

When Denser Credit Is Not Enough: Evidence-Calibrated Policy Optimization for Long-Horizon LLM Agent Training

Yuanfan Li^{1,†}, Qi Zhou^{4,†}, Wenjing Duan^{4,†}, Lu Chen^{1,2,3,*}

¹ X-LANCE Lab, School of Computer Science, Shanghai Jiao Tong University, Shanghai, China

² Shanghai Innovation Institution, Shanghai, China

³ Jiangsu Key Lab of Language Computing, Suzhou, China

⁴ Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an, China

[†] Equal contribution, * Correspondence to: chenlusz@sjtu.edu.cn

Abstract

Long-horizon LLM agents require reinforcement learning methods that can assign credit to intermediate decisions under sparse and delayed rewards. Recent group-based methods such as GiGPO improve over GRPO by constructing step-level advantages at repeated anchor states. However, we show that such dense credit can be statistically unreliable: under limited rollouts, rare but lucky actions may receive overly large advantages, producing divergent anchor bias and late-stage training oscillation. We propose Evidence-Calibrated Policy Optimization (ECPO), a critic-free policy optimization algorithm that calibrates step-level credit before policy updates. ECPO combines Evidence-Calibrated Action Advantage, which groups rollouts by canonical actions and shrinks low-count estimates, with Variance-Gated Credit Weighting, which suppresses anchor states dominated by within-action noise. Experiments on ALFWorld and WebShop with Qwen2.5-1.5B/7B show that ECPO consistently outperforms strong baselines, improving GiGPO by **+5.2/+7.3** success points on ALFWorld/WebShop with Qwen2.5-1.5B while adding only **0.1%** additional advantage-computation overhead. Code is available at <https://github.com/AgentQ0/ECPO>.

1 Introduction

Large language models (LLMs) have achieved remarkable success in diverse NLP tasks (Vaswani et al., 2017; Brown et al., 2020; Singh et al., 2025), and there is increasing interest in deploying them as autonomous agents capable of long-horizon decision-making (Bai et al., 2022; Shao et al., 2024; Feng et al., 2025). Such *agentic* LLMs interact with complex environments, execute multi-step plans, and must cope with delayed rewards. These capabilities pose significant challenges for reinforcement learning (RL), particularly in assigning credit to intermediate decisions

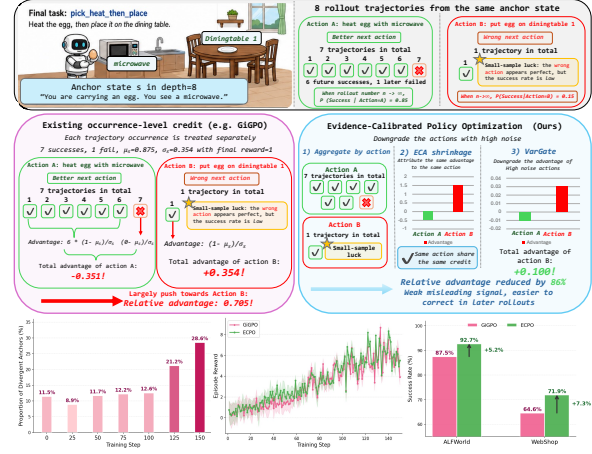


Figure 1: **Motivation and overview of ECPO.** **Top:** an ALFWorld case illustrating *divergent anchor bias*. We define a divergent anchor as an anchor state where at least one canonical action is sampled more than once, while other actions may have only singleton evidence. Under such imbalanced evidence, GiGPO can over-reward a rare lucky action based on its observed return. **Bottom-left:** divergent anchors become more frequent during training, increasing from 9% to 28%. **Bottom-middle:** this bias leads to late-stage reward oscillation; ECPO reduces the final reward standard deviation from $\sigma = 0.746$ to $\sigma = 0.555$. **Bottom-right:** by calibrating action-level evidence, ECPO improves final success rates over GiGPO on both ALFWorld and WebShop.

in long-horizon trajectories (Pignatelli et al., 2023; Wei et al., 2025).

GRPO (Shao et al., 2024) optimizes policies with group-relative trajectory returns, but its credit signal remains tied to final episode outcomes and is therefore sparse in long-horizon agentic tasks. GiGPO (Feng et al., 2025), an agent-oriented variant of GRPO, mitigates this sparsity by assigning step-level credit at repeated *anchor states*: for rollouts that revisit the same state, it compares their subsequent returns and assigns relative advantages to the actions taken from that state. This enables intermediate decisions to receive credit according to their apparent contribution to eventual task suc-

cess.

However, Figure 1 (bottom middle) shows that denser step-level credit does not automatically translate into stable optimization. Despite providing intermediate supervision beyond sparse terminal rewards, GiGPO exhibits pronounced late-stage reward oscillations, with a final reward standard deviation of $\sigma = 0.746$. This observation suggests that the key limitation may lie not only in credit sparsity, but also in the reliability of the constructed step-level credit signal.

To understand the source of this instability, we analyze anchor-level credit assignment in GiGPO and identify a failure mode that we call *divergent anchor bias*. Figure 1 (top) shows a representative ALFWorld anchor state with 8 rollouts. At this state, action A is sampled by seven rollouts and leads to six eventual successes, while action B is sampled only once and happens to succeed. Since GiGPO estimates step-level advantages directly from observed downstream returns, the rare action B can receive an overly large positive advantage, even though its true downstream success probability may be lower than that of action A under sufficient sampling. In this case, GiGPO confuses a lucky observation with reliable action quality, thereby pushing the policy toward a high-variance and potentially suboptimal action.

This bias becomes more severe as training progresses. As the policy improves, action distributions at anchor states become increasingly concentrated: dominant actions may appear many times, while low-probability exploratory actions are still occasionally sampled. Such imbalanced anchor groups make rare actions especially vulnerable to overestimation when they happen to succeed. As shown in Figure 1 (bottom left), the fraction of divergent anchors increases from 9% to 28% during training. This explains the late-stage reward oscillation in Figure 1 (bottom middle): once divergent anchors become frequent, GiGPO injects high-variance step-level gradients into policy updates, slowing convergence and destabilizing the later training process.

The underlying mechanism is statistical. Step-level credit assignment aims to identify, at each anchor state, which action has the highest expected downstream success. Yet the true posterior success probability of each action is unknown. With a limited rollout budget, the algorithm can only approximate it using empirical outcomes from a small number of samples. When an action is ob-

served only once, its empirical success rate is either 0 or 1, which is a highly unreliable point estimate. Therefore, directly converting observed returns into step-level advantages can over-reward rare but lucky actions and under-reward frequent actions with more stable evidence.

Motivated by this diagnosis, we propose **Evidence-Calibrated Policy Optimization (ECPO)**, which calibrates step-level credit before using it for policy updates. ECPO contains two components. *Evidence-Calibrated Action Advantage* (ECA) groups rollouts by canonical actions and applies shrinkage estimation to obtain calibrated action-level advantages, reducing the overestimation of low-count actions. *Variance-Gated Credit Weighting* (VarGate) decomposes anchor-level return variation into between-action signal and within-action noise, and suppresses step-level credit when the observed action differences are statistically unreliable. Together, these components retain the benefit of dense step-level supervision while reducing the variance introduced by small-sample anchor estimates.

We evaluate ECPO on two long-horizon agent benchmarks: ALFWorld (Shridhar et al., 2020) and WebShop (Yao et al., 2022a) with Qwen2.5-1.5B/7B-Instruct (Qwen Team et al., 2024). Results show that ECPO consistently outperforms prompt-based agents, actor-critic RL baselines (Schulman et al., 2017), and prior group-based policy optimization methods such as GRPO (Shao et al., 2024) and GiGPO (Feng et al., 2025). In particular, with Qwen2.5-1.5B, ECPO improves over GiGPO by **+5.2** points on ALFWorld and **+7.3** success points on WebShop. With Qwen2.5-7B, ECPO further improves the overall ALFWorld success rate from 90.8% to **91.9%**, and improves WebShop success rate from 72.4% to **74.7%**. We also analyze different rollout group sizes and find that ECPO consistently improves over GiGPO under $N \in \{4, 8, 10\}$, with gains of **+4.7**, **+5.2**, and **+3.9** points, respectively. These gains come with negligible computational overhead: ECPO adds only **0.1%** extra advantage-computation overhead over GiGPO, suggesting that evidence-calibrated credit construction provides a practical and scalable way to stabilize long-horizon LLM agent training.

2 Related Work

Reinforcement learning for LLM agents. RL has been widely used to train language agents for dynamic, interactive, and long-horizon environments. Early work applied DQN (Mnih et al., 2015) to text-based games and language-conditioned decision making (Narasimhan et al., 2015; He et al., 2016; Hausknecht et al., 2020), while recent benchmarks extend this setting to embodied tasks, web navigation, and application-centered tool execution (Shridhar et al., 2020; Yao et al., 2022a; Trivedi et al., 2024). Prompt-based agents such as ReAct (Yao et al., 2022b) improve interaction by interleaving reasoning and acting, but lack environment-specific policy learning. Recent RL agents explore hierarchical RL, search-guided optimization, entropy-regularized tuning, memory-efficient PPO variants, and end-to-end multi-turn RL (Zhou et al., 2024; Wang et al., 2025). More recent methods address sparse credit: Tree-GRPO uses tree-structured rollouts (Ji et al., 2025), GiGPO constructs step-level advantages from repeated anchor states (Feng et al., 2025), HCAPO refines credit through hindsight reasoning (Tan et al., 2026), and RAPO augments exploration with retrieved off-policy traces (Zhang et al., 2026). However, these methods densify credit while assuming the estimated step signal is reliable. In contrast, ECPO studies credit reliability itself, showing that small rollout groups induce divergent anchor bias and calibrating action-level evidence before updates.

