hat{A}_t$ to all tokens of turn t and whiten across the batch.
  Store targets and advantage weights in $\mathcal{D}_n$ and hold them fixed during all updates (see Appendix C for the optional invalid-action penalty).
  ▷ Critic update
  for $c = 1$ **to** N_c **do**
    Sample a minibatch $\mathcal{B} = \{(x_t, a_t, \bar{p}_t)\}$ from $\mathcal{D}_n$.
    Value loss: $\mathcal{L}_V = -\frac{1}{|\mathcal{B}|} \sum_{(x_t, \bar{p}_t) \in \mathcal{B}} \sum_{i=0}^{M-1} \bar{p}_{t,i} \log p_i(x_t)$.
    Q loss: $\mathcal{L}_Q = -\frac{1}{|\mathcal{B}|} \sum_{(x_t, a_t, \bar{p}_t) \in \mathcal{B}} \sum_{i=0}^{M-1} \bar{p}_{t,i} \log p_i(x_t, a_t)$.
    Update ϕ by minimizing $\mathcal{L}_V + \mathcal{L}_Q$.
  end for
  ▷ Clipped actor surrogate; no exact behavior-policy importance correction
  for $i = 1$ **to** N_a **do**
    Sample a minibatch from $\mathcal{D}_n$.
    Compute sequence-level ratio $\rho_t(\theta) = (\pi_\theta(a_t \mid x_t) / \pi_{\theta_{\text{old}}}(a_t \mid x_t))^{1/L_t}$.
    Actor loss $\mathcal{L}_{\text{ACTOR}}(\theta) = -\frac{1}{|\mathcal{D}_n|} \sum_{t \in \mathcal{D}_n} \min\Big\{ \rho_t(\theta) \widetilde{A}_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\, \widetilde{A}_t \Big\}$.
    Update θ by minimizing $\mathcal{L}_{\text{ACTOR}}$ (optionally plus a KL penalty).
  end for
end for
return θ, ϕ

C Details on Distributional Critic

This appendix completes the implementation-level description of the Distributional Critic deferred from the main text. Rather than collapsing each return into a scalar, the critic maintains a categorical distribution over a common, fixed support, following Bellemare et al. (2017).

C.1 Categorical Representation and Heads

The support is a uniform grid of atoms $\{z_i = V_{\mathrm{MIN}} + i\Delta z\}_{i=0}^{M-1}$ spanning the return interval $[V_{\mathrm{MIN}}, V_{\mathrm{MAX}}]$, with spacing $\Delta z := \frac{V_{\mathrm{MAX}} - V_{\mathrm{MIN}}}{M-1}$. The categorical form is useful because, with terminal-only task rewards and $\gamma = 0.95$, the return at turn t is $G_t = \gamma^{T-1-t} r_{T-1}$: continuations from the same context can succeed at different turns, yielding distinct discounted returns whose standard deviation is generally not determined by their mean.

The Q-head is a network $\phi : \mathcal{X} \times \mathcal{A} \to \mathbb{R}^M$ mapping the context–action pair to M logits; softmax normalization gives the atom probabilities, and the Q-value is the mean of the resulting distribution:

$$\begin{aligned} Z_\phi(x,a) = z_i \quad \text{w.p.} \quad p_i(x,a) &:= \frac{e^{\phi_i(x,a)}}{\sum_j e^{\phi_j(x,a)}}, \\ Q_\phi(x,a) = \mathbb{E}[Z_\phi(x,a)] &= \sum_{i=0}^{M-1} p_i(x,a) z_i. \end{aligned} \tag{15}$$

The value head $\psi : \mathcal{X} \to \mathbb{R}^M$ is defined analogously from the context x alone, yielding the value distribution and the value function:

$$\begin{aligned} Z_\psi(x) = z_i \quad \text{w.p.} \quad p_i(x) &:= \frac{e^{\psi_i(x)}}{\sum_j e^{\psi_j(x)}}, \\ V_\psi(x) = \mathbb{E}[Z_\psi(x)] &= \sum_{i=0}^{M-1} p_i(x) z_i. \end{aligned} \tag{16}$$

In our implementation the two heads reside in a single network ($\psi = \phi$), differing only in whether the action is concatenated to the input. In all experiments, we use $M = 51$ uniformly spaced atoms over $[V_{\min}, V_{\max}] = [-2, 10]$.

C.2 Evaluation Target

At iteration n, the frozen actor and selection critic snapshots $(\theta_{\mathrm{old}}, \phi_{\mathrm{old}})$ induce the behavior policy μ_n through Algorithm 1, with ties resolved by the first maximizing candidate index. The rollout batch $\mathcal{D}_n$ contains trajectories from this fixed behavior; the updates below do not mix in trajectories from earlier iterations. The generic evaluation policy π in the main-text critic definitions is therefore instantiated as μ_n, so the intended critic targets are $Z^{\mu_n}(x)$ and $Z^{\mu_n}(x,a)$, with subsequent actions selected by μ_n rather than by the base generator alone.

C.3 Projected Distributional λ-Return Targets

We implement GAE (Schulman et al., 2015) through a distributional TD(λ) target. Consider a trajectory from $\mathcal{D}_n$ with turns $t = 0, \ldots, T-1$. The recursion below uses zero continuation at a terminal endpoint, including the task's finite-horizon endpoint; a collection cutoff before the task ends instead bootstraps from the endpoint value distribution, as described at the end of this subsection.

Let $\nu_t = \mathcal{L}(Z_{\phi_{\mathrm{old}}}(x_t))$ denote the state-only prediction of the frozen critic snapshot. All bootstrap predictions, target distributions, and advantages are computed before parameter updates and held fixed throughout the iteration. For brevity, $p_j(x_t)$ and $V_\psi(x_t)$ in the target and advantage equations below refer to this snapshot, while predictions in the training losses use the trainable parameters. Define $b_t(z) = r_t + \gamma z$ and let $(b_t)_{\#}\nu$ denote the distribution of $r_t + \gamma X$ for $X \sim \nu$. The projected target distributions satisfy

$$\begin{aligned} \widehat{\eta}^{\lambda}_{T-1} &= \Phi \delta_{r_{T-1}}, \\ \widehat{\eta}^{\lambda}_{t} &= \Phi\big((b_t)_{\#}\left[(1-\lambda)\nu_{t+1} + \lambda \widehat{\eta}^{\lambda}_{t+1}\right]\big), \qquad t = T-2, \ldots, 0. \end{aligned} \tag{17}$$

The weighted sum inside the brackets is a mixture of probability measures, not a weighted sum of random return values, so the bootstrap distribution is retained throughout the recursion. Omitting Φ gives the corresponding representation-independent distributional TD(λ) recursion; our categorical implementation projects at every turn.

On the fixed atom support, the categorical projection is defined by the interpolation weights

$$h_i(y) = \left[1 - \frac{\left| [y]_{V_{\mathrm{MIN}}}^{V_{\mathrm{MAX}}} - z_i \right|}{\Delta z} \right]_0^1, \tag{18}$$

where $[\cdot]_a^b$ clips its argument to $[a, b]$. Writing $\bar{p}_{t,i}$ for the probability mass of $\widehat{\eta}_t^\lambda$ at z_i, the backward recursion is computed in three steps—seed, mix, and project:

$$\begin{aligned} \bar{p}_{T-1,i} &= h_i(r_{T-1}), \\ m_{t,j} &= (1-\lambda) p_j(x_{t+1}) + \lambda \bar{p}_{t+1,j}, \\ \bar{p}_{t,i} &= \sum_{j=0}^{M-1} m_{t,j}\, h_i(r_t + \gamma z_j). \end{aligned} \tag{19}$$

The reward and discount transform each atom location, while the projection redistributes its probability mass over neighboring atoms. For a nonterminal collection cutoff, initialize $\widehat{\eta}_T^\lambda = \nu_T$ and apply the same recursion starting at $t = T - 1$; equivalently, set $\bar{p}_{T,i} = p_i(x_T)$.

C.4 Critic Loss and Advantages

Both input forms are fitted to the same target for each observed turn, using the cross-entropy losses

$$\ell_{V,t} = -\sum_i \bar{p}_{t,i} \log p_i(x_t), \qquad \ell_{Q,t} = -\sum_i \bar{p}_{t,i} \log p_i(x_t, a_t), \tag{20}$$

which minimize $D_{\mathrm{KL}}(\bar{p}_t \| p(\cdot))$ with respect to the predictions. Since $a_t \sim \mu_n(\cdot \mid x_t)$, fitting the state-only input averages the targets over the behavior policy's actions, consistently with $V^{\mu_n}(x) = \mathbb{E}_{a \sim \mu_n} Q^{\mu_n}(x, a)$; this specifies the evaluation target without assuming exact realizability or convergence.

The scalar return and raw actor advantage are extracted only after the distributional target has been constructed:

$$g_t^\lambda = \sum_i z_i \bar{p}_{t,i}, \qquad \hat{A}_t = g_t^\lambda - V_\psi(x_t). \tag{21}$$

Without support-boundary clipping, categorical projection preserves expectations, so the means follow the scalar GAE recursion

$$g_t^\lambda = r_t + \gamma \left((1-\lambda) V_\psi(x_{t+1}) + \lambda g_{t+1}^\lambda \right), \tag{22}$$

and, with the corresponding terminal or bootstrap endpoint value, the advantage admits the usual GAE identity

$$\hat{A}_t = \sum_{\ell=0}^{T-t-1} (\gamma\lambda)^\ell \delta_{t+\ell}, \qquad \delta_t = r_t + \gamma V_\psi(x_{t+1}) - V_\psi(x_t). \tag{23}$$

The raw advantage is broadcast to the turn's valid response tokens and whitened before the actor update; when enabled, the current turn's invalid-action penalty is subtracted after whitening and is excluded from the critic targets.

Mean preservation does not imply variance preservation. When clipping occurs, Equation equation 21 uses the actual projected target mean. For a deterministic return $c \in [z_j, z_{j+1}]$, linear categorical projection yields

$$\mathrm{Var}(\Phi\delta_c) = (c - z_j)(z_{j+1} - c) \le \frac{\Delta z^2}{4},$$

so the learned categorical dispersion used for selection can remain positive even when the true return is deterministic.

C.5 Local Reweighting under Candidate Selection

This subsection concerns the actor update rather than the critic. Fix a context x and the rollout snapshots, and let $p_n(\cdot \mid x)$ be the actual candidate proposal, equal to $\pi_{\theta_{\text{old}}}(\cdot \mid x)$ in our sampling formulation. Each of the K candidates has marginal distribution p_n, and the selector returns one of them. Hence

$$\mu_n(a \mid x) \le \mathbb{E}\left[\sum_{k=1}^{K} \mathbf{1}\{a^{(k)} = a\}\right] = K p_n(a \mid x). \tag{24}$$

Thus $c_n = \mu_n / p_n$ on the proposal support satisfies $0 \le c_n \le K$ and $\mathbb{E}_{p_n}[c_n] = 1$. Let $\ell_\theta(x, a, A)$ denote the clipped actor term. For a fixed advantage function $A(x, a)$, changing measure and using positive homogeneity in A yields

$$\mathbb{E}_{a\sim\mu_n}[\ell_\theta(x, a, A)] = \mathbb{E}_{a\sim p_n}[c_n \ell_\theta(x, a, A)] = \mathbb{E}_{a\sim p_n}[\ell_\theta(x, a, c_n A)]. \tag{25}$$

For random advantages, both sides retain the same conditional signal distribution given (x, a). Selection therefore reweights locally available actions, while history visitation and continuation returns still depend on μ_n. Updating turns throughout the collected trajectories provides a way to learn both useful actions and their continuations, with the updated policy supplying subsequent candidates, which motivates transfer to standalone execution.

C.6 Local Importance Correction for Stochastic Selection

The reweighting identity in Appendix C.5 describes how selection changes the actor's training distribution. To test the practical effect of explicitly correcting the current selection step, we consider a stochastic variant of the selector. Fix a context x and the rollout snapshots at iteration n. Let $C = (a^{(1)}, \dots, a^{(K)})$ be an ordered tuple of candidates sampled independently from $p_n(\cdot \mid x)$, and select an index J with probability

$$q_j(C, x) = \frac{\exp\big(U(x, a^{(j)})/T_{\text{sel}}\big)}{\sum_{k=1}^{K} \exp\big(U(x, a^{(k)})/T_{\text{sel}}\big)}, \qquad T_{\text{sel}} > 0, \tag{26}$$

where U is the mean–dispersion utility in Equation equation 9. The executed turn is $a^{(J)}$.

Correction in the candidate-index space. Write $p_n^K(C \mid x) = \prod_{k=1}^{K} p_n(a^{(k)} \mid x)$. The joint distribution of the tuple and selected index, and a reference distribution with uniform index selection, are respectively

$$P_q(C, J \mid x) = p_n^K(C \mid x)\, q_J(C, x), \qquad P_U(C, J \mid x) = \frac{p_n^K(C \mid x)}{K}. \tag{27}$$

Because the softmax selector has full support, their importance ratio is

$$w_J = \frac{P_U(C, J \mid x)}{P_q(C, J \mid x)} = \frac{1}{K q_J(C, x)}. \tag{28}$$

For fixed candidate values $f_1, \dots, f_K$,

$$\mathbb{E}_{J\sim q}[w_J f_J \mid C, x] = \sum_{j=1}^{K} q_j(C, x) \frac{f_j}{K q_j(C, x)} = \frac{1}{K} \sum_{j=1}^{K} f_j. \tag{29}$$

If $f_j = f(x, a^{(j)})$ is integrable, averaging over the sampled candidates therefore recovers $\mathbb{E}_{a\sim p_n(\cdot|x)}[f(x, a)]$. The same statement holds for a random learning signal when its conditional distribution given (x, a) is kept identical on both sides. This establishes an exact correction of the current candidate-index selection step. It does not require evaluating the marginal behavior probability $\mu_n(a \mid x)$, and the realized w_J is not generally equal to $p_n(a^{(J)} \mid x)/\mu_n(a^{(J)} \mid x)$.

Actor update and scope. For each executed turn t, we record q_{t,J_t} during rollout and form $w_t = 1/(Kq_{t,J_t})$. The corrected policy-gradient term is

$$\mathcal{L}^{\text{corr}}_{\text{ACTOR}}(\theta) = -\frac{1}{|\mathcal{B}|}\sum_{t\in\mathcal{B}} \text{sg}(w_t)\min\Big\{\rho_t(\theta)\widetilde{A}_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\widetilde{A}_t\Big\}, \tag{30}$$

where sg denotes stop-gradient and $\mathcal{B}$ is the turn minibatch defined in Equation equation 14. The weight multiplies the clipped loss without being clipped, self-normalized, or raised to a response-length-dependent power. Old base-policy probabilities and the GSPO ratio ρ_t remain unchanged. Only the policy-gradient term is reweighted; the critic losses and explicit KL and entropy regularizers retain their original weighting. Any current-turn format penalty included in $\widetilde{A}_t$ is consequently weighted with that turn's policy-gradient term.

The identity above concerns a fixed context and learning signal. In multi-turn training, visited histories and continuation returns still come from the selected behavior, and batch-based advantage processing and the clipped, length-normalized surrogate are retained. The corrected update therefore remains a surrogate for learning from these trajectories; Equation equation 29 does not identify its expected gradient with the gradient of the standalone actor's return. Hard argmax assigns zero probability to unselected indices conditional on C, so this full-support candidate-index correction does not apply directly to the default hard selector.

Second moment of the correction. Conditioned on a candidate tuple,

$$\mathbb{E}_{J\sim q}[w_J^2 \mid C, x] = \frac{1}{K^2}\sum_{j=1}^{K}\frac{1}{q_j(C,x)}. \tag{31}$$

Small index propensities can therefore produce large inverse-probability weights. This is compatible with $c_n = \mu_n/p_n \le K$: the bound in Appendix C.5 concerns the forward marginal reweighting, whereas w_J corrects in the reverse direction on the candidate-index space. Appendix F.6 evaluates the net effect of this correction on learning.

C.7 Scope and Limitations

The selection critic is inherited from preceding updates and need not accurately evaluate the newly induced μ_n at the start of collection; successive updates approximately track the changing behavior, while predictions for unexecuted candidates rely on function approximation from collected data. The fixed-policy theorem concerns idealized categorical evaluation and does not directly guarantee the practical projected λ-return estimates or the changing-policy training process.

For each actor update, the collected trajectories, old base-policy probabilities, and processed advantage weights are fixed, so the actor optimizes a well-defined empirical clipped surrogate that uses behavior-return advantages to transfer selection-guided experience into the base policy. The weights evaluate continuation under μ_n, which can differ from continuation under the base policy even with an exact critic. As shown above, selection induces a bounded local reweighting of the per-action surrogate. This characterization does not establish equivalence between the expected surrogate gradient and the gradient of the standalone actor's expected return. We use the surrogate to transfer selection-guided experience into the base policy and assess its effect on standalone performance empirically. Clipping limits the incentive for large probability-ratio changes on sampled turns without imposing a strict trust-region constraint; improvement in standalone base-policy return is assessed empirically.

D PRESERVATION OF TURN-LEVEL ADVANTAGE-ESTIMATION GUARANTEES

This appendix establishes that learning a categorical turn-level critic can preserve the form and statistical order of the scalar fitted-policy-evaluation guarantee of Zhou et al. (2024). We first control the Bellman regression error of the critic's mean and then apply scalar error propagation. The guarantee concerns the mean-induced advantage; the mean–dispersion action-selection rule is evaluated empirically. The dispersion term is a behavior-shaping signal that reads outcome-relevant branching under the current policy.

D.1 NOTATION AND ASSUMPTIONS

Consider a discounted turn-level MDP with fixed stationary policy π, initial state distribution μ_0, and discount $\gamma \in (0,1)$. Write $P^\pi f(x,a) = \mathbb{E}[f(x',a') \mid x,a]$, where the expectation uses the environment transition and $a' \sim \pi(\cdot \mid x')$. The scalar Bellman operator and normalized discounted occupancy measure are

$$\mathcal{T}^\pi f(x,a) = \mathbb{E}[r + \gamma f(x',a') \mid x,a], \qquad d^\pi = (1-\gamma)\sum_{t=0}^{\infty} \gamma^t \mathcal{L}_\pi(x_t,a_t).$$

Here $(x_0,a_0) \sim \mu_0\pi$, and $d^\pi(x,a) = d_X^\pi(x)\pi(a \mid x)$, with d_X^π the state marginal. The true advantage is $A^\pi(x,a) = Q^\pi(x,a) - \mathbb{E}_{a'\sim\pi(\cdot|x)}Q^\pi(x,a')$.

Assumption 1 (Bounded Returns and Mean-Preserving Projection) *Rewards satisfy $0 \le r \le 1$. Fix $R \ge (1-\gamma)^{-1}$ and $J \ge 2$ ordered atoms $0 = z_1 < \cdots < z_J = R$. For $y \in [z_j, z_{j+1}]$, the categorical projection $\Phi\delta_y$ assigns probabilities $(z_{j+1}-y)/(z_{j+1}-z_j)$ and $(y-z_j)/(z_{j+1}-z_j)$ to z_j and z_{j+1}, respectively, and zero to other atoms. Extend Φ to distributions by mixing these projected point masses. No boundary clipping is performed.*

Let $\mathcal{P}$ be a nonempty finite class of functions from state-action pairs to the probability simplex Δ_J. Define

$$Z_p(x,a) = \sum_{j=1}^{J} p_j(x,a)\delta_{z_j}, \qquad q_p(x,a) = \sum_{j=1}^{J} z_j p_j(x,a), \qquad p \in \mathcal{P}.$$

Here Z_p is a probability measure and q_p its mean. We use J for the number of atoms and M for the samples per iteration. Assumption 1 ensures $r + \gamma z_j \in [0,R]$ and $0 \le Q^\pi \le R$.

For a transition (x,a,r,x') and $a' \sim \pi(\cdot \mid x')$, define the random categorical target and its conditional mean:

$$w_p = \sum_{j=1}^{J} p_j(x',a')\Phi\delta_{r+\gamma z_j}, \qquad (\mathcal{T}_\Phi^\pi p)(x,a) = \mathbb{E}[w_p \mid x,a]. \tag{32}$$

Thus $\mathcal{T}_\Phi^\pi p$ is the probability vector of the projected distributional Bellman backup, including randomness in rewards, transitions, and the next action.

Assumption 2 (Projected Distributional Bellman Completeness) *For the fixed policy π, every $p \in \mathcal{P}$ satisfies $\mathcal{T}_\Phi^\pi p \in \mathcal{P}$ as a function on the state-action space.*

This is exact realizability of the projected categorical target. It does not require a fixed atom set to represent the unprojected Bellman distribution.

Assumption 3 (Density Ratio Coefficient) *There exists a finite constant $C_{\nu,\pi}$ such that the offline state-action distribution ν satisfies*

$$d^\pi(x,a) \le C_{\nu,\pi}\nu(x,a). \tag{33}$$

For general spaces, this means that the Radon–Nikodym derivative of d^π with respect to ν is bounded by $C_{\nu,\pi}$ almost everywhere.

Assumption 4 (Sampling and Empirical Risk Minimization) *The finite class $\mathcal{P}$ is fixed independently of the training data. Each of K independent datasets contains M independent transitions with $(x, a) \sim \nu$ and (r, x') drawn from the MDP conditional on (x, a). Additional action samples are independent across observations conditional on their next states. Each iteration computes an exact empirical cross-entropy minimizer over $\mathcal{P}$.*

The cardinality $|\mathcal{P}|$ counts candidate functions, not network parameters or architectures. Extending this finite-class result to a continuously parameterized network requires a corresponding function-class complexity bound.

D.2 Categorical Fitted Policy Evaluation

For $w, p \in \Delta_J$, let $\mathrm{CE}(w, p) = -\sum_j w_j \log p_j$, with $0 \log 0 = 0$ and positive mass on zero predicted probability giving infinite loss. Minimizing this loss is equivalent to minimizing $D_{\mathrm{KL}}(w \| p)$ because target entropy is independent of p.

Algorithm 3 Categorical Distributional Fitted Policy Evaluation (DFPE)

Require: Fixed policy π, class $\mathcal{P}$, atoms $z_{1:J}$, projection Φ, iterations K
Require: Independent datasets $\mathcal{D}_{1:K}$, each with M transitions as in Assumption 4
Initialize $p_0 \in \mathcal{P}$ independently of the datasets.
for $k = 1$ **to** K **do**
  for each $(x_i, a_i, r_i, x_i') \in \mathcal{D}_k$ **do**
    Sample $a_i' \sim \pi(\cdot \mid x_i')$.
    Set $w_{k,i} = \sum_{j=1}^{J} p_{k-1,j}(x_i', a_i') \Phi \delta_{r_i + \gamma z_j}$.
  end for
  Compute $p_k \in \arg\min_{p \in \mathcal{P}} \sum_{i=1}^{M} \mathrm{CE}(w_{k,i}, p(x_i, a_i))$.
end for
return $\bar{p} = K^{-1} \sum_{k=1}^{K} p_k$, $\bar{Z} = Z_{\bar{p}}$, and $\bar{Q} = q_{\bar{p}}$.

The output $\bar{Z} = K^{-1} \sum_k Z_{p_k}$ is a mixture of probability measures, and $\bar{Q} = K^{-1} \sum_k q_{p_k}$. The mixture need not belong to $\mathcal{P}$. Alternatively, sample $z' \sim Z_{p_{k-1}}(x_i', a_i')$ and use $w_{k,i} = \Phi \delta_{r_i + \gamma z'}$. The analysis applies to either target construction because both have conditional expectation $\mathcal{T}_\Phi^\pi p_{k-1}$.

D.3 From Categorical Learning to Mean Bellman Error

Lemma 1 (Mean Preservation) *Under Assumption 1, for every $p \in \mathcal{P}$,*

$$q_{\mathcal{T}_\Phi^\pi p} = \mathcal{T}^\pi q_p. \tag{34}$$

Consequently, Assumption 2 implies scalar Bellman completeness of $\{q_p : p \in \mathcal{P}\}$.