Reinforcement learning for large language models. RL is central to improving LLMs. RLHF trains models to follow human preferences (Ziegler et al., 2019; Stiennon et al., 2020; Ouyang et al., 2022; Bai et al., 2022), with PPO (Schulman et al., 2017) widely used despite value-model overhead. To reduce this cost, critic-free estimators such as REINFORCE and RLOO have been revisited (Williams, 1992; Ahmadian et al., 2024). For reasoning, GRPO (Shao et al., 2024) estimates group-relative advantages from multiple responses, inspiring scalable algorithms including Dr.GRPO, DAPO, CPPO, and GSPO (Liu et al., 2025; Yu et al., 2025; Lin et al., 2025; Zheng et al., 2025), with applications to mathematical reasoning, search, and tool use (Shao et al., 2024; Dong et al., 2025a,b). Our work follows this critic-free group-based paradigm, but targets long-horizon agentic tasks

where credit must be assigned across environment-dependent actions rather than whole responses. ECPO complements existing methods by introducing evidence-calibrated step-level advantages while preserving efficiency.

3 Preliminaries

Problem setup. We consider a long-horizon agentic RL setting, where an LLM agent completes a multi-step task specified by $x \sim p(\mathcal{X})$. At step $t = 1, \dots, T$, the agent observes a state $s_t \in \mathcal{S}$, generates a textual action $a_t \in \mathcal{V}^{\leq n}$, receives a reward $r_t \in \mathbb{R}$, and transits to s_{t+1} . A full episode forms a trajectory

$$\tau = \{(s_1, a_1, r_1), \dots, (s_T, a_T, r_T)\}. \quad (1)$$

The policy $\pi_\theta(a_t \mid s_t, x)$ defines the action distribution conditioned on the task description and current state. Since rewards in long-horizon tasks are often sparse or delayed, assigning credit to intermediate actions is challenging.

Group-based RL. Given a task x , group-based RL samples N trajectories $\{\tau_i\}_{i=1}^N$ from the old policy $\pi_{\theta_{\text{old}}}$ and computes advantages from within-group return statistics without learning a separate value function. Let $R_i = R(\tau_i)$ be the return of trajectory τ_i . GRPO (Shao et al., 2024) estimates the trajectory-level advantage as

$$A_i^{\text{GRPO}} = \frac{R_i - \mu_R}{\sigma_R + \epsilon}, \quad \mu_R = \frac{1}{N} \sum_{j=1}^N R_j, \quad (2)$$

where σ_R is the standard deviation of $\{R_j\}_{j=1}^N$ and ϵ is a small constant. This critic-free formulation is scalable, but it assigns the same trajectory-level advantage to all actions in an episode.

Step-level credit in GiGPO. GiGPO (Feng et al., 2025) introduces step-level credit by exploiting repeated *anchor states*. An anchor state is a state s visited by multiple rollout occurrences in the same trajectory group, allowing actions from the same local context to be compared. For trajectory $\tau_i = \{(s_{i,t}, a_{i,t}, r_{i,t})\}_{t=1}^{T_i}$, the future return from step t is

$$G_{i,t} = \sum_{\ell=t}^{T_i} \gamma^{\ell-t} r_{i,\ell}, \quad (3)$$

where γ is the discount factor. For anchor state s , GiGPO collects its occurrence set

$$\mathcal{I}_s = \{(i, t) \mid s_{i,t} = s\}. \quad (4)$$

It then normalizes future returns within this anchor group:

$$A_{i,t}^{\text{anchor}} = \frac{G_{i,t} - \mu_s}{\sigma_s + \epsilon}, \quad (i, t) \in \mathcal{I}_s, \quad (5)$$

where μ_s and σ_s are the mean and standard deviation of $\{G_{j,k} \mid (j, k) \in \mathcal{I}_s\}$. The final advantage is

$$\hat{A}_{i,t}^{\text{GiGPO}} = A_i^{\text{GRPO}} + \omega A_{i,t}^{\text{anchor}}, \quad (6)$$

where ω controls the strength of step-level credit. GiGPO provides denser supervision than GRPO, but its anchor advantage is computed from raw occurrence-level returns and is therefore sensitive to small or imbalanced anchor groups.

4 Methodology

As discussed in Section 3, GiGPO improves over trajectory-level GRPO by assigning step-level credit at repeated anchor states. However, its anchor advantage $A_{i,t}^{\text{anchor}}$ is computed from raw future returns. Under limited rollouts, this occurrence-level estimator can be unreliable: a rare action with one lucky successful rollout may receive a larger advantage than a frequent action supported by more stable evidence. This produces divergent anchor bias, where the policy update is driven by accidental downstream success rather than reliable action quality, leading to high-variance updates in later training.

We propose **Evidence-Calibrated Policy Optimization** (ECPO), a critic-free algorithm that treats step-level credit construction as a statistical evidence calibration problem. Figure 2 illustrates the overall workflow. Rather than directly trusting occurrence-level returns, ECPO aggregates rollouts by canonical actions, calibrates action-level estimates with shrinkage, and estimates anchor reliability through variance decomposition. The final update combines the group-level trajectory advantage with calibrated and reliability-gated step-level credit. In the following subsections, we introduce *Evidence-Calibrated Action Advantage* (ECA), *Variance-Gated Credit Weighting* (VarGate), and the resulting policy optimization objective. We analyze ECPO theoretically in Appendix D and summarize the training procedure in Algorithm 1.

4.1 Evidence-Calibrated Action Advantage

GiGPO assigns step-level credit by normalizing the future returns of all occurrences that revisit the

same anchor state. This is effective when anchor groups are balanced and sufficiently sampled, but unreliable under limited rollout budgets. When an action is observed only once, its empirical return is a high-variance point estimate: a rare lucky action can receive an overly large advantage even if its true downstream success probability is low. The key issue is that occurrence-level credit confounds current action quality with later trajectory randomness. Thus, step-level credit should be estimated at the action level, and low-count actions should be calibrated before policy updates.

We introduce *Evidence-Calibrated Action Advantage*, which groups occurrences by canonical actions and applies shrinkage estimation. For an anchor state s with occurrence set $\mathcal{I}_s$, each raw textual action $a_{i,t}$ is mapped to a canonical action

$$u_{i,t} = \text{can}(a_{i,t}), \quad (7)$$

where $\text{can}(\cdot)$ is a deterministic environment-grounded canonicalizer that removes superficial textual variation while preserving executable action identity. Invalid or unmatched outputs are mapped to a special invalid-action token rather than merged with valid actions. The canonical action set is

$$\mathcal{U}_s = \{u_{i,t} \mid (i, t) \in \mathcal{I}_s\}. \quad (8)$$

In implementation, we lowercase and trim each generated action, remove redundant punctuation, normalize template variants, and match the result against the environment’s admissible action set at the current state. A unique match is mapped to that admissible action string; otherwise it is mapped to the invalid-action token. Anchor states are equally deterministic: we use the environment’s structured state identifier when available, and otherwise use an exactly matched normalized observation string. This avoids learned or semantic clustering and makes both action and anchor construction reproducible. For each $u \in \mathcal{U}_s$, we define

$$\begin{aligned} \mathcal{I}_{s,u} &= \{(i, t) \in \mathcal{I}_s \mid u_{i,t} = u\}, \\ n_{s,u} &= |\mathcal{I}_{s,u}|. \end{aligned} \quad (9)$$

Let $G_{i,t}$ be the future return of occurrence (i, t) , and let μ_s be the mean future return over $\mathcal{I}_s$. The empirical return of action u is

$$\bar{G}_{s,u} = \frac{1}{n_{s,u}} \sum_{(i,t) \in \mathcal{I}_{s,u}} G_{i,t}. \quad (10)$$

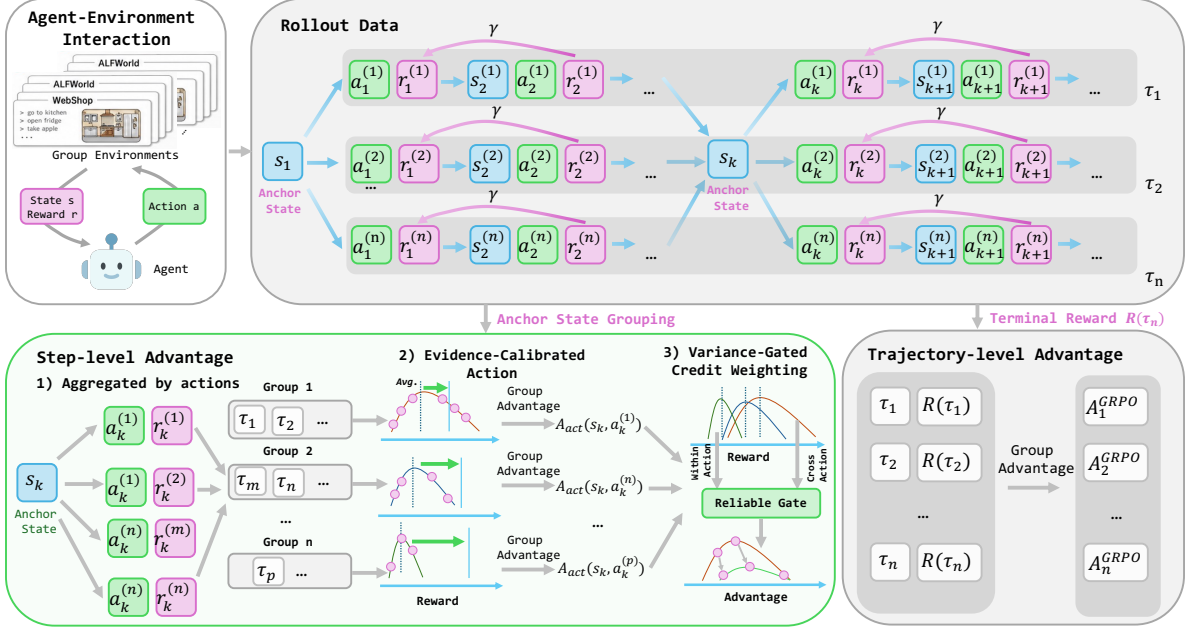


Figure 2: **Overview of ECPO.** ECPO collects grouped rollout trajectories and constructs both trajectory-level and step-level credit. At repeated anchor states, rollout occurrences are grouped by canonical actions; *Evidence-Calibrated Action Advantage* shrinks low-count action estimates, while *Variance-Gated Credit Weighting* down-weights unreliable anchor signals. The calibrated step-level credit is then combined with the trajectory-level group advantage for critic-free policy optimization.

To avoid over-trusting low-count actions, we shrink the empirical action return toward the anchor-level mean:

$$\tilde{\mu}_{s,u} = \frac{n_{s,u}\bar{G}_{s,u} + \kappa\mu_s}{n_{s,u} + \kappa}, \quad (11)$$

where $\tilde{\mu}_{s,u}$ is the calibrated action-level return and $\kappa > 0$ controls shrinkage strength. Smaller $n_{s,u}$ yields stronger shrinkage toward μ_s , while larger $n_{s,u}$ relies more on $\bar{G}_{s,u}$.

We then normalize calibrated action estimates within the same anchor state. The action-level standard deviation is

$$\sigma_s^{\text{act}} = \left(\frac{1}{|\mathcal{U}_s|} \sum_{u \in \mathcal{U}_s} (\tilde{\mu}_{s,u} - \mu_s)^2 \right)^{1/2}. \quad (12)$$

The calibrated action advantage is

$$A_{\text{act}}(s_{i,t}, a_{i,t}) = \frac{\tilde{\mu}_{s_{i,t}, u_{i,t}} - \mu_{s_{i,t}}}{\sigma_{s_{i,t}}^{\text{act}} + \epsilon}, \quad (13)$$

where $\epsilon > 0$ is a numerical constant. All occurrences taking the same canonical action at the same anchor state share the same A_{act} , so step-level credit is based on calibrated action-level evidence rather than noisy individual occurrences.

4.2 Variance-Gated Credit Weighting

Although ECA reduces low-count overestimation, calibrated action estimates should not always be trusted equally. Some anchor states exhibit clear action-dependent return differences, where the current action strongly explains downstream success. Other anchors are dominated by within-action randomness: even trajectories taking the same action may later diverge and obtain different returns. In such cases, injecting step-level credit can still introduce noisy gradients. Therefore, beyond calibrating action values, ECPO must also decide whether an anchor state provides reliable evidence for step-level policy updates.

We introduce *Variance-Gated Credit Weighting*, which decomposes anchor-level return variation into between-action signal and within-action noise. For anchor state s , the between-action variance is

$$B_s = \frac{1}{n_s} \sum_{u \in \mathcal{U}_s} n_{s,u} (\bar{G}_{s,u} - \mu_s)^2, \quad (14)$$

where $n_s = |\mathcal{I}_s|$. A large B_s indicates that action choice explains substantial return variation.

For each canonical action u , the within-action variance is

$$\text{Var}_s(G | u) = \frac{1}{n_{s,u}} \sum_{(i,t) \in \mathcal{I}_{s,u}} (G_{i,t} - \bar{G}_{s,u})^2. \quad (15)$$

The anchor-level within-action variance is

$$W_s = \frac{1}{n_s} \sum_{u \in \mathcal{U}_s} n_{s,u} \text{Var}_s(G | u). \quad (16)$$

Here, W_s captures downstream uncertainty after conditioning on the current action.

To further account for sample size, we define

$$g_s = \tanh\left(\frac{n_s}{\tau}\right), \quad (17)$$

where $\tau > 0$ controls saturation. The final reliability weight is

$$\rho_{\text{VG}}(s) = g_s \cdot \frac{B_s}{B_s + W_s + \epsilon}. \quad (18)$$

When B_s dominates W_s , $\rho_{\text{VG}}(s)$ remains large and the step-level signal is trusted; when W_s dominates, the anchor is noisy and its contribution is suppressed. For non-comparable anchors, such as those with fewer than two canonical actions, the reliability weight is set to zero and the update falls back to trajectory-level credit.

4.3 Policy Optimization with ECPO

The two components above produce calibrated step-level credit while preserving the stability and scalability of group-based RL. ECPO keeps the original trajectory-level group advantage as the global learning signal and injects step-level credit only when the anchor provides reliable action-level evidence. This design retains critic-free optimization, but changes the role of anchor states: they are no longer treated as automatically trustworthy sources of dense supervision, but as statistical evidence whose reliability must be estimated before affecting the policy update.

Let A_i^{GRPO} denote the group-level trajectory advantage of τ_i . For each occurrence (i, t) , ECPO constructs the final advantage as

$$\hat{A}_{i,t}^{\text{ECPO}} = A_i^{\text{GRPO}} + \omega \rho_{\text{VG}}(s_{i,t}) A_{\text{act}}(s_{i,t}, a_{i,t}), \quad (19)$$

where $\omega \geq 0$ controls the strength of calibrated step-level credit. When the anchor is unreliable, $\rho_{\text{VG}}(s_{i,t})$ becomes small and the update falls back to trajectory-level group optimization; otherwise, ECPO uses the calibrated step signal to refine the update.

The textual action $a_{i,t}$ is generated as a token sequence $a_{i,t} = (y_{i,t,1}, \dots, y_{i,t,L_{i,t}})$, where $L_{i,t}$ is the action length. Let $c_{i,t,\ell}$ denote the token context of token $y_{i,t,\ell}$, including the task description,

interaction history, current state, and previous action tokens. The token-level probability ratio is

$$q_{i,t,\ell}(\theta) = \frac{\pi_{\theta}(y_{i,t,\ell} | c_{i,t,\ell})}{\pi_{\theta_{\text{old}}}(y_{i,t,\ell} | c_{i,t,\ell})}. \quad (20)$$

We optimize the policy with the following clipped objective:

$$\begin{aligned} \mathcal{J}_{\text{ECPO}}(\theta) = \mathbb{E}_{i,t,\ell} \Big[& \min \left(q_{i,t,\ell}(\theta) \hat{A}_{i,t}^{\text{ECPO}}, \right. \\ & \left. \text{clip} \left(q_{i,t,\ell}(\theta), 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}} \right) \hat{A}_{i,t}^{\text{ECPO}} \right) \\ & \left. - \beta \mathbb{E}_{i,t,\ell} \left[D_{\text{KL}} \left(\pi_{\theta}(\cdot | c_{i,t,\ell}) \parallel \pi_{\text{ref}}(\cdot | c_{i,t,\ell}) \right) \right] \right]. \end{aligned} \quad (21)$$

where $\epsilon_{\text{clip}} > 0$ is the clipping threshold, π_{ref} is the reference policy, $D_{\text{KL}}(\cdot \parallel \cdot)$ denotes the KL divergence, and $\beta \geq 0$ is the KL coefficient. The same occurrence-level advantage $\hat{A}_{i,t}^{\text{ECPO}}$ is assigned to all tokens of $a_{i,t}$, so the policy promotes or suppresses the entire action according to its evidence-calibrated credit.

5 Experiments and Results

In this section, we evaluate the effectiveness of ECPO on long-horizon agentic tasks and conduct ablation studies to verify the contribution of each component.