Proof. Linear interpolation gives $\sum_j z_j (\Phi \delta_y)_j = y$ for every $y \in [0, R]$. Applying this identity to equation 32 yields

$$q_{\mathcal{T}_\Phi^\pi p}(x, a) = \mathbb{E}\left[\sum_j p_j(x', a')(r + \gamma z_j) \,\middle|\, x, a \right] = \mathbb{E}[r + \gamma q_p(x', a') \mid x, a].$$

The last assertion follows by taking means of the representative of $\mathcal{T}_\Phi^\pi p$ in $\mathcal{P}$. □

Lemma 2 (Categorical Regression Controls the Mean) *Let $(u_i, w_i)_{i=1}^M$ be independent identically distributed observations with $u_i \sim \nu$, $w_i \in \Delta_J$, and $p^*(u) = \mathbb{E}[w_i \mid u_i = u]$. Suppose $p^* \in \mathcal{P}$, where $\mathcal{P}$ is a fixed finite class. If $\widehat{p}$ minimizes $\sum_i \mathrm{CE}(w_i, p(u_i))$ over $\mathcal{P}$, then, for any $\eta \in (0, 1)$, with probability at least $1 - \eta$,*

$$\mathbb{E}_\nu[(q_{\widehat{p}} - q_{p^*})^2] \le \frac{2R^2 \log(|\mathcal{P}|/\eta)}{M}. \tag{35}$$

Proof. Use the squared Hellinger distance convention

$$H^2(p,p^*)=\sum_j(\sqrt{p_j}-\sqrt{p_j^*})^2, \qquad h_p^2=\mathbb{E}_\nu H^2(p(u),p^*(u)).$$

For a fixed candidate p, write

$$D_i(p)=\mathrm{CE}(w_i,p^*(u_i))-\mathrm{CE}(w_i,p(u_i))=\sum_{j:p_j^*(u_i)>0} w_{i,j}\log\frac{p_j(u_i)}{p_j^*(u_i)}.$$

If $p_j^*(u)=0$, nonnegativity of $w_{i,j}$ implies $w_{i,j}=0$ almost surely conditional on $u_i=u$. Thus the restriction loses no target mass. The comparator loss is finite almost surely; a candidate assigning zero probability to positive target mass has $D_i=-\infty$ and $\exp(D_i/2)=0$. By convexity of the exponential, including these cases by continuity,

$$\exp(D_i(p)/2)\le\sum_{j:p_j^*(u_i)>0} w_{i,j}\sqrt{\frac{p_j(u_i)}{p_j^*(u_i)}}.$$

Taking conditional expectations and then averaging over u_i gives

$$\mathbb{E}\exp(D_i(p)/2)\le\mathbb{E}_\nu\sum_j\sqrt{p_j(u)p_j^*(u)}=1-\frac{h_p^2}{2}\le\exp(-h_p^2/2).$$

Independence and Markov's inequality imply

$$\Pr\left\{\sum_{i=1}^M D_i(p)\ge 0\right\}\le\mathbb{E}\exp\left(\frac{1}{2}\sum_{i=1}^M D_i(p)\right)\le\exp(-Mh_p^2/2).$$

A union bound over candidates with $h_p^2>2\log(|\mathcal{P}|/\eta)/M$ shows that, with probability at least $1-\eta$, none can have empirical loss at most that of p^*. Since $\widehat{p}$ does, its $h_{\widehat{p}}^2$ is at most this threshold.

For probability vectors on atoms in $[0,R]$,

$$|q_p-q_{p^*}|\le R\,\mathrm{TV}(p,p^*), \qquad \mathrm{TV}(p,p^*)=\tfrac{1}{2}\sum_j|p_j-p_j^*|\le H(p,p^*).$$

The second inequality follows by factoring $p_j-p_j^*=(\sqrt{p_j}-\sqrt{p_j^*})(\sqrt{p_j}+\sqrt{p_j^*})$ and using Cauchy–Schwarz with $\sum_j(\sqrt{p_j}+\sqrt{p_j^*})^2\le 4$. Squaring and integrating proves equation 35. No uniform positive lower bound on predicted probabilities is needed. □

Lemma 3 (Mean Bellman Residual of Categorical DFPE) *Under Assumptions 1–4, for any $\delta\in(0,1)$, Algorithm 3 satisfies, with probability at least $1-\delta$,*

$$\mathbb{E}_{d^\pi}[(\bar{Q}-\mathcal{T}^\pi\bar{Q})^2]\le\varepsilon_{\mathrm{stat}},$$
$$\varepsilon_{\mathrm{stat}}=C_{\nu,\pi}R^2\left(\frac{1}{K}+\frac{2\log(|\mathcal{P}|K/\delta)}{M}\right). \tag{36}$$

Proof. Condition on all observations and algorithmic randomness before iteration k. Then p_{k-1} is fixed and the current observations are independent of that history. The conditional target mean is $p_k^*=\mathcal{T}_\Phi^\pi p_{k-1}\in\mathcal{P}$ by completeness. Lemmas 1 and 2, with $\eta=\delta/K$, imply

$$\mathbb{E}_\nu[(q_{p_k}-\mathcal{T}^\pi q_{p_{k-1}})^2]\le e_M:=\frac{2R^2\log(|\mathcal{P}|K/\delta)}{M}. \tag{37}$$

The conditional failure probability is at most δ/K for every history. Averaging over histories and taking a union bound makes equation 37 simultaneous for all k with probability at least $1-\delta$.

On this event, write $q_k=q_{p_k}$. Affinity of the scalar Bellman operator gives

$$\bar{Q}-\mathcal{T}^\pi\bar{Q}=\frac{1}{K}\left(q_1-\mathcal{T}^\pi q_K+\sum_{k=2}^K(q_k-\mathcal{T}^\pi q_{k-1})\right).$$

All q_k and $\mathcal{T}^\pi q_k$ lie in $[0,R]$. Jensen's inequality therefore yields

$$\mathbb{E}_\nu[(\bar{Q}-\mathcal{T}^\pi\bar{Q})^2]\le\frac{1}{K}\left(R^2+(K-1)e_M\right)\le\frac{R^2}{K}+e_M.$$

Changing measure with Assumption 3 proves the claim. □

D.4 Preservation of the Turn-Level Advantage Bound

Theorem 2 (Turn-Level Advantage Bounds under Categorical Critic Learning) *Fix a stationary policy π. Under Assumptions 1–4, run Algorithm 3 for K iterations with M samples per iteration, using $N = MK$ transitions in total. Let $\bar{Z} = Z_{\bar{p}}$, $\bar{Q} = \mathbb{E}[\bar{Z}]$, and*

$$\bar{A}(x,a) = \bar{Q}(x,a) - \mathbb{E}_{a'\sim\pi(\cdot|x)}\bar{Q}(x,a').$$

For every $\delta \in (0,1)$, with probability at least $1-\delta$,

$$\mathbb{E}_{d^\pi}[(\bar{A} - A^\pi)^2] \le \frac{4}{1-\gamma}\left(\varepsilon_{\mathrm{stat}} + 2R\sqrt{\varepsilon_{\mathrm{stat}}}\right), \tag{38}$$

where $\varepsilon_{\mathrm{stat}}$ is given in equation 36. Thus categorical learning preserves the scalar turn-level FPE bound's dependence on the Bellman residual and on the turn-level discount, with its own function-class complexity entering $\varepsilon_{\mathrm{stat}}$.

Proof. Work on the event in Lemma 3. Set

$$e = \bar{Q} - Q^\pi, \qquad \Delta = \mathbb{E}_{d^\pi} e^2, \qquad b = \mathbb{E}_{d^\pi}(\bar{Q} - \mathcal{T}^\pi\bar{Q})^2 \le \varepsilon_{\mathrm{stat}}.$$

The discounted occupancy identity is $d^\pi = (1-\gamma)\mu_0\pi + \gamma d^\pi P^\pi$. Consequently, for every nonnegative g,

$$\mathbb{E}_{d^\pi} P^\pi g \le \gamma^{-1}\mathbb{E}_{d^\pi} g. \tag{39}$$

This identity does not require d^π to be stationary. Since $Q^\pi = \mathcal{T}^\pi Q^\pi$, Jensen's inequality and equation 39 give

$$\mathbb{E}_{d^\pi}(\mathcal{T}^\pi\bar{Q} - Q^\pi)^2 = \gamma^2\mathbb{E}_{d^\pi}(P^\pi e)^2 \le \gamma^2\mathbb{E}_{d^\pi} P^\pi(e^2) \le \gamma\Delta.$$

Expanding $e = (\bar{Q} - \mathcal{T}^\pi\bar{Q}) + (\mathcal{T}^\pi\bar{Q} - Q^\pi)$ and applying Cauchy–Schwarz yields

$$\Delta \le b + 2\sqrt{b\gamma\Delta} + \gamma\Delta.$$

Both $\bar{Q}$ and Q^π lie in $[0,R]$, so $\sqrt{\gamma\Delta} \le R$. Rearranging and using monotonicity in b gives

$$\Delta \le \frac{b + 2R\sqrt{b}}{1-\gamma} \le \frac{\varepsilon_{\mathrm{stat}} + 2R\sqrt{\varepsilon_{\mathrm{stat}}}}{1-\gamma}. \tag{40}$$

Finally,

$$\bar{A}(x,a) - A^\pi(x,a) = e(x,a) - \mathbb{E}_{a'\sim\pi(\cdot|x)} e(x,a').$$

Using $(u-v)^2 \le 2u^2 + 2v^2$ and conditional Jensen gives $\mathbb{E}_{d^\pi}(\bar{A} - A^\pi)^2 \le 4\Delta$, proving equation 38. In fact, the conditional-variance identity gives the stronger inequality

$$\mathbb{E}_{d^\pi}(\bar{A} - A^\pi)^2 = \mathbb{E}_{x\sim d_X^\pi}\,\mathrm{Var}_{a\sim\pi(\cdot|x)}\, e(x,a) \le \Delta.$$

We retain the factor 4 in the statement to display the same form as the scalar FPE comparison. □

Statistical order. The residual bound has the same $K^{-1} + \log(|\mathcal{P}|K/\delta)/M$ structure as finite-class scalar FPE. With a balanced choice of K and M under a total sample budget, it has order $C_{\nu,\pi}R^2\sqrt{\log(|\mathcal{P}|N/\delta)/N}$, up to universal constants and integer rounding. Preservation refers to this order and the turn-level error-propagation factor; it does not assert identical numerical bounds for distributional and scalar classes of different complexity.

D.5 Conditional Comparison with Token-Level Evaluation

Corollary 1 (Preservation of the Temporal-Scale Factor) *Consider a token-level representation with $L_{\max} \ge 1$ token steps per turn, using padding when necessary, discount $\beta = \gamma^{1/L_{\max}}$, and reward timing and scaling chosen to express returns in the same units. Let $\widetilde{Q}$ be a separately learned token-level critic on prefix–token pairs, with occupancy $\widetilde{d}^\pi$, Bellman operator $\widetilde{\mathcal{T}}^\pi$, and mean-centered advantage $\widetilde{A}$. Suppose its true Q-function and estimate, and their turn-level counterparts, lie in a common interval $[0,R]$. If their respective high-probability scalar Bellman residual*

bounds are $\varepsilon_{\mathrm{tok}}$ and $\varepsilon_{\mathrm{turn}}$, then their advantage mean-squared errors under their respective occupancy measures are bounded by

$$B_{\mathrm{turn}} = \frac{4}{1-\gamma}\left(\varepsilon_{\mathrm{turn}} + 2R\sqrt{\varepsilon_{\mathrm{turn}}}\right), \qquad B_{\mathrm{tok}} = \frac{4}{1-\beta}\left(\varepsilon_{\mathrm{tok}} + 2R\sqrt{\varepsilon_{\mathrm{tok}}}\right). \tag{41}$$

If both residuals are bounded by the same positive budget ε, the bounds evaluated at that common budget satisfy

$$\frac{B_{\mathrm{turn}}(\varepsilon)}{B_{\mathrm{tok}}(\varepsilon)} = \frac{1-\beta}{1-\gamma} = \frac{1}{\sum_{i=0}^{L_{\max}-1}\beta^i} \le \frac{1}{L_{\max}\gamma}. \tag{42}$$

For $L_{\max} > 1$, the exact ratio is strictly less than one.