5.1 Results on Long-Horizon Agentic Tasks

Experimental setup. We evaluate ECPO on two long-horizon agentic benchmarks: ALFWorld (Shridhar et al., 2020) and WebShop (Yao et al., 2022a). For ALFWorld, we report the average success rate including six subtasks: Pick, Clean, Cool, Look, Heat, and Pick2. For WebShop, we report both the average task score and success rate. All results are averaged over 3 random seeds. Each run is trained for 150 iterations. Qwen2.5-1.5B experiments use 8 A100-80G GPUs for approximately 12 hours, while Qwen2.5-7B experiments use 16 A100-80G GPUs for approximately 16 hours. We use Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Qwen Team et al., 2024) as trainable base policies, and compare ECPO with three groups of baselines: closed-source LLM agents, including *Gemini-3.5-Flash* and *GLM-5.1*; prompt-based open-source agents, including direct prompting, ReAct (Yao et al., 2022b), and Reflexion (Shinn et al., 2023); and RL training baselines, including PPO with a critic (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), and

Table 1: **Main results on ALFWorld and WebShop.** For ALFWorld, we report success rates for each subtask as well as the overall result. For WebShop, we report the average task score and success rate. Results are averaged over 3 random seeds. Within each trainable model block, the best result is in **bold** and the second-best result is underlined.

Method	ALFWorld							WebShop	
	Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Success
<i>Closed-source LLMs</i>									
Gemini-3.5-Flash	91.2 $\pm$ 3.9	67.1 $\pm$ 2.5	33.3 $\pm$ 6.8	51.5 $\pm$ 4.3	33.3 $\pm$ 7.8	57.5 $\pm$ 1.5	57.6 $\pm$ 3.3	5.8 $\pm$ 1.4	3.9 $\pm$ 0.6
GLM-5.1	92.9 $\pm$ 5.2	82.9 $\pm$ 1.6	66.7 $\pm$ 3.4	72.7 $\pm$ 7.4	85.4 $\pm$ 5.9	65.0 $\pm$ 2.4	77.3 $\pm$ 1.7	12.9 $\pm$ 4.2	8.1 $\pm$ 3.3
<i>Qwen2.5-1.5B-Instruct</i>									
Prompting	5.9	3.3	4.2	5.5	9.7	0.0	4.1	23.1	5.2
ReAct	17.4	15.7	7.7	20.5	6.2	2.0	12.8	40.1	11.3
Reflexion	35.3	21.7	19.4	22.2	13.6	3.7	21.8	55.8	21.9
PPO	64.8 $\pm$ 3.5	57.1 $\pm$ 4.9	46.4 $\pm$ 4.0	40.5 $\pm$ 6.9	60.6 $\pm$ 6.6	47.4 $\pm$ 1.9	54.4 $\pm$ 3.1	73.8 $\pm$ 3.0	51.5 $\pm$ 2.9
RLOO	<u>88.3$\pm$3.0</u>	71.0 $\pm$ 5.9	66.4 $\pm$ 5.5	52.8 $\pm$ 8.6	62.8 $\pm$ 8.7	56.9 $\pm$ 4.7	69.7 $\pm$ 2.5	73.9 $\pm$ 5.6	52.1 $\pm$ 6.7
GRPO	85.3 $\pm$ 1.5	84.5 $\pm$ 6.8	59.7 $\pm$ 5.0	53.7 $\pm$ 8.0	78.2 $\pm$ 7.9	53.5 $\pm$ 5.6	72.8 $\pm$ 3.6	64.1 $\pm$ 5.7	52.6 $\pm$ 4.3
GiGPO _{w/std}	83.2 $\pm$ 4.6	<u>94.6$\pm$4.0</u>	<u>84.1$\pm$5.4</u>	<u>82.2$\pm$1.6</u>	<u>84.2$\pm$0.0</u>	91.7$\pm$4.7	<u>87.5$\pm$0.6</u>	<u>80.4$\pm$4.2</u>	<u>64.6$\pm$3.1</u>
ECPO	96.7$\pm$2.7	95.7$\pm$1.5	87.0$\pm$0.0	94.4$\pm$7.9	100.0$\pm$0.0	<u>81.7$\pm$2.4</u>	92.7$\pm$0.7	84.4$\pm$3.2	71.9$\pm$1.7
<i>Qwen2.5-7B-Instruct</i>									
Prompting	33.4	19.3	2.8	21.6	6.9	3.2	14.8	26.4	7.8
ReAct	48.5	34.3	18.2	35.4	13.2	17.6	31.2	46.2	19.5
Reflexion	62.0	44.9	36.3	41.6	30.9	23.8	42.7	58.1	28.8
PPO	92.3 $\pm$ 4.0	92.5 $\pm$ 2.4	<u>80.3$\pm$2.0</u>	64.0 $\pm$ 8.4	<u>89.5$\pm$7.0</u>	68.8 $\pm$ 8.3	80.4 $\pm$ 2.7	81.4 $\pm$ 3.1	68.7 $\pm$ 5.1
RLOO	87.6 $\pm$ 4.3	87.3 $\pm$ 5.8	71.9 $\pm$ 5.2	78.2 $\pm$ 8.3	81.3 $\pm$ 7.6	48.9 $\pm$ 8.4	75.5 $\pm$ 4.6	80.3 $\pm$ 3.2	65.7 $\pm$ 4.0
GRPO	90.8 $\pm$ 5.1	89.3 $\pm$ 5.4	72.5 $\pm$ 5.4	66.1 $\pm$ 6.7	74.7 $\pm$ 6.9	64.7 $\pm$ 7.3	77.6 $\pm$ 5.2	79.3 $\pm$ 2.8	66.1 $\pm$ 3.7
GiGPO _{w/std}	97.7$\pm$1.6	<u>98.8$\pm$1.6</u>	89.3$\pm$8.2	82.7$\pm$7.9	83.7 $\pm$ 7.2	<u>79.2$\pm$6.6</u>	<u>90.8$\pm$1.3</u>	<u>84.4$\pm$0.5</u>	<u>72.4$\pm$2.6</u>
ECPO	<u>94.4$\pm$1.6</u>	100.0$\pm$0.0	79.7 $\pm$ 2.1	<u>82.2$\pm$1.6</u>	91.2$\pm$2.5	93.3$\pm$2.4	91.9$\pm$0.7	85.5$\pm$0.5	74.7$\pm$3.0

GiGPO (Feng et al., 2025). Unless otherwise specified, we follow the training configuration of GiGPO, with a rollout group size of 8. For ECPO, we set the shrinkage strength to $\kappa = 2.0$ and the VarGate temperature to $\tau = 2.0$. We further use $d_{\min} = 7$ as the depth threshold and $\rho_{\min} = 0.5$ as the reliability floor for StrictValid anchors. Full hyperparameter settings are provided in Appendix A. We additionally evaluate generalization to search-augmented QA in Section 5.3. We also analyze the training dynamics in Appendix B.1 and the cost of training time in Appendix 6.2.

Experiment results. Table 1 reports the main results on ALFWorld and WebShop. We summarize three findings. **1) Evidence-calibrated credit improves long-horizon agent performance.** ECPO achieves the best overall performance under the Qwen2.5-1.5B setting. On ALFWorld, it improves the overall success rate from 87.5 to **92.7**, and on WebShop it improves the success rate from 64.6 to **71.9**. The gains over GiGPO show that denser step-level credit alone is not sufficient; calibrating the reliability of such credit leads to better policy learning in long-horizon environments. **2) The**

gains are consistent with our divergent-anchor diagnosis. GiGPO assigns step-level advantages from raw future returns, which can over-reward rare but lucky actions when anchor groups are imbalanced. ECPO directly addresses this issue by shrinking low-count action estimates and gating unreliable anchor states. This produces more stable optimization: as shown in Figure 1, ECPO reduces the final reward standard deviation from $\sigma = 0.746$ to $\sigma = 0.555$. Thus, the improvement is not merely due to adding another step-level signal, but to making the step-level signal statistically more reliable. **3) ECPO remains effective across model scales and tasks.** The advantage of ECPO also holds when scaling to Qwen2.5-7B. On WebShop, it further improves over GiGPO in both task score and success rate, and on ALFWorld it achieves the best overall 7B result in Table 1. These results suggest that evidence-calibrated policy optimization provides a general mechanism for stabilizing group-based RL, especially in settings where repeated anchor states contain imbalanced and noisy action evidence.

Table 2: **Ablation study on ALFWorld with Qwen2.5-1.5B-Instruct.** We report success rates averaged over 3 random seeds. “ECA” and “VG” indicate whether Evidence-Calibrated Action Advantage and Variance-Gated Credit Weighting are enabled.

Method	ECA	VG	All	Pick	Pick2	Look	Heat	Cool	Clean
GiGPO	–	–	87.5±0.6	83.2±4.6	91.7 ±4.7	82.2±1.6	84.2±0.0	84.1±5.4	94.6±4.0
ECA-only	✓	–	89.8±0.6	88.8±1.5	90.0±0.0	82.2±1.6	93.0±2.5	82.6±3.6	95.6±1.6
VarGate-only	–	✓	89.6±1.3	84.3±4.0	90.0±4.1	82.2±1.6	89.5±0.0	87.0 ±0.0	97.8 ±1.6
ECPO	✓	✓	92.7 ±0.7	96.7 ±2.7	81.7±2.4	94.4 ±7.9	100.0 ±0.0	87.0 ±0.0	95.7±1.5

5.2 Ablation Study

Experimental setup. We conduct ablation studies on ALFWorld with Qwen2.5-1.5B-Instruct. All variants follow the same training and evaluation protocol as the main experiments, and results are averaged over 3 random seeds on the `valid_seen` split. We compare four variants: GiGPO as the baseline, **+ECA** which replaces occurrence-level anchor credit with Evidence-Calibrated Action Advantage but does not use VarGate, **+VarGate** which applies Variance-Gated Credit Weighting to raw occurrence-level anchor credit without ECA shrinkage, and the full ECPO with both components.

Experiment results. Table 2 reports the ablation results. We summarize three findings. **1) Each component independently improves GiGPO.** Adding ECA alone improves the overall success rate from 87.5 to 89.8, showing that action-level shrinkage effectively reduces the bias of low-count anchor estimates. Adding VarGate alone improves the overall success rate to 89.6, indicating that variance-based reliability gating can suppress noisy anchor states even without action-level shrinkage. These results validate the two sources of instability identified in our analysis: small-sample action estimation and unreliable anchor-level credit. **2) ECA and VarGate are complementary.** The full ECPO achieves the best overall success rate of **92.7**, outperforming GiGPO by **+5.2** points. This gain is larger than either component alone and also exceeds their simple individual improvements, suggesting a synergistic effect. ECA first calibrates action-level evidence, while VarGate further decides whether the calibrated step signal is reliable enough to affect the policy update. **3) The full method mainly improves tasks sensitive to unstable step credit.** ECPO brings large gains on Pick, Look, and Heat, where long-horizon dependencies make noisy step-level credit especially harmful. At the same time, Pick2 decreases compared with GiGPO, suggesting that

some subtasks may already benefit from the original anchor signal and require less aggressive calibration. Overall, the ablation confirms our central claim: reliable step-level credit requires both action-level evidence calibration and anchor-level reliability control.

5.3 Generalization to Search-Augmented QA

We further evaluate ECPO on search-augmented QA, where an agent must decide whether to search or answer within four interaction turns. We train Qwen2.5-3B-Instruct on NQ (Kwiatkowski et al., 2019) and HotpotQA (Yang et al., 2018), and evaluate on in-domain and out-of-domain single-hop and multi-hop datasets. For a controlled comparison, GiGPO and ECPO use the same training protocol: 256 training examples, rollout group size 5, actor learning rate 1×10^{-6} , and 200 iterations on 8 A100 GPUs.

Table 3 shows that ECPO improves the average over GiGPO from 42.5 to **43.4**. The clearest gains are on out-of-domain single-hop datasets, where PopQA improves from 41.5 to 44.1 and TriviaQA from 59.5 to 60.7. The smaller gain than on ALFWorld and WebShop is consistent with our diagnosis: evidence calibration is most useful in longer-horizon environments with more repeated anchors and more imbalanced action evidence.

6 Discussion

6.1 Effect of Rollout Group Size

Experimental setup. We further study how the rollout group size affects step-level credit assignment. We vary the group size as $N \in \{4, 8, 10\}$ on ALFWorld with Qwen2.5-1.5B-Instruct, while keeping all other training hyperparameters unchanged. For each setting, we compare GiGPO and ECPO at training step 150 and report the validation success rate. We also analyze optimization stability using the rolling standard deviation of validation success rate, the maximum single-step success-rate jump, and the average advantage range.

Experiment results. Table 4 shows that ECPO consistently improves over GiGPO under $N \in \{4, 8, 10\}$, with success-rate gains of **+4.7**, **+5.2**, and **+3.9** points, respectively. The largest gain occurs at $N = 8$, where GiGPO is stronger than at $N = 4$ but retains unstable step-level credit. This matches our diagnosis: once the policy becomes more capable, repeated anchor states be-

Table 3: **Performance on search-augmented QA.** Models are trained on NQ and HotpotQA. † and * denote in-domain and out-of-domain datasets. Best results in each group are bolded.

Method	Single-Hop QA			Multi-Hop QA				Avg.
	NQ [†]	TriviaQA [*]	PopQA [*]	HotpotQA [†]	2Wiki [*]	MuSiQue [*]	Bamboogle [*]	
R1-Instruct	27.0	53.7	19.9	23.7	29.2	7.2	29.3	27.1
Search-R1	34.1	54.5	37.8	32.4	31.9	10.3	26.4	32.5
ZeroSearch	41.4	57.4	44.8	27.4	30.0	9.8	11.1	31.7
StepSearch	–	–	–	34.5	32.0	17.4	34.4	–
GiGPO	44.0	59.5	41.5	37.0	37.0	12.1	26.4	42.5
ECPO	43.7	60.7	44.1	37.6	36.6	12.7	27.2	43.4

Table 4: **Effect of rollout group size on ALFWorld.** We evaluate GiGPO and ECPO with $N \in \{4, 8, 10\}$ rollouts using Qwen2.5-1.5B-Instruct. ECPO improves the final success rate across different rollout budgets.

Group Size N	GiGPO	ECPO	Δ
4	71.9%	76.6%	+4.7pp
8	87.5%	92.7%	+5.2pp
10	90.6%	94.5%	+3.9pp

Table 5: **Per-step training time.** Only the advantage-computation difference is attributable to ECPO; generation time varies with sampled trajectory length.

Method	Adv.	Gen.	Actor	Total	Extra Cost
GRPO	0.53s	183.0s	35.2s	282.9s	–
GiGPO	0.66s	207.7s	30.2s	302.8s	–
ECPO	0.96s	232.6s	30.0s	339.2s	Adv.: +0.30s

come more frequent, but the sampled evidence at each anchor may still be imbalanced, making rare lucky actions influential in policy updates. Increasing rollouts alone does not solve the problem: GiGPO remains below ECPO at every group size. GiGPO’s largest single-step jump occurs at $N = 8$, revealing influential yet unreliable anchor signals. ECPO mitigates this failure: ECA suppresses small-sample action bias, while VarGate filters noisy anchors and large policy jumps. Overall, reliable credit construction matters more than simply increasing rollout count for stable long-horizon training. Additional group-size results appear in Appendix B.2.

6.2 Training-Time Efficiency

Table 5 isolates evidence-calibration cost. Advantage computation rises from 0.66s to 0.96s over GiGPO (+0.30s), only **0.10%** of total step time, while actor-update time is unchanged. ECPO adds no critic or optimizer update. Because genera-

tion varies with trajectory length, only the 0.30s difference is attributable overhead; rollout generation remains the bottleneck. The small increment reflects that ECPO reuses collected rollouts and computes only lightweight statistics over anchor groups, avoiding the memory and synchronization costs of maintaining a learned value model. This design also simplifies integration into critic-free pipelines: calibration operates on rollout metadata before the policy update and requires neither additional model placement nor a separate gradient-synchronization path. The measured overhead is therefore lightweight post-processing rather than an additional learning component. Moreover, calibration uses materialized scalar returns and action identifiers, so its cost does not scale with model parameters in the same way as actor optimization.

7 Conclusion

We identify divergent anchor bias as a key source of instability in step-level credit assignment for long-horizon LLM agents: denser credit can become harmful when rare lucky actions are over-rewarded under limited rollouts. To address this issue, we propose ECPO, which calibrates action-level evidence through shrinkage and gates unreliable anchor states via variance decomposition. Experiments on ALFWorld and WebShop show that ECPO improves over GiGPO by **+5.2** and **+7.3** success points with Qwen2.5-1.5B, while reducing late-stage reward variance and adding only **0.1%** advantage-computation overhead. Search-augmented QA results further show that the method transfers beyond embodied and web-interaction environments. These results suggest that reliable credit construction is crucial for stable long-horizon agent training.

Limitations

Although ECPO improves the reliability of step-level credit assignment, it still has several limitations. First, our method relies on repeated anchor states to construct calibrated step-level advantages. In environments where state repetition is rare or where states are difficult to canonicalize, the amount of usable anchor-level evidence may be limited, and ECPO may fall back more often to trajectory-level advantages. Second, our experiments focus on ALFWorld, WebShop and SearchQA with Qwen2.5-based agents. While these benchmarks cover embodied planning, web interaction and QA, broader evaluation on more diverse tool-use, multi-agent, and real-world deployment settings is needed to further validate the generality of evidence-calibrated credit assignment.

Ethics Statement

This work studies reinforcement learning methods for improving long-horizon LLM agents. The proposed method aims to make policy optimization more stable and reliable by reducing noisy credit assignment, and does not introduce new tools, external data collection, or user-facing autonomous capabilities beyond the evaluated benchmark environments. However, stronger LLM agents may also increase risks when deployed in open-ended settings, especially if they are connected to real tools or services. Therefore, practical deployment should include appropriate safety constraints, human oversight, logging, and environment-specific access control. All experiments are conducted on standard research benchmarks, and no private or sensitive user data is used.

Acknowledgments

This work was supported by the China NSFC Projects (62576212, 92370206, 62120106006, U23B2057) and the Frontier Technologies R&D Program of Jiangsu (BF2025029).

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. 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)*, pages 12248–12267.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *Preprint*, arXiv:2204.05862.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Guanting Dong, Yifei Chen, Xiaoxi Li, Jiajie Jin, Hongjin Qian, Yutao Zhu, Hangyu Mao, Guorui Zhou, Zhicheng Dou, and Ji-Rong Wen. 2025a. Tool-star: Empowering llm-brained multi-tool reasoner via reinforcement learning. *arXiv preprint arXiv:2505.16410*.

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

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. [Group-in-group policy optimization for llm agent training](#). *Advances in Neural Information Processing Systems*, 38:51797–51830.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. [Interactive fiction games: A colossal adventure](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7903–7910.