For a sample-budget comparison, let N denote the number of turn-level transitions. Suppose instead that, for a common positive reference budget ε_N, the residual guarantees satisfy

$$\varepsilon_{\mathrm{turn}} \le \varepsilon_N, \qquad \varepsilon_{\mathrm{tok}} \le \frac{\varepsilon_N}{\sqrt{L_{\max}}}. \tag{43}$$

Define the corresponding upper bounds evaluated at these reference budgets by

$$B_{\mathrm{turn},N} = \frac{4}{1-\gamma}\left(\varepsilon_N + 2R\sqrt{\varepsilon_N}\right), \qquad B_{\mathrm{tok},N} = \frac{4}{(1-\beta)\sqrt{L_{\max}}}\left(\varepsilon_N + 2RL_{\max}^{1/4}\sqrt{\varepsilon_N}\right). \tag{44}$$

Then

$$\frac{B_{\mathrm{turn},N}}{B_{\mathrm{tok},N}} = \frac{1-\beta}{1-\gamma}\sqrt{L_{\max}}\,\frac{\varepsilon_N + 2R\sqrt{\varepsilon_N}}{\varepsilon_N + 2RL_{\max}^{1/4}\sqrt{\varepsilon_N}} \le \frac{1}{\gamma\sqrt{L_{\max}}}. \tag{45}$$

Thus, when $\gamma\sqrt{L_{\max}} > 1$, the turn-level reference upper bound is smaller by a factor of at least $\gamma\sqrt{L_{\max}}$, recovering the comparison factor of Zhou et al. (2024) under equation 43. Both comparisons concern the displayed upper bounds, not ratios of realized estimation errors or of the tightest available bounds.

Proof. The scalar error-propagation part of Theorem 2 uses only a Bellman residual bound, a common value bound, and the discounted occupancy identity. Apply it separately with discounts γ and β. For a simultaneous guarantee with failure probability at most δ, obtain each residual guarantee at failure probability $\delta/2$ and take a union bound. At a common budget, cancellation gives equation 42, where $\sum_{i=0}^{L_{\max}-1}\beta^i \ge L_{\max}\beta^{L_{\max}-1} \ge L_{\max}\gamma$. Under equation 43, monotonicity of $u + 2R\sqrt{u}$ gives equation 44. Dividing these bounds yields the equality in equation 45. Its inequality follows from equation 42 and $L_{\max}^{1/4} \ge 1$. □

The padded construction requires an explicit reward-timing convention. For example, if a turn reward r is paid on the final token step, using token reward $\beta^{-(L_{\max}-1)}r$ makes the discounted return at turn boundaries equal to the original turn return. Intermediate prefix values can have a different range, which must be included in the common bound R used in the comparison.

When the sample-budget condition holds. Theorem 2 supplies $\varepsilon_{\mathrm{turn}}$ for categorical DFPE. One sufficient setting for equation 43 is that the turn-level and token-level learners have N and $L_{\max}N$ independent effective observations, respectively, and each satisfies the corresponding completeness and sampling assumptions. Their residual guarantees must admit a common envelope $A\sqrt{\ell/n}$ at sample size n, after separately balancing their iteration and batch sizes. Here A bounds the coverage and value-range constants, and ℓ is a common upper bound on both function-class logarithmic terms at failure probability $\delta/2$, with universal constants and integer rounding absorbed into A. Taking $\varepsilon_N = A\sqrt{\ell/N}$ then gives equation 43. This is an effective-sample-budget comparison: splitting the same N turn transitions into $L_{\max}N$ token transitions does not by itself yield $L_{\max}N$ independent observations. Applying the factor to such a shared dataset requires a separate analysis establishing the token-level residual rate despite within-turn dependence. Without that analysis or the stated residual condition, only the conditional comparison in equation 41 is justified.

E EXPERIMENT DETAILS

E.1 BASELINE DETAILS

In the following part, we introduce the decision modules of several baselines used in our paper:

PPO (Proximal Policy Optimization) (Schulman et al., 2017) is an on-policy actor-critic algorithm widely used for fine-tuning LLMs, which constrains the policy change via a clipped surrogate objective. Its main updates are:

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta)\hat{A}_t,\ \text{clip}\left(r_t(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_t \right) \right] \tag{46}$$

where $r_t(\theta) = \frac{\pi_\theta(a_t|x_t)}{\pi_{\theta_{\text{old}}}(a_t|x_t)}$, and $\hat{A}_t$ is the token-level advantage estimated via GAE:

$$\hat{A}_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}, \quad \delta_t = r_t + \gamma V(x_{t+1}) - V(x_t) \tag{47}$$

TurnPPO is the turn-level PPO variant we design as the direct precursor of DCGPO, i.e., the baseline before any improvement proposed by DCGPO. Unlike standard PPO, which maintains a token-level scalar critic, TurnPPO maintains a scalar turn-level critic V_ϕ that estimates the value of each response (turn), and the advantage estimated by this critic is shared by all tokens within the same response. To stabilize the update, TurnPPO adopts the same sequence-level importance-ratio correction as GSPO (Zheng et al., 2025), using a length-normalized turn-level importance ratio shared by all tokens of the turn, exactly as in DCGPO. TurnPPO can therefore be viewed as a turn-level distributional critic without the distributional part. Its critic is updated by minimizing the squared error against the GAE λ-return:

$$J_V(\phi) = \mathbb{E}_{(x_t,a_t)\sim\mathcal{D}} \left[(G_t^\lambda - V_\phi(x_t))^2 \right], \tag{48}$$

where G_t^λ is the GAE λ-return at turn t, and the turn-level advantage is $\hat{A}_t = G_t^\lambda - V_\phi(x_t)$. With this advantage, the actor is updated with the same turn-level objective as DCGPO,

$$\mathcal{L}_{\text{ACTOR}}(\theta) = -\mathbb{E}_{\tau\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{\sum_{t=0}^{T-1} L_t} \sum_{t=0}^{T-1} \sum_{i=1}^{L_t} \min\Big(\rho_t(\theta)\,\hat{A}_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t \Big) \right], \tag{49}$$

where $\rho_t(\theta) = \left(\frac{\pi_\theta(a_t|x_t)}{\pi_{\theta_{\text{old}}}(a_t|x_t)} \right)^{1/L_t}$ is the length-normalized turn-level importance ratio shared by all L_t tokens of the turn.

GRPO (Group Relative Policy Optimization) (Shao et al., 2024) is a critic-free, memory-efficient RL algorithm for LLMs that eliminates the need for a separate value network. For advantage estimation, GRPO samples a group of G responses for each query from the old policy, computes their rewards, and then normalizes these rewards within the group to obtain relative advantages. The group-relative advantage estimation and policy optimization objective are:

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})} \tag{50}$$

$$\begin{aligned} J_{\text{GRPO}}(\theta) =& \mathbb{E}_{x\sim\mathcal{D},\{a_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(a|x)} \left[\frac{1}{G} \sum_{i=1}^{G} \min\left(\rho_\theta(a_i|x)A_i, \text{clip}\left(\rho_\theta(a_i|x), 1-\epsilon, 1+\epsilon\right) A_i \right) \right] \\ & - \beta\, \mathbb{D}_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right) \end{aligned} \tag{51}$$

where $\rho_\theta(a_i|x) = \frac{\pi_\theta(a_i|x)}{\pi_{\theta_{\text{old}}}(a_i|x)}$.

GSPO (Group Sequence Policy Optimization) (Zheng et al., 2025) is a critic-free RL algorithm for training large language models. Unlike GRPO, which applies token-level importance ratios that introduce high-variance noise and cause training collapse—especially in MoE models—GSPO

adopts a theoretically grounded sequence-level importance ratio and performs clipping, rewarding, and optimization directly at the sequence level, ensuring full alignment with sequence-level rewards. The core objective is:

$$J_{GSPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},(a_i)_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(\rho_i(\theta)\hat{A}_i, \text{clip}(\rho_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right)\right], \quad (52)$$

where the sequence-level importance ratio is length-normalized as:

$$\rho_i(\theta) = \left(\frac{\pi_\theta(a_i|x)}{\pi_{\theta_{\text{old}}}(a_i|x)}\right)^{1/|a_i|}. \quad (53)$$

GiGPO (Group-in-Group Policy Optimization) (Feng et al., 2026) is a critic-free RL algorithm for training long-horizon LLM agents. GiGPO uses a combined advantage function: it computes macro episode-level advantages and micro step-level advantages by retroactively grouping actions that share the same environment state across trajectories:

$$\begin{aligned} A^S(a_t^{(i)}) &= \frac{R_t^{(i)} - \text{mean}\left(\left\{R_t^{(j)} \mid (a_t^{(j)}, R_t^{(j)}) \in G^S(\tilde{x})\right\}\right)}{F_{\text{norm}}\left(\left\{R_t^{(j)} \mid (a_t^{(j)}, R_t^{(j)}) \in G^S(\tilde{x})\right\}\right)}, \\ A(a_t^{(i)}) &= A^E(\tau_i) + \omega \cdot A^S(a_t^{(i)}) \end{aligned} \quad (54)$$

where A^E and A^S denote episode-level relative advantage and turn-level relative advantages, respectively, ω is a weighting coefficient that balances the two advantage components, and $G^S(\tilde{x})$ is the turn-level group constructed for each unique anchor state $\tilde{x}$.

E.2 Training Details

Hyperparameters for WebShop The experimental setup in our work remains consistent with that in GiGPO and other related works (Feng et al., 2026), and all methods adopt the same parameter settings. The maximum prompt and response lengths are capped at 4096 and 512 tokens, respectively, and each episode allows up to 15 environment steps. The actor learning rate is 1e-6 and the critic's is 1e-5, with the critic rate applied only in PPO, TurnPPO, and DCGPO, while other methods use only the actor rate. Reward design follows a rule-based scheme: +10 for success, 0 for failure, and a penalty of -0.1 for any invalid action. In all group-based RL methods, we employ a group size of 8 and sample 16 distinct groups per rollout, yielding 16 × 8 = 128 parallel environments. The rollout temperature is 1.0, while validation uses 0.4; the mini-batch size is 64, and the KL-divergence loss coefficient is 0.01. For GiGPO specifically, the weighting coefficient ω is fixed at 1.0 following vanilla setting (Feng et al., 2026). For methods that use turn-level advantage computation, such as DCGPO, TurnPPO and GiGPO, we set the turn-level γ to 0.95, while for token-level methods, γ is set to 1.0.

Hyperparameters for ALFWorld The experimental setup in our work remains consistent with that in GiGPO and other related works (Feng et al., 2026), and all methods adopt the same parameter settings. The maximum prompt and response lengths are capped at 2048 and 512 tokens, respectively, and each episode allows up to 50 environment steps. The actor learning rate is 1e-6 and the critic's is 1e-5, with the critic rate applied only in PPO, TurnPPO, and DCGPO, while other methods use only the actor rate. Reward design follows a rule-based scheme: +10 for success, 0 for failure, and a penalty of -0.1 for any invalid action. In all group-based RL methods, we employ a group size of 8 and sample 16 distinct groups per rollout, yielding 16 × 8 = 128 parallel environments. The rollout temperature is 1.0, while validation uses 0.4; the mini-batch size is 256, and the KL-divergence loss coefficient is 0.01. For GiGPO specifically, the weighting coefficient ω is fixed at 1.0 following vanilla setting (Feng et al., 2026). For methods that use turn-level advantage computation, such as DCGPO, TurnPPO and GiGPO, we set the turn-level γ to 0.95, while for token-level methods, γ is set to 1.0.

Table 2: Inference settings for final evaluation. All methods use the same settings within each benchmark and backbone. Evaluation seeds specify repeated rollouts of a fixed checkpoint and are distinct from independent training seeds.

Setting	ALFWorld	WebShop
Evaluation tasks	140 seen + 134 unseen	500 goals
Evaluation seeds	$\{0, 1, 2\}$	$\{0\}$
Episodes per checkpoint	822	500
Maximum interaction turns	50	15
History window (turns)	2	2
Maximum prompt length (tokens)	2048	4096
Maximum response length (tokens)	512	512
Sampling temperature	0.4	0.4
Top-p	1.0	1.0
Top-k filtering	Disabled	Disabled
Responses per turn	1	1

E.3 Evaluation Protocol

Checkpoints and inference. The following protocol applies to the final performance comparisons. For each method and backbone, we evaluate the actor checkpoints at training step 200 from three independent training runs with different seeds. Evaluation uses the trained actor directly, without loading a critic or performing candidate selection. At each turn, the actor samples a single response using the inference settings in Table 2. We retain the training environment interface, prompt format, and action parsing procedure. For Qwen3 models, native chat-template thinking is disabled, while the task-specific reasoning instructions are retained.