Ji He, Jianshu Chen, Xiaodong He, Jianfeng Gao, Lihong Li, Li Deng, and Mari Ostendorf. 2016. Deep reinforcement learning with a natural language action space. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1621–1630.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. 2025. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

- Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. 2025. [Cppo: Accelerating the training of group relative policy optimization-based reasoning models](#). *Advances in Neural Information Processing Systems*, 38:67902–67927.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.
- Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. 2015. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1–11.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. 2023. A survey of temporal credit assignment in deep reinforcement learning. *arXiv preprint arXiv:2312.01072*.
- Qwen Team, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2024. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. 2024. Deepseek-math: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.
- Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. 2026. Hindsight credit assignment for long-horizon llm agents. *arXiv preprint arXiv:2603.08754*.
- Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16022–16076.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. 2025. [Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning](#). *Preprint*, arXiv:2504.20073.
- Zhepei Wei, Wenlin Yao, Yao Liu, Weizhi Zhang, Qin Lu, Liang Qiu, Changlong Yu, Puyang Xu, Chao Zhang, Bing Yin, Hyokun Yun, and Lihong Li. 2025. [WebAgent-r1: Training web agents via end-to-end multi-turn reinforcement learning](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 7909–7928, Suzhou, China. Association for Computational Linguistics.
- Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering.

In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.

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

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022b. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. 2025. [Dapo: An open-source llm reinforcement learning system at scale](#). *Advances in Neural Information Processing Systems*, 38:125532–125554.

Siwei Zhang, Yun Xiong, Xi Chen, Zi’an Jia, Renhong Huang, Jiarong Xu, and Jiawei Zhang. 2026. Rapo: Expanding exploration for llm agents via retrieval-augmented policy optimization. *arXiv preprint arXiv:2603.03078*.

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

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*.

A Experimental Details

A.1 Details of Training

Hyperparameters for ALFWorld. All methods are configured with identical basic hyperparameters for fair comparison. The maximum prompt length is 2048 tokens, and the maximum response length is 512 tokens. Each episode allows up to 50 environment steps. The learning rate is set to 1×10^{-6} for the actor and 1×10^{-5} for the critic, where the critic is used only in PPO. We adopt a rule-based reward: a reward of 10 is assigned for task success and 0 for failure, and invalid actions receive a penalty of -0.1 . For all group-based RL methods, we use a group size of 8 and sample 16 groups per rollout, resulting in $16 \times 8 = 128$ parallel environments. PPO instead uses 128 separate

environments for rollouts. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 256, and the KL-divergence loss coefficient is 0.01. For step-level credit methods, the weighting coefficient ω is fixed at 1, and the discount factor γ is set to 0.95. For ECPO, we set the shrinkage strength to $\kappa = 2.0$, the VarGate temperature to $\tau = 2.0$, the depth threshold to $d_{\min} = 7$, and the reliability floor to $\rho_{\min} = 0.5$.

Hyperparameters for WebShop. All methods are configured with identical basic hyperparameters. The maximum prompt length is 4096 tokens, and the maximum response length is 512 tokens. Each episode is limited to 15 environment steps. The learning rate is 1×10^{-6} for the actor and 1×10^{-5} for the critic, where the critic is used only in PPO. We use the same rule-based reward as ALFWorld: 10 for success and 0 for failure, with a penalty of -0.1 for invalid actions. All group-based RL methods use a group size of 8 and sample 16 groups per rollout, totaling $16 \times 8 = 128$ parallel environments. PPO uses 128 distinct environments for rollouts. The rollout temperature is set to 1.0, while the validation temperature is set to 0.4. The mini-batch size is 64, and the KL-divergence loss coefficient is 0.01. For step-level credit methods, the weighting coefficient ω is fixed at 1, and the discount factor γ is set to 0.95. For ECPO, we use the same evidence-calibration hyperparameters as ALFWorld: $\kappa = 2.0$, $\tau = 2.0$, $d_{\min} = 7$, and $\rho_{\min} = 0.5$.

Computing details. For ALFWorld and WebShop, all Qwen2.5-1.5B experiments are trained on 8 A100-80G GPUs with 12 hours, and all Qwen2.5-7B experiments are trained on 16 A100-80G GPUs with 16 hours. Each experiment is trained for 150 iterations. All reported results are averaged over 3 random seeds.

A.2 Implementation Details of Anchor Validation

StrictValid anchors. In implementation, calibrated step-level credit is applied only to anchors that pass a lightweight validity check. An anchor is considered StrictValid if it satisfies four conditions: it appears after the shallow exploration stage, has repeated visits, contains at least two canonical actions, and has non-degenerate return

variation. Formally, for anchor state s , we use

$$\text{StrictValid}(s) = 1 \left[\text{depth}(s) \geq d_{\min}, n_s \geq 2, |\mathcal{U}_s| \geq 2, B_s + W_s > \epsilon \right]. \quad (22)$$

If an anchor does not pass this check, we set its step-level reliability weight to zero and the update falls back to the trajectory-level advantage. For anchors passing the check, we optionally apply a reliability floor:

$$\rho_{\text{VG}}(s) \leftarrow \max(\rho_{\text{VG}}(s), \rho_{\min}). \quad (23)$$

This prevents reliable anchors from being overly suppressed by small numerical fluctuations. In our experiments, we set $d_{\min} = 7$ and $\rho_{\min} = 0.5$. These two parameters are used only for implementation stability; the core method is defined by the ECA shrinkage estimator and the VarGate reliability weight.

A.3 Prompt Templates

ALFWorld prompt. For ALFWorld, we use the following prompt template with recent interaction history and admissible actions.

ALFWorld Prompt Template

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.

WebShop prompt. For WebShop, we use the following prompt template with recent interaction history and available actions.

WebShop Prompt Template

You are an expert autonomous agent operating in the WebShop e-commerce 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 for

the current situation are: [{available_actions}].

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 `<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.

B Additional Results

B.1 Training Dynamics Analysis

Final success rates show the endpoint performance of an agent, but they do not reveal whether the policy reaches this performance through stable learning or through noisy late-stage updates. To examine the optimization process more directly, we analyze both the validation curves and the variance of step-level advantage estimates. This analysis is designed to support our central claim: the benefit of ECPO comes not merely from adding denser step-level credit, but from making such credit statistically more reliable.

Training dynamics. Figure 3a compares the learning curves of GiGPO and ECPO. On ALFWorld, the two methods behave similarly in the early stage, but ECPO begins to separate from GiGPO after around step 60 and finally reaches 92.7% validation success rate, compared with 87.5% for GiGPO. Its shaded region is also narrower, indicating more stable convergence. On WebShop, ECPO maintains a consistent advantage in the middle and late stages, and the gap continues to enlarge as training proceeds, ending at 71.9% versus 64.6%. This pattern suggests that the benefit of ECPO is cumulative: more reliable gradients lead to more persistent policy improvement over training.

The reward curves further support this interpretation. Compared with GiGPO, ECPO achieves a higher reward trajectory with a narrower rolling standard deviation. This indicates that the policy updates produced by ECPO are more directionally consistent and less affected by noisy anchor-level estimates. Thus, ECPO improves not only the final success rate, but also the learning efficiency and stability of the whole training process.

Advantage-variance diagnostic. Figure 3b isolates the effect of ECA on step-level advantage estimation. The standard deviation of ECA-calibrated advantages remains consistently lower than that of raw GiGPO-style anchor advantages throughout training, showing that ECA is effective

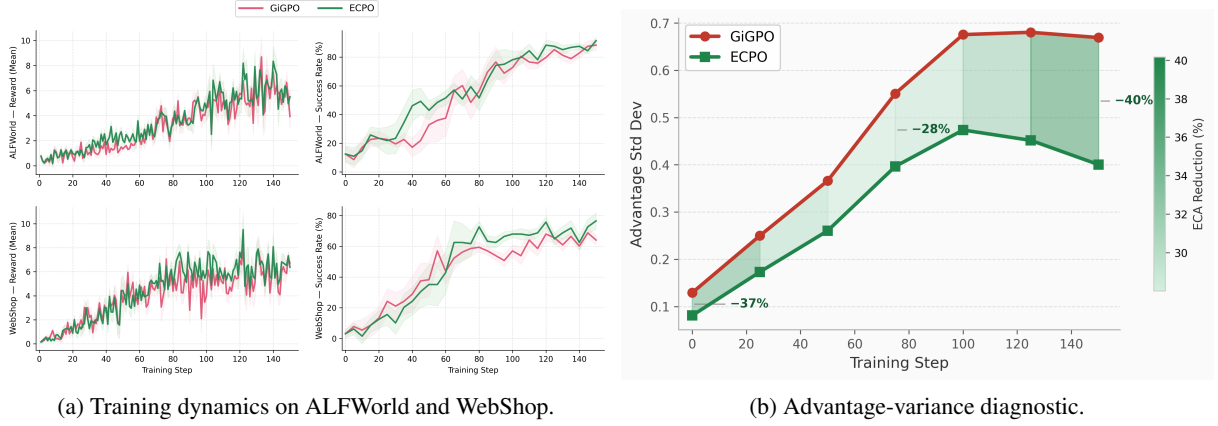


Figure 3: **Training dynamics and advantage-variance diagnostics.** **Left:** ECPO achieves higher final validation performance and more stable reward trajectories on both ALFWorld and WebShop. **Right:** ECA consistently reduces the standard deviation of step-level advantages throughout training, with stronger correction when divergent anchors become more frequent. Together, these results show that evidence-calibrated credit improves not only final performance but also the stability of policy optimization.

across the entire optimization process rather than only at a particular stage. Although both curves increase as training progresses, reflecting increasingly deterministic policies and larger action-level differences, ECA keeps the advantage variance at a lower level.