ALFWorld. We evaluate the text-based environment using the `eval_in_distribution` and `eval_out_of_distribution` splits, referred to as *seen* and *unseen*, respectively. These are held-out task splits corresponding to seen and unseen layouts. Using the `json_2.1.1` task files and the environment's task filters, we retain 140 seen and 134 unseen games across the six supported task types. The filters exclude unsupported task variants, including movable-receptacle and sliced-object tasks, as well as tasks without valid, solvable game files. Each game is evaluated once for each of the three evaluation seeds, yielding 822 episodes per checkpoint. Success is determined by the environment's task-completion signal. We report success rates separately for the two splits and for individual task types. The overall success rate within each split is computed over all episodes, weighting task types by their numbers of games. Failed and timed-out episodes remain in the denominator.

WebShop. We use the 1,000-product configuration with synthetic goals. The goal generation and ordering seed is fixed. Following Feng et al. (2026), we use goals with indices 0–499 form the held-out evaluation pool, and the remaining goals form the training pool. Training-time validation samples from this same held-out pool; final evaluation exhaustively evaluates all 500 goals once per checkpoint, using evaluation seed 0. An episode is successful if it terminates with a purchase whose original WebShop reward equals 1. We report the percentage of successful episodes and the average original task reward multiplied by 100, denoted as *Score*. Partial-match purchases retain their original rewards when computing Score. Episodes that reach the interaction limit without completing a purchase receive zero score and count as failures. All 500 episodes are included in both metrics. Training-time reward scaling and invalid-action penalties are excluded from the reported evaluation metrics.

Aggregation across seeds. For ALFWorld, we first average each metric over the three evaluation seeds within each checkpoint. For WebShop, each checkpoint contributes its metrics from the single complete 500-goal evaluation. We then compute the mean and sample standard deviation across the three independent training seeds. Thus, the reported variation measures differences across independently trained policies; repeated evaluations of the same checkpoint are averaged before this aggregation. Final evaluation uses the complete task sets described above, whereas training-time validation uses sampled episodes.

E.4 PROMPTS

Prompt Template for WebShop

You are browsing an online shop. Based on the instruction, buy a product that is close to the requested description. You should only choose an action from the available actions listed below. Valid actions are ‘search[...]‘ and ‘click[...]‘.
Task: {task_description}
Prior to this turn, you have already taken {step_count} turn(s). Below are the most recent {history_length} observations and action sequences: {action_history}
You are now at turn {current_step}. Current observation: {current_observation}
Available actions: [{available_actions}]
Respond with exactly one action: <think>...</think><answer>click[buy now]</answer>

Figure 5: The prompt template of WebShop agents.

Prompt Template for ALFWorld

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: {task_description} Prior to this step, you have already taken {step_count} step(s). Below are the most recent {history_length} observations and the corresponding actions you took: {action_history} You are now at step {current_step} and your current observation is: {current_observation} Your admissible actions of the current situation are: [{admissible_actions}].
Now it’s your turn to take an action. You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <think></think>tags. Once you’ve finished your reasoning, you should choose an admissible action for current step and present it within <action></action>tags.

Figure 6: The prompt template of ALFWorld agents.

F ADDITIONAL RESULTS

F.1 EXPLORATION METRICS

Both the preliminary study (Section 3) and the exploration analysis (Section 5) share the same state-level indicators, which we define here once and reuse throughout the paper. Let N_{interact} be the number of training environment interactions in an update, N_{new} the number of interactions that land in a state first encountered relative to the cumulative training visitation set, and N_{revisit} the number of interactions that return to a state already visited within the same episode. State keys combine task identity with the observable page, search field, and clickable items, excluding dialogue history; initial states enter the novelty count, while resets are not counted as interactions. The new-state rate NR, episode revisit rate RR, and validation success rate SR are

$$NR = \frac{N_{\text{new}}}{N_{\text{interact}}}, \qquad RR = \frac{N_{\text{revisit}}}{N_{\text{interact}}}, \qquad SR = \frac{N_{\text{success}}}{N_{\text{episode}}}, \tag{55}$$

where N_{success} and N_{episode} count the successful and total validation episodes, respectively. Thus NR measures observable-state novelty, RR captures wasteful revisits within an episode, and SR measures task success. Higher NR and lower RR indicate more efficient exploration.

Table 3: Exploration efficiency on WebShop, measured by revisit rate (RR), new-state rate (NR), and actor-only validation success rate (SR). Results are averaged over 3 random seeds (mean $\pm$ std at step 150).

Method	RR	NR	SR
PPO	57.4 ± 2.3	0.107 ± 0.011	67.2 ± 7.5
PPO (w/o entropy)	56.5 ± 4.2	0.124 ± 0.035	67.2 ± 5.6
DCGPO (β=0)	46.5 ± 3.0	0.169 ± 0.042	75.0 ± 4.8
DCGPO (K=1)	48.4 ± 0.6	0.208 ± 0.002	73.2 ± 3.6
DCGPO	43.7 ± 2.8	0.234 ± 0.024	79.9 ± 1.2

F.2 Preliminary Study

Stochastic generation and entropy regularization encourage diversity in model outputs, but do they also ensure effective exploration of the environment? To examine this question, we conduct a preliminary study on WebShop (Yao et al., 2022) using Qwen3-4B, comparing ***PPO*** with a token-level entropy coefficient of 0.001 against ***PPO w/o entropy***, which differs only in setting this coefficient to 0. Both configurations share the same environment, prompts, and task sampling setup. We analyze the first 150 training updates over three seeds, with 16 task groups and 8 rollouts per group, totaling 128 episodes per update. State-visitation statistics are collected during training, while the actor is evaluated directly on a fixed set of 128 validation tasks every five updates; the resulting training curves are shown in Figure 2. We measure the new-state rate NR, the episode revisit rate RR, and the validation success rate SR (Appendix F.1), together with the token entropy H_{token} of Equation 4, which measures conditional token entropy at visited contexts and generated prefixes rather than entropy over environment actions.

Generation-side randomness alone does not ensure effective exploration. As shown in Table 3, PPO improves task success while continuing to revisit states frequently: 57.4% of training interactions return to previously visited states within the same episode. Adding the entropy bonus yields slightly higher mean token entropy, but neither a higher new-state rate nor a lower revisit rate. The key insight is a mismatch between the level at which diversity is encouraged and the level at which exploration matters: token entropy rewards uncertainty in generation without distinguishing responses by the states or returns they may lead to. Different responses can induce the same environment action, while visiting a new state need not advance the task. Generation therefore supplies possible behaviors, but does not by itself assess which candidate is worth executing.

Distribution-guided selection improves exploration and the learned policy. The exploration analysis in Section 5 extends this study by adding DCGPO and its selection ablations to Figure 2(b)–(d) under the same protocol and metrics. Table 3 reports the revisit rate, new-state rate, and validation success rate at step 150. Relative to PPO and PPO w/o entropy, DCGPO roughly doubles the new-state rate and reduces episode revisits from 57.4% to 43.7%, while attaining higher SR. Compared with the $K = 1$ variant, full DCGPO achieves a higher NR, fewer episode revisits, and a 6.7 percentage-point gain in SR; since validation disables candidate selection, this gain reflects a stronger learned actor. Fixing $K = 3$, the $\beta = 0$ variant selects candidates solely by predicted mean return; adding the dispersion term raises NR from 16.9% to 23.4%, increases SR by 4.9 percentage points, and lowers RR. These results indicate that dispersion-guided selection sustains state novelty beyond mean-greedy selection, with an accompanying policy benefit, supporting the use of downstream return information to allocate training interactions.

F.3 Performance

We present the empirical results in Tables 4 and 5, which extend the main-text comparison to all four base models. Below we discuss the two environments in turn and then summarize the conclusions drawn from the wider model range.

ALFWorld. The pattern reported in the main text generalizes to all four base models: DCGPO attains the best overall success rate on both the seen and unseen splits for every base model, and

Table 4: ALFWorld success rates (%) on the seen and unseen splits across four base models. Results are averaged over 3 random seeds. Bold entries denote the best method for each base model on each split.

Split	Method	Success rate (%)						
		Pick	Look	Clean	Heat	Cool	Pick2	All
Seen	*Base: Qwen2.5-1.5B-Instruct*							
	PPO	89.2 ± 2.4	56.4 ± 8.9	81.5 ± 11.0	88.2 ± 2.4	79.1 ± 6.0	69.0 ± 6.3	79.3 ± 3.4
	TurnPPO	78.7 ± 4.7	59.8 ± 3.9	67.1 ± 1.4	55.6 ± 6.4	56.4 ± 1.5	49.1 ± 2.1	63.0 ± 0.1
	GRPO	86.0 ± 6.2	67.5 ± 6.5	84.8 ± 3.1	81.9 ± 4.8	80.0 ± 5.3	70.4 ± 8.1	79.8 ± 4.8
	GSPO	71.1 ± 3.9	54.7 ± 5.3	69.1 ± 6.4	61.8 ± 22.3	67.1 ± 13.7	46.8 ± 8.4	63.3 ± 7.7
	GiGPO	96.2 ± 1.0	82.9 ± 3.9	95.1 ± 1.2	$\mathbf{99.3 \pm 1.2}$	89.3 ± 3.5	$\mathbf{86.1 \pm 4.8}$	92.1 ± 1.9
	Ours	$\mathbf{98.4 \pm 2.0}$	$\mathbf{89.7 \pm 4.4}$	$\mathbf{100.0 \pm 0.0}$	93.8 ± 0.0	$\mathbf{94.7 \pm 2.7}$	84.3 ± 7.1	$\mathbf{94.3 \pm 1.2}$
	Base: Qwen2.5-7B-Instruct							
	PPO	93.0 ± 2.4	69.2 ± 6.8	92.2 ± 2.9	79.9 ± 4.8	72.9 ± 8.9	75.0 ± 5.6	82.5 ± 2.4
	TurnPPO	90.5 ± 3.4	70.1 ± 7.4	82.3 ± 10.1	67.4 ± 14.2	67.1 ± 0.8	71.3 ± 13.5	76.9 ± 5.6
	GRPO	91.1 ± 0.6	66.7 ± 11.2	85.2 ± 7.5	74.3 ± 3.2	67.6 ± 4.3	62.0 ± 11.6	76.6 ± 2.6
	GSPO	97.5 ± 1.5	69.2 ± 5.1	90.1 ± 10.7	83.3 ± 5.5	75.6 ± 0.8	78.2 ± 7.0	84.6 ± 3.9
	GiGPO	99.1 ± 1.7	87.2 ± 6.8	$\mathbf{97.5 \pm 2.1}$	84.0 ± 8.7	88.9 ± 1.5	93.1 ± 2.4	93.1 ± 0.9
	Ours	$\mathbf{100.0 \pm 0.0}$	$\mathbf{89.7 \pm 6.8}$	97.1 ± 2.6	$\mathbf{95.8 \pm 2.1}$	$\mathbf{93.3 \pm 2.3}$	$\mathbf{94.9 \pm 6.6}$	$\mathbf{96.0 \pm 1.7}$
	Base: Qwen3-1.7B							
	PPO	61.9 ± 9.2	40.2 ± 5.3	56.0 ± 8.2	52.8 ± 7.9	40.4 ± 15.5	27.8 ± 14.5	48.0 ± 9.4
	TurnPPO	62.5 ± 3.1	48.7 ± 2.6	64.2 ± 4.3	58.3 ± 5.5	51.1 ± 10.2	37.0 ± 10.4	54.7 ± 3.3
	GRPO	69.8 ± 5.6	42.7 ± 1.5	46.1 ± 8.0	47.2 ± 4.3	45.3 ± 5.8	33.8 ± 11.7	49.6 ± 1.5
	GSPO	63.8 ± 4.4	42.7 ± 1.5	59.7 ± 13.3	56.9 ± 11.9	48.9 ± 15.5	33.3 ± 10.0	52.4 ± 8.2
	GiGPO	71.4 ± 5.8	47.0 ± 8.2	69.1 ± 3.3	59.7 ± 1.2	64.4 ± 8.5	55.6 ± 10.5	63.4 ± 4.9
	Ours	$\mathbf{93.7 \pm 2.0}$	$\mathbf{82.1 \pm 4.4}$	$\mathbf{91.8 \pm 2.6}$	$\mathbf{79.2 \pm 0.0}$	$\mathbf{84.0 \pm 1.3}$	$\mathbf{88.4 \pm 1.6}$	$\mathbf{87.9 \pm 0.7}$
	Base: Qwen3-4B							
	PPO	87.9 ± 4.8	62.4 ± 13.2	68.7 ± 4.3	65.3 ± 1.2	68.0 ± 11.6	41.7 ± 4.8	67.8 ± 0.4
	TurnPPO	90.2 ± 1.5	70.9 ± 8.2	60.5 ± 3.3	75.7 ± 8.7	64.0 ± 12.0	60.2 ± 5.3	71.2 ± 3.7
	GRPO	94.0 ± 3.6	65.8 ± 3.9	70.4 ± 9.8	59.7 ± 12.0	63.6 ± 4.3	51.4 ± 8.7	70.2 ± 4.3
	GSPO	88.9 ± 4.4	67.5 ± 3.9	72.8 ± 4.5	76.4 ± 6.7	67.6 ± 5.6	63.0 ± 2.9	74.1 ± 3.9
	GiGPO	95.9 ± 2.0	82.1 ± 13.3	94.7 ± 3.1	85.4 ± 9.6	81.8 ± 3.4	75.5 ± 2.1	87.1 ± 4.2
	Ours	$\mathbf{100.0 \pm 0.0}$	$\mathbf{89.7 \pm 6.8}$	$\mathbf{96.3 \pm 0.0}$	$\mathbf{92.4 \pm 6.0}$	$\mathbf{87.1 \pm 5.6}$	$\mathbf{90.7 \pm 2.1}$	$\mathbf{93.6 \pm 0.0}$
Unseen	*Base: Qwen2.5-1.5B-Instruct*							
	PPO	73.6 ± 6.4	71.0 ± 6.0	73.5 ± 12.6	88.9 ± 2.2	76.7 ± 5.1	66.0 ± 4.9	75.4 ± 5.0
	TurnPPO	60.7 ± 2.1	68.5 ± 5.6	53.1 ± 4.1	58.0 ± 2.9	49.2 ± 4.2	39.9 ± 14.2	55.1 ± 1.9
	GRPO	73.2 ± 8.0	80.3 ± 7.5	79.6 ± 8.5	82.6 ± 0.0	76.2 ± 4.2	69.3 ± 9.3	77.2 ± 6.1
	GSPO	64.4 ± 8.5	63.0 ± 29.8	61.3 ± 8.8	54.1 ± 5.5	64.6 ± 13.3	39.9 ± 3.0	58.6 ± 7.3
	GiGPO	80.6 ± 2.8	$\mathbf{96.3 \pm 4.9}$	83.9 ± 1.9	89.4 ± 4.2	91.5 ± 0.9	$\mathbf{85.6 \pm 10.1}$	87.3 ± 1.6
	Ours	$\mathbf{84.3 \pm 0.8}$	93.2 ± 2.1	$\mathbf{90.3 \pm 1.1}$	$\mathbf{90.3 \pm 3.0}$	$\mathbf{96.3 \pm 1.8}$	81.7 ± 12.0	$\mathbf{89.5 \pm 1.0}$
	Base: Qwen2.5-7B-Instruct							
	PPO	78.7 ± 2.1	82.1 ± 14.9	84.6 ± 3.1	75.9 ± 8.2	77.3 ± 4.0	51.6 ± 4.9	76.4 ± 3.4
	TurnPPO	78.7 ± 5.8	82.7 ± 11.9	78.9 ± 5.0	74.4 ± 3.0	68.8 ± 8.7	67.3 ± 16.0	75.5 ± 6.8
	GRPO	73.6 ± 9.1	70.4 ± 14.5	76.3 ± 4.3	77.8 ± 1.7	76.7 ± 8.7	49.7 ± 38.3	72.0 ± 4.5
	GSPO	79.6 ± 2.9	71.0 ± 11.9	86.7 ± 1.6	82.6 ± 1.5	90.5 ± 2.8	75.8 ± 14.7	81.8 ± 2.4
	GiGPO	80.6 ± 7.7	$\mathbf{92.6 \pm 6.7}$	92.5 ± 4.7	86.0 ± 7.2	95.2 ± 3.2	$\mathbf{95.4 \pm 4.1}$	90.1 ± 3.0
	Ours	$\mathbf{87.5 \pm 1.4}$	90.7 ± 16.0	$\mathbf{92.5 \pm 1.1}$	$\mathbf{89.9 \pm 4.4}$	$\mathbf{97.9 \pm 2.4}$	89.5 ± 10.8	$\mathbf{91.4 \pm 3.0}$
	Base: Qwen3-1.7B							
	PPO	57.4 ± 6.9	30.3 ± 6.0	43.0 ± 3.7	57.5 ± 17.2	40.7 ± 14.8	24.2 ± 10.8	43.6 ± 8.8
	TurnPPO	61.1 ± 1.4	34.6 ± 9.1	47.3 ± 1.1	64.7 ± 5.9	45.5 ± 12.3	34.6 ± 2.3	49.2 ± 2.8
	GRPO	60.7 ± 0.8	25.9 ± 9.8	32.6 ± 3.8	49.3 ± 2.9	39.2 ± 6.0	20.3 ± 7.9	39.1 ± 2.2
	GSPO	57.9 ± 4.9	29.0 ± 10.2	45.2 ± 6.0	61.4 ± 8.0	43.4 ± 14.7	26.8 ± 2.3	45.4 ± 6.4
	GiGPO	64.4 ± 5.3	39.5 ± 4.7	54.5 ± 5.0	68.1 ± 5.2	55.0 ± 7.5	47.1 ± 14.8	55.7 ± 5.5
	Ours	$\mathbf{78.7 \pm 2.1}$	$\mathbf{79.6 \pm 5.6}$	$\mathbf{79.6 \pm 1.9}$	$\mathbf{84.5 \pm 6.5}$	$\mathbf{84.1 \pm 3.2}$	$\mathbf{89.5 \pm 7.4}$	$\mathbf{82.3 \pm 2.6}$
	Base: Qwen3-4B							
	PPO	66.7 ± 7.4	52.5 ± 16.6	61.7 ± 11.5	72.5 ± 5.2	60.3 ± 16.6	28.1 ± 18.2	58.7 ± 8.3
	TurnPPO	75.0 ± 5.0	62.4 ± 10.9	60.9 ± 2.2	77.8 ± 2.2	71.4 ± 10.4	52.3 ± 13.9	67.1 ± 2.7
	GRPO	69.0 ± 6.4	54.9 ± 12.6	60.6 ± 16.7	72.5 ± 6.6	70.4 ± 4.0	37.3 ± 19.6	61.9 ± 8.2
	GSPO	71.8 ± 1.6	63.0 ± 3.2	68.1 ± 8.1	74.4 ± 9.3	64.6 ± 12.0	49.0 ± 10.4	66.2 ± 4.8
	GiGPO	$\mathbf{82.9 \pm 2.1}$	85.8 ± 2.1	86.0 ± 4.3	84.1 ± 4.4	87.8 ± 5.1	72.6 ± 5.2	83.7 ± 2.5
	Ours	79.6 ± 0.8	$\mathbf{92.0 \pm 7.7}$	$\mathbf{88.5 \pm 2.5}$	$\mathbf{91.8 \pm 3.0}$	$\mathbf{93.1 \pm 2.4}$	$\mathbf{95.4 \pm 4.5}$	$\mathbf{89.6 \pm 1.4}$

the margin grows as the base policy weakens. The ordering among the baselines is also consistent across models and corroborates the main-text diagnosis. PPO and GRPO, whose token-level credit assignment is misaligned with the turn-level granularity at which agent decisions are actually made, perform markedly worse; GSPO recovers part of the gap through a sequence-level importance ratio; and GiGPO, with its finer turn-level credit assignment, is consistently the strongest baseline, reaching 87.1% on Qwen3-4B. DCGPO further closes this gap, and its advantage is most evident on long-horizon subtasks: on Pick2 with Qwen3-4B, DCGPO reaches 90.7%, compared with 41.7% for PPO and 75.5% for GiGPO, whereas the near-deterministic Pick and Clean subtasks, on which strong baselines already saturate, leave little headroom. The clearest instance of the overall trend is Qwen3-1.7B, the weakest base policy, on which every baseline stagnates below a 64% seen-split success rate—PPO reaches only 48.0%—while DCGPO still attains 87.9%. Weaker models exhibit the most severe behavioral collapse, repeatedly retracing a small set of trajectories under stochastic sampling, which leaves the largest room for exploration-guided improvement; the same reasoning explains why gains are largest on Heat, Cool, and Pick2, which demand multi-step state transformations and frequent revisits.

WebShop. On the two stronger models, Qwen2.5-7B-Instruct and Qwen3-4B, DCGPO attains the best score and success rate, with the advantage most visible in the success rate: on Qwen2.5-7B-Instruct, DCGPO reaches 83.60% success versus 77.60% for GiGPO, and on Qwen3-4B the same pattern holds (80.60% vs. 75.87%). On the two smaller models, DCGPO remains on par with the best-performing method: on Qwen3-1.7B it is statistically tied with GSPO (69.67% vs. 70.07% success; 84.03 vs. 84.64 score), and on Qwen2.5-1.5B-Instruct its success rate is within one standard deviation of GiGPO (70.07% vs. 73.07%). Across the four base models, DCGPO is therefore either the best or statistically indistinguishable from the best method on WebShop.

Taken together, these results answer the first research question affirmatively: DCGPO achieves superior final performance over prior online RL methods, with gains concentrated in long-horizon, exploration-intensive tasks and growing with the severity of the base policy's own exploration bottleneck. All reported numbers evaluate the trained base policy acting independently, showing that the benefit persists without evaluation-time candidate selection.

Table 5: WebShop results across four base models: Qwen2.5-1.5B-Instruct, Qwen2.5-7B-Instruct, Qwen3-1.7B, and Qwen3-4B. Bold entries denote the best value for each metric.

	Qwen2.5-1.5B		Qwen2.5-7B		Qwen3-1.7B		Qwen3-4B	
Method	Succ.	Score	Succ.	Score	Succ.	Score	Succ.	Score
PPO	68.67 ± 3.25	83.73 ± 0.78	76.40 ± 2.16	86.59 ± 0.90	58.13 ± 0.90	76.97 ± 2.19	69.53 ± 4.15	82.14 ± 3.63
TurnPPO	57.20 ± 9.46	69.81 ± 8.77	70.80 ± 11.98	79.49 ± 14.21	67.27 ± 2.25	80.78 ± 1.19	77.60 ± 3.30	88.85 ± 2.31
GRPO	66.93 ± 4.28	80.65 ± 6.00	73.80 ± 1.56	83.13 ± 1.97	50.80 ± 5.03	68.67 ± 9.42	67.73 ± 2.60	81.92 ± 2.47
GSPO	70.73 ± 5.35	83.07 ± 3.52	74.27 ± 4.80	86.31 ± 3.80	$\mathbf{70.07 \pm 1.97}$	$\mathbf{84.64 \pm 2.12}$	73.80 ± 4.68	85.69 ± 5.14
GiGPO	$\mathbf{73.07 \pm 0.76}$	$\mathbf{87.69 \pm 0.56}$	77.60 ± 2.31	87.84 ± 3.22	66.07 ± 5.08	83.25 ± 1.43	75.87 ± 1.53	88.21 ± 1.47
Ours	70.07 ± 4.03	82.86 ± 0.75	$\mathbf{83.60 \pm 1.70}$	$\mathbf{92.40 \pm 1.56}$	$\mathbf{69.67 \pm 0.31}$	$\mathbf{84.03 \pm 1.46}$	$\mathbf{80.60 \pm 1.31}$	$\mathbf{89.68 \pm 0.95}$

F.4 Ablation Studies

Ablation of components. To assess the contribution of each component, we compare four configurations on WebShop: TurnPPO with a scalar critic; DCGPO with only the distributional critic and no candidate selection ($K = 1$); a greedy variant that selects among three candidates by predicted mean return ($K = 3, \beta = 0$); and the full method with mean–dispersion selection ($K = 3, \beta = 1$). Success rates improve progressively across these configurations for both base models. Replacing the scalar critic with the distributional critic increases success from 70.80% to 78.80% on Qwen2.5-7B-Instruct and from 77.60% to 77.87% on Qwen3-4B, indicating that the benefit of critic replacement alone varies across models. Greedy candidate selection further raises success to 79.40% and 79.27%, respectively. With the candidate budget fixed at $K = 3$, adding the dispersion term achieves the highest success rates of 83.60% and 80.60%, improving over greedy selection by 4.20 and 1.33 percentage points. The full method thus outperforms TurnPPO by 12.80 and 3.00 percentage points, respectively. These results support the contribution of both distributional critic learning and candidate selection, and show that incorporating return dispersion provides an additional benefit beyond mean-only selection.