More importantly, the reduction becomes stronger again in the later stage. The relative reduction is 37% at step 0, decreases to 28% around step 75, and rises to 40% by step 150. This late-stage strengthening coincides with the period where divergent anchors become more frequent, indicating that ECA automatically applies stronger correction when small-sample action estimates become more problematic. This behavior directly matches the design of ECA: low-count actions receive stronger shrinkage toward the anchor-level mean, preventing rare but lucky actions from producing high-variance advantages.

Overall, Figure 3 connects the mechanism and the empirical outcome. The diagnostic plot verifies that ECA reduces the variance of step-level credit, while the training curves show that this variance reduction translates into smoother optimization and higher final performance. These results support our story that reliable credit construction, rather than simply denser step-level supervision, is the key to stable long-horizon LLM agent training.

B.2 Additional Analysis on Small-Rollout Evidence Calibration

To further understand why ECPO remains effective under small rollout budgets, we analyze two diag-

nostic signals: the VarGate reliability weight ρ_{VG} and the advantage range. These diagnostics complement the group-size discussion in Section 6.1 by showing how ECPO adapts its credit correction when the amount of anchor-level evidence changes.

VarGate adapts to rollout evidence. Figure 4a shows the training dynamics of the VarGate reliability weight ρ_{VG} under $N = 4$ and $N = 8$. Since $\rho_{VG} \in [0, 1]$ measures how reliable the anchor-level step signal is, a smaller value means that ECPO relies less on step-level credit and more on trajectory-level group advantage. The $N = 4$ curve remains consistently lower than the $N = 8$ curve, with average reliability weights of 0.538 and 0.720, respectively. This indicates that when fewer rollouts are available, each anchor state contains less action-level evidence, and VarGate automatically detects this uncertainty by reducing the contribution of step-level advantages. Thus, ECPO does not require manually changing the step-credit weight for different rollout budgets; its reliability gate provides an adaptive response to finite-sample noise.

ECA is more effective when small-sample bias is stronger. Figure 4b compares the advantage range, defined as the maximum advantage minus the minimum advantage within a batch. A larger range indicates more dispersed advantage estimates, where some actions may be excessively rewarded while others are excessively penalized, leading to less reliable gradient directions. When $N = 4$, GiGPO has an average advantage range

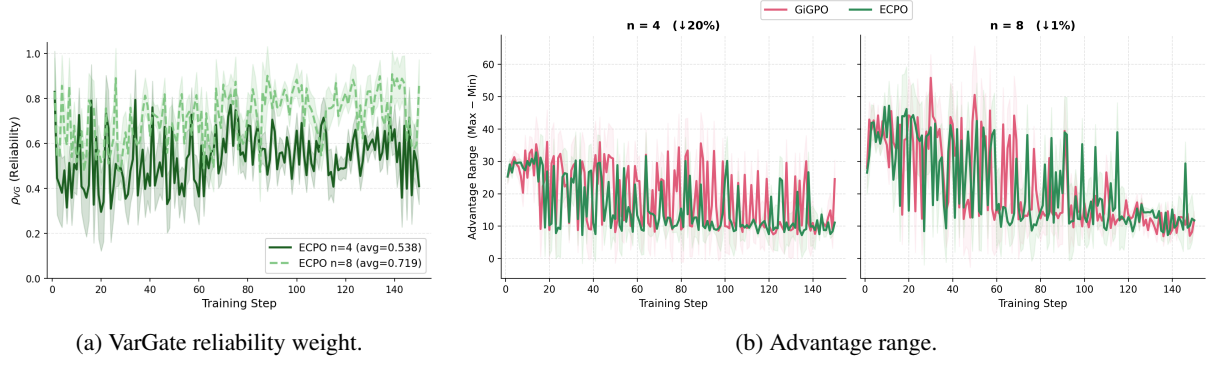


Figure 4: **Diagnostics under different rollout group sizes.** **Left:** VarGate assigns lower reliability weights under smaller rollout budgets, indicating that ECPO automatically downweights unreliable step-level credit. **Right:** ECA reduces the advantage range more clearly when $N = 4$, showing stronger correction under more severe small-sample bias.

of 18.65, while ECPO reduces it to 14.97, corresponding to a 20% reduction. This shows that ECA shrinkage effectively suppresses extreme estimates caused by low-count actions. When $N = 8$, the ranges of GiGPO and ECPO become much closer, 21.35 versus 21.03, suggesting that action-level estimates are already less sparse and leave less room for shrinkage correction.

These diagnostics support the mechanism of ECPO under different rollout budgets. When N is small, the dominant issue is unreliable action estimation; ECA reduces extreme action advantages, while VarGate lowers the overall trust in anchor-level step signals. When N becomes larger, the action estimates become more stable, so ECA intervenes less aggressively, while VarGate continues to filter noisy anchors. Therefore, the benefit of ECPO is not a fixed heuristic effect: it adapts to the amount and reliability of rollout evidence, providing stronger correction precisely when small-sample bias is more severe.

C Pseudo Code of ECPO

Algorithm 1 presents the training procedure of ECPO.

D Theoretical Analysis of Evidence Calibration

This section provides a theoretical analysis of why raw occurrence-level anchor credit can over-reward rare but lucky actions, and why ECPO reduces this effect through evidence calibration.

D.1 Finite Rollouts Cannot Reliably Identify Action Quality

Consider an anchor state s and a canonical action u . Let the binary downstream success variable after taking action u be $Y \in \{0, 1\}$, with posterior success probability

$$p_u = \Pr(Y = 1 \mid s, u). \quad (24)$$

The ideal step-level credit should depend on p_u , because p_u measures the true expected downstream success of action u at state s . However, in group-based agentic RL, p_u is unknown and must be estimated from only $n_{s,u}$ rollout occurrences. When $n_{s,u}$ is small, especially when $n_{s,u} = 1$, the empirical estimate

$$\hat{p}_u = \frac{1}{n_{s,u}} \sum_{j=1}^{n_{s,u}} Y_j \quad (25)$$

is a high-variance point estimate.

This limitation is information-theoretic and does not depend on a specific algorithm. Consider two possible posterior success probabilities p and $p + \Delta$. From n Bernoulli samples, any method that tries to distinguish $\text{Bernoulli}(p)$ from $\text{Bernoulli}(p + \Delta)$ has error probability lower bounded by Le Cam’s inequality:

$$P_{\text{err}} \geq \frac{1}{2} (1 - \text{TV}(P_p^n, P_{p+\Delta}^n)), \quad (26)$$

where P_p^n denotes the distribution of n i.i.d. Bernoulli samples with success probability p . Using Pinsker’s inequality,

$$\begin{aligned} & \text{TV}(P_p^n, P_{p+\Delta}^n) \\ & \leq \sqrt{\frac{n}{2} D_{\text{KL}}(\text{Bern}(p) \parallel \text{Bern}(p + \Delta))}. \end{aligned} \quad (27)$$

For small Δ , the KL divergence satisfies

$$D_{\text{KL}}(\text{Bern}(p) \parallel \text{Bern}(p + \Delta)) = O\left(\frac{\Delta^2}{p(1-p)}\right). \quad (28)$$

Therefore, when

$$\Delta = O\left(\sqrt{\frac{p(1-p)}{n}}\right), \quad (29)$$

the two posterior success probabilities cannot be reliably distinguished. Thus, with small $n_{s,u}$, no distribution-free estimator can consistently determine whether an action is truly good or only lucky from rollout observations alone. This motivates calibrating the observed action-level evidence before using it for policy optimization.

D.2 Expected Over-Rewarding of Rare Lucky Actions in GiGPO

We now analyze a typical divergent anchor. Assume that at anchor state s , there are two canonical actions: a frequent good action u^+ and a rare bad action u^- . The good action is sampled m times, while the bad action is sampled once:

$$n_{s,u^+} = m, \quad n_{s,u^-} = 1. \quad (30)$$

Let their true posterior success probabilities be

$$\begin{aligned} p_+ &= \Pr(Y = 1 \mid s, u^+), \\ p_- &= \Pr(Y = 1 \mid s, u^-), \quad p_+ > p_-. \end{aligned} \quad (31)$$

Let $K \sim \text{Binomial}(m, p_+)$ denote the number of successful rollouts among the m occurrences of u^+ , and let $Y^- \sim \text{Bernoulli}(p_-)$ denote the observed outcome of the singleton bad action.

GiGPO normalizes raw future returns at the occurrence level. Conditioned on the event that the bad action succeeds once, i.e., $Y^- = 1$, the anchor-level mean return is

$$\mu_K = \frac{K+1}{m+1}. \quad (32)$$

For binary future returns, the corresponding anchor-level standard deviation is

$$\sigma_K = \sqrt{\mu_K(1-\mu_K)}. \quad (33)$$

The GiGPO advantage assigned to the singleton bad action is then

$$A_{\text{GiGPO}}^-(K) = \frac{1 - \mu_K}{\sigma_K + \epsilon}. \quad (34)$$

The average GiGPO advantage of the frequent good action is

$$\bar{A}_{\text{GiGPO}}^+(K) = \frac{K/m - \mu_K}{\sigma_K + \epsilon}. \quad (35)$$

Therefore, when the singleton bad action succeeds and the frequent good action is not perfectly successful, i.e., $Y^- = 1$ and $K < m$, GiGPO assigns the bad action a higher average step advantage:

$$A_{\text{GiGPO}}^-(K) - \bar{A}_{\text{GiGPO}}^+(K) = \frac{1 - K/m}{\sigma_K + \epsilon} > 0. \quad (36)$$

Thus, although $p_- < p_+$, the bad action can receive a larger step-level advantage due to a lucky singleton outcome.

The probability of this over-rewarding event is

$$\Pr(A_{\text{GiGPO}}^- > \bar{A}_{\text{GiGPO}}^+) = p_- (1 - p_+^m). \quad (37)$$

The expected positive advantage gap assigned to the bad action is

$$\begin{aligned} \mathbb{E}[\Delta_{\text{GiGPO}}^-] &= p_- \sum_{k=0}^{m-1} \binom{m}{k} p_+^k (1 - p_+)^{m-k} \\ &\quad \cdot \frac{1 - k/m}{\sqrt{\mu_k(1 - \mu_k)} + \epsilon}, \end{aligned} \quad (38)$$

where

$$\mu_k = \frac{k+1}{m+1}. \quad (39)$$

Equation (38) shows that the expected erroneous advantage is strictly positive whenever $p_- > 0$ and $p_+ < 1$. Therefore, raw occurrence-level credit can systematically over-reward rare but lucky actions under finite rollout budgets.

D.3 Expected Reduction under ECPO

ECPO reduces this over-rewarding effect in two steps. First, Evidence-Calibrated Action Advantage shrinks low-count action estimates toward the anchor-level mean. For the singleton bad action with observed return $Y^- = 1$, the ECA-calibrated return is

$$\tilde{\mu}_{s,u^-} = \frac{1 + \kappa \mu_K}{1 + \kappa}. \quad (40)$$

Hence its centered evidence becomes

$$\tilde{\mu}_{s,u^-} - \mu_K = \frac{1 - \mu_K}{1 + \kappa}. \quad (41)$$

Compared with the raw centered evidence $1 - \mu_K$ used by GiGPO, the singleton bad action is shrunk by a factor of

$$\frac{1}{1 + \kappa}. \quad (42)$$

With the default $\kappa = 2$, this factor is $1/3$.

Second, Variance-Gated Credit Weighting multiplies the calibrated action advantage by the reliability weight

$$\rho_{\text{VG}}(s) = g_s \cdot \frac{B_s}{B_s + W_s + \epsilon}, \quad (43)$$

$$0 \leq \rho_{\text{VG}}(s) \leq 1.$$

Thus, under the same local normalization scale, the erroneous singleton bad-action advantage satisfies

$$\begin{aligned} \mathbb{E} [\Delta_{\text{ECPO}}^-] &\lesssim \mathbb{E} \left[\frac{\rho_{\text{VG}}(s)}{1 + \kappa} \Delta_{\text{GiGPO}}^- \right] \\ &\leq \frac{1}{1 + \kappa} \mathbb{E} [\Delta_{\text{GiGPO}}^-]. \end{aligned} \quad (44)$$

Therefore, ECPO provably reduces the expected positive advantage assigned to rare lucky actions. With $\kappa = 2$, ECA alone reduces the centered singleton evidence by at least a factor of 3, and VarGate further suppresses anchors whose observed differences are dominated by within-action noise.

D.4 Training-Stage Noise Estimated from Validation Success Rate

The previous analysis shows that the harmful event is driven by singleton lucky successes. In practice, the probability that a rare action obtains a successful downstream rollout increases as the agent becomes more capable, because later trajectories are more likely to recover from imperfect intermediate actions. We therefore use the GiGPO validation success rate at training step t , denoted by SR_t , as a proxy for the lucky-success noise mass:

$$p_t^- \propto \text{SR}_t. \quad (45)$$

This does not claim that every action has success probability SR_t ; rather, it estimates how likely a singleton rollout is to appear successful due to downstream recovery or stochastic continuation.

Table 6 reports five representative training steps. The singleton lucky-success proxy increases substantially during training, from 0.1250 at step 0 to 0.8828 at step 150. According to Eq. (41), ECA reduces this singleton-centered evidence by a factor of $1/(1 + \kappa) = 1/3$ before VarGate further gates unreliable anchors.

This explains why divergent anchor bias becomes more harmful in the later stage of training. As SR_t increases, a rare action is more likely to obtain a lucky successful rollout, and GiGPO is

Table 6: **Noise proxy estimated from GiGPO validation success rate.** We use SR_t as a proxy for the probability that a singleton rollout appears successful. With $\kappa = 2$, ECA reduces the centered singleton evidence by a factor of $1/3$ before applying VarGate.

Step	SR_t	ECA-scaled proxy
0	0.1250	0.0417
50	0.3281	0.1094
75	0.4844	0.1615
125	0.8516	0.2839
150	0.8828	0.2943

more likely to convert this singleton observation into a large positive advantage. ECPO counters this effect by applying stronger shrinkage to low-count actions and suppressing unreliable anchor states through VarGate. Consequently, even when late-stage training produces more singleton lucky successes, their expected contribution to the policy gradient is substantially reduced.

Algorithm 1 ECPO Training Procedure

```
1: Input: training task set  $\mathcal{D} = \{x\}$ , policy  $\pi_\theta$ , reference
   policy  $\pi_{\text{ref}}$ , environment  $\mathcal{E}$ , group size  $N$ , maximum iter-
   ations  $K$ , discount factor  $\gamma$ , step-credit weight  $\omega$ , shrink-
   age strength  $\kappa$ , VarGate temperature  $\tau$ , PPO clipping
   threshold  $\epsilon_{\text{clip}}$ , KL coefficient  $\beta$ .
2: Initialize: policy parameters  $\theta$ .
3:   *** ECPO training begins ***
4: for  $k = 1$  to  $K$  do
5:   Set old policy  $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$ .
6:   for each task description  $x \in \mathcal{D}$  do
7:     *** Step A: Group rollout collection. ***
8:     Sample  $N$  trajectories  $\{\tau_i\}_{i=1}^N$  from  $\pi_{\theta_{\text{old}}}$  in envi-
       ronment  $\mathcal{E}$ .
9:     Compute trajectory returns  $\{R_i\}_{i=1}^N$ .
10:    Compute trajectory-level advantages  $A_i^{\text{GRPO}}$  by
       Eq. (2).
11:    *** Step B: Anchor-state construction. ***
12:    for each trajectory  $\tau_i$  do
13:      for each step  $t = 1, \dots, T_i$  do
14:        Compute future return  $G_{i,t}$  by Eq. (3).
15:        Add occurrence  $(i, t)$  into the anchor set
            $\mathcal{I}_{s_{i,t}}$ .
16:      end for
17:    end for
18:    *** Step C: Evidence-Calibrated Action
       Advantage. ***
19:    for each anchor state  $s$  do
20:      Canonicalize each action  $a_{i,t}$  as  $u_{i,t} =$ 
         $\text{can}(a_{i,t})$ .
21:      Group occurrences in  $\mathcal{I}_s$  by canonical actions
         $\mathcal{U}_s$ .
22:      for each canonical action  $u \in \mathcal{U}_s$  do
23:        Compute action occurrence set  $\mathcal{I}_{s,u}$  and
        count  $n_{s,u}$ .
24:        Compute empirical action return  $\bar{G}_{s,u}$ .
25:        Compute shrinkage-calibrated return  $\tilde{\mu}_{s,u}$ 
        by Eq. (11).
26:      end for
27:      Compute calibrated action advantage
         $A_{\text{act}}(s_{i,t}, a_{i,t})$  by Eq. (13).
28:    end for
29:    *** Step D: Variance-Gated Credit Weighting. ***
30:    for each anchor state  $s$  do
31:      Compute between-action variance  $B_s$  by
        Eq. (14).
32:      Compute within-action variance  $W_s$  by
        Eq. (15).
33:      Compute reliability weight  $\rho_{\text{VG}}(s)$  by Eq. (18).
34:      if  $s$  has fewer than two canonical actions then
35:        Set  $\rho_{\text{VG}}(s) \leftarrow 0$ .
36:      end if
37:    end for
38:    *** Step E: Policy optimization. ***
39:    for each occurrence  $(i, t)$  do
40:      Compute final advantage  $\hat{A}_{i,t}^{\text{ECPO}}$  by Eq. (19).
41:      Assign  $\hat{A}_{i,t}^{\text{ECPO}}$  to all tokens of action  $a_{i,t}$ .
42:    end for
43:    Optimize  $\pi_\theta$  with the clipped objective in Eq. (21).
44:  end for
45: end for
46: Output: trained policy  $\pi_\theta$ .
```