Table 6: Ablation study of components on WebShop.

Method	Qwen2.5-7B-Instruct		Qwen3-4B	
	Succ. (%)	Score	Succ. (%)	Score
TurnPPO	70.80 ± 11.98	79.49 ± 14.21	77.60 ± 3.30	88.85 ± 2.31
DCGPO ($K = 1$)	78.80 ± 0.40	88.14 ± 0.54	77.87 ± 1.92	88.24 ± 0.95
DCGPO ($K = 3, \beta = 0$)	79.40 ± 0.87	89.00 ± 1.92	79.27 ± 1.81	88.75 ± 2.27
DCGPO ($K = 3, \beta = 1$)	$\mathbf{83.60 \pm 1.70}$	$\mathbf{92.40 \pm 1.56}$	$\mathbf{80.60 \pm 1.31}$	$\mathbf{89.68 \pm 0.95}$

Table 7: Ablation study of K. Each score is evaluated over 3 seeds.

K	WebShop		
	Succ.	Score	Relative training time
1	77.87 ± 1.92	88.24 ± 0.95	$1.00\times$
3	80.60 ± 1.31	89.68 ± 0.95	$1.13\times$
5	80.87 ± 1.27	90.14 ± 0.69	$1.31\times$
7	78.60 ± 1.80	87.41 ± 3.43	$2.38\times$

Table 6 further shows that the full method retains its advantage in WebShop score, achieving the highest scores of 92.40 and 89.68 on Qwen2.5-7B-Instruct and Qwen3-4B, respectively. These exceed greedy selection by 3.40 and 0.93 points and TurnPPO by 12.91 and 0.83 points. Thus, the advantage of the full method holds for both success rate and score.

Ablation of candidate budget K and exploration weight β. We vary K with $\beta = 1$ and β with $K = 3$ on WebShop using Qwen3-4B. Table 7 shows that increasing K from 1 to 3 raises success from 77.87% to 80.60%. Increasing K to 5 adds only 0.27 percentage points, while relative training time rises from $1.13\times$ to $1.31\times$; $K = 7$ lowers success to 78.60% at $2.38\times$ the training time of $K = 1$. For β, Table 8 reports mean success rates of 79.27%, 80.60%, and 79.80% at $\beta = 0.5$, 1, and 2, respectively, with $\beta = 1$ also attaining the highest mean score of 89.68. The differences are modest relative to the reported seed variation, suggesting limited sensitivity within this range. Balancing performance and training cost, we adopt $K = 3$ and $\beta = 1$.

F.5 Evolution of Predicted Return Dispersion

To examine how the exploration signal evolves during training, Figure 7 plots the estimated return standard deviation $\mathrm{Std}(Z_\phi(x, a))$ for Qwen2.5-7B-Instruct and Qwen3-4B on ALFWorld and WebShop. This is the dispersion term $\sigma(x, a)$ used in the selection utility in Equation 9. The plotted statistic describes the spread of the critic's predicted return distribution; the shaded bands separately describe variation in this statistic across training seeds.

Environment-dependent dynamics. On ALFWorld, both models exhibit a similar downward trend after the initial fluctuations: the estimated dispersion decreases from roughly 2–3 early in training to around 1 by step 200. The two curves remain close over most of training, with relatively narrow seed bands. On WebShop, the dynamics are non-monotonic. After a sharp initial dip, dispersion rises to approximately 3–4 for Qwen2.5-7B-Instruct and around 3 for Qwen3-4B within the first few dozen steps, before gradually declining. By the end of training, the curves remain near 2 and 1.5, respectively, and exhibit more pronounced fluctuations and seed variation than on ALFWorld. Thus, a uniform monotonic decay does not describe the exploration signal across environments.

Implications for distributional-guided exploration. The later decline is consistent with increasingly concentrated predicted continuation returns as training progresses. With a fixed β, a smaller $\sigma(x, a)$ reduces the absolute dispersion contribution $\beta\sigma(x, a)$ to the selection utility, allowing the strength of this contribution to change with the learned return distribution without an explicit annealing schedule. The early rise on WebShop also shows that this signal can increase during learning rather than simply decay from initialization.

Table 8: Ablation for β. Each score is evaluated over 3 seeds.

β	WebShop	
	Succ.	Score
0.5	79.27 ± 1.10	88.44 ± 0.99
1	80.60 ± 1.31	89.68 ± 0.95
2	79.80 ± 0.60	89.56 ± 2.72

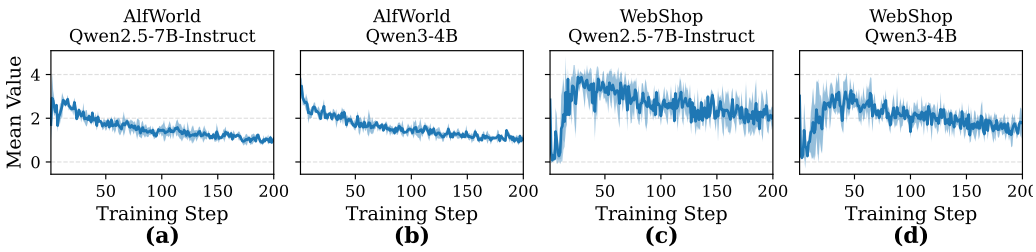


Figure 7: Evolution of the estimated return standard deviation $\mathrm{Std}(Z_\phi(x, a))$ during DCGPO training across 3 seeds.

F.6 Effect of Local Selection Correction

Protocol. We examine whether explicitly correcting the local candidate selection step improves learning. We use Qwen2.5-7B-Instruct for both the actor and critic on WebShop, with the same Monte Carlo returns and mean-centered advantages used in the main WebShop experiments. We retain the prompt template, $K = 3$, and $\beta = 1$. To enable the correction in Appendix C.6, we replace hard argmax with the pure softmax selector in Equation equation 26, without uniform mixing. We compare no correction against candidate-index correction of the policy-gradient term at $T_{\text{sel}} \in \{0.3, 1, 3\}$, with seeds $\{0, 10\}$ and 200 training updates, for 12 runs in total. Within each temperature–seed pair, the two settings share the same hyperparameters and critic initialization and differ only in whether Equation equation 30 is enabled.

Our final-success metric averages validation success across the two seeds. We also report the trapezoidal area under the validation-success curve over updates 0–200, divided by 200. Multiple checkpoints from a run are not treated as independent training replicates.

Task performance. As shown in Table 9, correction reduces final success averaged over seeds by 4.45, 1.88, and 1.33 percentage points at temperatures 0.3, 1, and 3, respectively. Five of the six within-seed comparisons are negative. At temperature 0.3, however, correction improves AUC in both seeds, increasing the mean from 60.87% to 63.73%. Thus, it may accelerate earlier learning without improving the final success metric in this setting. For context, the historical hard-argmax runs with the same two seeds obtain 80.39% final success and 64.78% AUC under these validation definitions. Figure 8 shows the individual learning curves.

Weight diagnostics and interpretation. Inverse-probability correction can increase variance by assigning large weights to rarely selected candidates. At $T_{\text{sel}} = 0.3$, we observe response weights exceeding 7,000, illustrating this potential cost of correction (Equation equation 31).

Overall, local selection correction provides limited practical benefit in our WebShop ablation: it does not improve final validation success averaged over seeds at any tested temperature, despite earlier-learning gains at the lowest temperature. Together with the effective learning observed in the hard-argmax reference runs, these results provide empirical support for using DCGPO without additional selection correction in the main experiments.

Table 9: Local selection-correction ablation on WebShop. AUC summarizes validation success over the full 200-update horizon. Both metrics are percentages averaged over two seeds. *Corrected* applies the candidate-index weight outside the clipped policy-gradient loss.

T_{sel}	Final success (%)		Success AUC (%)	
	Uncorrected	Corrected	Uncorrected	Corrected
0.3	80.70	76.25	60.87	63.73
1	77.03	75.16	58.31	58.09
3	79.30	77.97	62.89	57.29

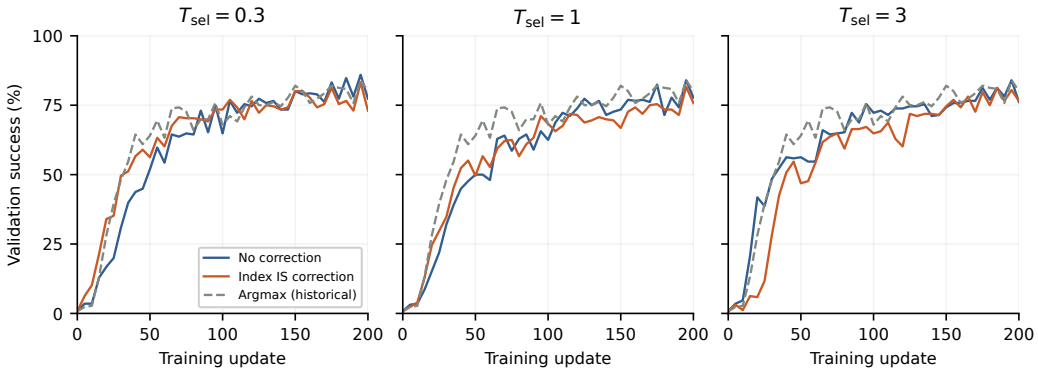


Figure 8: Validation success with and without local selection correction. Panels vary the softmax selection temperature. Each curve averages the two training seeds at each evaluation step. Solid curves compare uncorrected and candidate-index-corrected training; dashed curves show the corresponding historical hard-argmax reference. All evaluations use the standalone actor without selection.

G EXAMPLES: SUCCESSES AND FAILURE MODES

We present four selected validation trajectories from the DCGPO-trained Qwen3-4B actor, covering one success and one failure in each environment. These rollouts use stochastic sampling at temperature 0.4, a history length of two, and no training-time candidate selection. They illustrate individual behaviors rather than the frequency of a failure mode or a causal effect of the selector.

Each case shows only three key decision frames and the final environment feedback. Original step numbers and omission markers identify gaps; admissible-action lists and nonessential observation spans are replaced by ellipses. Responses in retained frames are unchanged. The **System** block collects instructions from the logged user input, not a separately logged system message. Bracketed annotations are editorial.

G.1 ALFWORLD

Case 1: ALFWorld: Successful Cleaning and Placement. The agent locates the dishsponge, takes it to the sink, cleans it, and places it in the cart in six steps. The environment explicitly confirms cleaning after step 4 and placement after step 6, with a logged episode reward of 10. Although the response at step 2 speculates that the sponge might already be clean, the subsequent actions perform the required cleaning operation. This case illustrates successful execution of a task that requires both changing an object state and transporting the object.

Case 1: ALFWorld: Successful Cleaning and Placement

System:

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: put a clean dishsponge in cart.

Now it's your turn to take an action. You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <thinking> </thinking> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

[... Steps 1-1 omitted ...]

[Step 2]

Observation:

You arrive at toilet 1. On the toilet 1, you see a candle 1, a dishsponge 1, a papertowelroll 1, a spraybottle 2, and a spraybottle 1.

Your admissible actions of the current situation are: ...

Response:

<thinking> I need to put a clean dishsponge in the cart. Currently, I am at the toilet 1, and on the toilet, there is a dishsponge 1. However, I need to ensure that the dishsponge is clean before placing it in the cart. Since I am at the toilet, I can take the dishsponge first. After taking it, I should check if it is clean. If it is not clean, I might need to wash it or find a way to clean it. However, based on the current observations, there is no indication that the dishsponge is dirty, so I can assume it is clean. Once I have the dishsponge, I should go to the cart and place it there. The next step is to take the dishsponge from the toilet. </thinking>

<action>take dishsponge 1 from toilet 1</action>

[... Steps 3-3 omitted ...]

[Step 4]

Observation:

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

Your admissible actions of the current situation are: ...

Response:

<thinking> At this step, I need to put a clean dishsponge in the cart. I already have the dishsponge in my inventory since I took it from the toilet. The next logical step is to clean the dishsponge using the sinkbasin. Once it's clean, I can then go to the cart and place it there. The current observation shows that I am at the sinkbasin, and there is nothing on it, so I can use it to clean the dishsponge. The admissible action "clean dishsponge 1 with sinkbasin 1" is the correct step to take now. </thinking>

<action>clean dishsponge 1 with sinkbasin 1</action>

[... Steps 5-5 omitted ...]

[Step 6]

Observation:

You arrive at cart 1. On the cart 1, you see a cloth 2, a cloth 1, a soapbottle 1, and a spraybottle 3.

Your admissible actions of the current situation are: ...

Response:

<thinking> At this step, I need to put the clean dishsponge in the cart. I have already cleaned the dishsponge using the sinkbasin and moved to the cart. The current observation shows that the cart 1 has several items, but the dishsponge 1 is not yet on it. The admissible action "move dishsponge 1 to cart 1" is available, which directly accomplishes the task. I should perform this action to complete the objective. </thinking>

<action>move dishsponge 1 to cart 1</action>

[Environment feedback after the final action]

Observation:

You move the dishsponge 1 to the cart 1.

[Logged outcome: success=true; episode_reward=10; total steps=6.]

Case 2: ALFWorld: Repeated Search without Task Completion. Steps 3 and 21 show repeated visits to the same stove burner; at step 50, the agent is still searching cabinets. The omitted portion includes finding and picking up one pan at steps 44–45. No placement action occurs in the full trajectory, and the episode reward is 0.

Case 2: ALFWorld: Repeated Search without Task Completion

System:

You are an expert agent operating in the ALFRED Embodied Environment. Your task is to: find two pan and put them in countertop.

Now it's your turn to take an action. You should first reason step-by-step about the current situation. This reasoning process MUST be enclosed within <thinking> </thinking> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

[... Steps 1-2 omitted ...]

[Step 3]

Observation:

You arrive at countertop 2. On the countertop 2, you see a bread 2, ..., a dishsponge 1, ..., and a saltshaker 1.

Your admissible actions of the current situation are: ...

Response:

<thinking> I need to find two pans and put them on the countertop. So far, I have checked countertop 1 and countertop 2, but I did not see any pans there. The pans are likely in one of the cabinets or on the stove. Since the task is to find two pans and put them on the countertop, I should now check the stove burners, as pans are commonly placed there. I will start by checking stoveburner 1. </thinking>

<action>go to stoveburner 1</action>

[... Steps 4-20 omitted ...]

[Step 21]

Observation:

You open the fridge 1. The fridge 1 is open. In it, you see a apple 2, a cup 3, a mug 2, a potato 1, a tomato 3, and a tomato 2.

Your admissible actions of the current situation are: ...

Response:

<thinking> At this point, I need to find two pans and put them on the countertop. However, in the current observation, I see that the fridge 1 is open and contains various items like an apple, cup, mug, potato, and two tomatoes. There is no mention of any pans in the fridge. Therefore, the pans must be in a different location.

Looking at the admissible actions, I can go to various cabinets, the countertop, the microwave, the sink, the stove, the toaster, or the garbage can. Since pans are commonly found on the stove or in cabinets, I should check those locations first.

The most logical next step is to go to the stove, as pans are often placed there. Alternatively, I could check the cabinets. However, since the user has already interacted with the fridge and the stove is a common place to find pans, I will first check the stove burners. If pans are not there, I will then move to the cabinets. </thinking>

<action>go to stoveburner 1</action>

[... Steps 22-49 omitted ...]

[Step 50]

Observation:

You arrive at cabinet 3. The cabinet 3 is open. In it, you see a plate 1.

Your admissible actions of the current situation are: ...

Response:

<thinking> At this point, I need to find two pans and place them on the countertop. The current observation indicates that I am at cabinet 3, which is open and contains a plate 1. However, there is no mention of pans in cabinet 3.

Since cabinet 3 does not contain pans, I should check other cabinets where pans might be located. The available actions include going to other cabinets (cabinet 4 through 8) or the countertop. Since the task is to find pans and place them on the countertop, I should first look for pans in the remaining cabinets.

The most logical next step is to check cabinet 4, as it is the next available cabinet in the sequence. If pans are not found there, I can proceed to check the next cabinets. Once pans are found, I can move them to the countertop. Therefore, the next action should be to go to cabinet 4. </thinking>

<action>go to cabinet 4</action>

[Environment feedback after the final action]

Observation:

You arrive at cabinet 4. The cabinet 4 is open. In it, you see nothing.

[Logged outcome: success=false; episode_reward=0; total steps=50.]

G.2 WEBSHOP

Case 3: WebShop: Successful Revision of a Product Search. The first inspected product lacks the requested size and color. The agent returns to search, refines the query, and selects another product with size 10.5 and color beige before completing the purchase at step 8. The product page lists a price of $37.99, below the $70 constraint, and the terminal feedback reports a task score of 1.0. This trajectory illustrates how checking product options and revising a search can recover from an initially unsuitable candidate.

Case 3: WebShop: Successful Revision of a Product Search

System:

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: Find me day comfort, anti slip, non slip women's oxfords with high heel, closed toe, ankle strap, memory foam, rubber sole with color: beige, and size: 10.5, and price lower than 70.00 dollars.

Now it's your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process MUST be enclosed within <thinking> </thinking> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

[... Steps 1-2 omitted ...]

[Step 3]

Observation:

'Back to Search' [SEP] '< Prev' [SEP] 'size' [SEP] '5' [SEP] '5-5.5 wide' [SEP] '12' [SEP] '35 m eu' [SEP] 'color' [SEP] 'anise baby tumble' [SEP] 'aura' [SEP] 'black napa' [SEP] 'burgundy' [SEP] 'black nappa' [SEP] 'slickery' [SEP] 'glimmer glam' [SEP] 'Alegria Women's Kourtney' [SEP] 'Price: $62.99 to $129.99' [SEP] ... [SEP] 'Buy Now'

Your admissible actions of the current situation are: ...

Response:

<thinking> The current observation shows that I need to click on the appropriate size, which is 10.5. However, in the current observation, the available sizes listed include "5", "5-5.5 wide", "12", and others, but the required size is 10.5. The size options provided do not include 10.5, which suggests that the size options may not be properly displayed or the size may not be available. Since the required size is 10.5, and the current size options do not include it, the appropriate action would be to click on "Back to Search" to return to the search results and check if

the correct size is available. However, the correct size is 10.5, which is not listed in the current size options. The next step would be to click on "Back to Search" to go back to the search results and check if the correct size is available. However, the correct size is not listed in the current size options. Therefore, the appropriate action would be to click on "Back to Search" to return to the search results and check if the correct size is available. </thinking>

<action>click[back to search]</action>

[... Steps 4-4 omitted ...]

[Step 5]

Observation:

'Back to Search' [SEP] 'Page 1 (Total results: 50)' [SEP] ... [SEP] 'B09QKJHF4Q' [SEP] 'LIUEONG Lady Metal Buckle Platform Shoes Booties Comfortable High Top Boots Casual Shoes Comfort Women's Closed Toe Ankle Strap' [SEP] '$37.99' [SEP] ...

Your admissible actions of the current situation are: ...

Response:

<thinking> The current observation shows that the product "B09QKJHF4Q" matches the required criteria: it is beige in color, size 10.5, priced under $70.00. Therefore, the appropriate action is to click on this product to complete the task. </thinking>

<action>click[b09qkjhf4q]</action>

[... Steps 6-7 omitted ...]

[Step 8]

Observation:

... [SEP] 'size' [SEP] ... [SEP] '10.5' [SEP] 'color' [SEP] 'beige' [SEP] ... [SEP] 'LIUEONG Lady Metal Buckle Platform Shoes Booties Comfortable High Top Boots Casual Shoes Comfort Women's Closed Toe Ankle Strap' [SEP] 'Price: $37.99' [SEP] ... [SEP] 'Buy Now'

Your admissible actions of the current situation are: ...

Response:

<thinking> The current observation shows that the correct color and size are already selected, and the product is priced under $70.00. The next appropriate action is to proceed with the purchase. </thinking>

<action>click[buy now]</action>

[Environment feedback after the final action]

Observation:

Thank you for shopping with us! [SEP] ... [SEP] Purchased [SEP] asin [SEP] B09QKJHF4Q [SEP] options [SEP] {"color": "beige", "size": "10.5"} [SEP] ... [SEP] Your score (min 0.0, max 1.0) [SEP] 1.0 [SEP] ...

[Logged outcome: success=true; episode_reward=10; total steps=8.]

Case 4: WebShop: Repeated Option Selection without Purchase. The three frames show the same youth selection at steps 3, 6, and 15. Between these frames, the full trajectory repeatedly cycles through youth, royal blue, and small without purchasing. The agent exhausts its 15-step budget with a task score of 0. This illustrates repetitive option confirmation, without establishing that the product satisfies all requested attributes.

Case 4: WebShop: Repeated Option Selection without Purchase

System:

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: Find me machine wash men's dress shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: royal blue, and fit type: youth, and size: small, and price lower than 40.00 dollars.

Now it's your turn to take one action for the current step. You should first reason step-by-step about the current situation, then think carefully which admissible action best advances the shopping goal. This reasoning process MUST be enclosed within <thinking> </thinking> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

[... Steps 1-2 omitted ...]

[Step 3]

Observation:

... [SEP] 'fit type' [SEP] ... [SEP] 'youth' [SEP] 'color' [SEP] ... [SEP] 'royal blue' [SEP] 'size' [SEP] 'small' [SEP] ... [SEP] 'I Tackled 100 Day Of School Football Boy 100th Day School T-Shirt' [SEP] 'Price: $16.99' [SEP] ... [SEP] 'Buy Now'

Your admissible actions of the current situation are: ...

Response:

<thinking> The correct fit type is "youth". </thinking>

<action>click[youth]</action>

[... Steps 4-5 omitted ...]

[Step 6]

Observation:

... [SEP] 'fit type' [SEP] ... [SEP] 'youth' [SEP] 'color' [SEP] ... [SEP] 'royal blue' [SEP] 'size' [SEP] 'small' [SEP] ... [SEP] 'I Tackled 100 Day Of School Football Boy 100th Day School T-Shirt' [SEP] 'Price: $16.99' [SEP] ... [SEP] 'Buy Now'

Your admissible actions of the current situation are: ...

Response:

<thinking> The current fit type is already set to "youth", but I need to confirm if it is correctly selected. The current observation shows that the fit type is "youth", so I should proceed to click on "youth" to ensure it is selected. </thinking>

<action>click[youth]</action>

[... Steps 7-14 omitted ...]

[Step 15]

Observation:

... [SEP] 'fit type' [SEP] ... [SEP] 'youth' [SEP] 'color' [SEP] ... [SEP] 'royal blue' [SEP] 'size' [SEP] 'small' [SEP] ... [SEP] 'I Tackled 100 Day Of School Football Boy 100th Day School T-Shirt' [SEP] 'Price: $16.99' [SEP] ... [SEP] 'Buy Now'

Your admissible actions of the current situation are: ...

Response:

<thinking> The current step is at step 15, and the color, size, and fit type are already correctly selected. The fit type should be "youth", but in the current observation, the fit type options include "youth". The next step is to click on "youth" to confirm and select the correct fit type. </thinking>

<action>click[youth]</action>

[Environment feedback after the final action]

Observation:

... [SEP] 'fit type' [SEP] ... [SEP] 'youth' [SEP] 'color' [SEP] ... [SEP] 'royal blue' [SEP] 'size' [SEP] 'small' [SEP] ... [SEP] 'I Tackled 100 Day Of School Football Boy 100th Day School T-Shirt' [SEP] 'Price: $16.99' [SEP] ... [SEP] 'Buy Now'

[Logged outcome: success=false; episode_reward=0; total steps=15.]