ADASTEP: ADAPTIVE STEP CREDIT WEIGHTING FOR AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon LLM agents are typically trained with sparse outcome rewards, making trajectory-level objectives too coarse to distinguish the contribution of individual decisions. Step-level credit assignment provides finer-grained supervision, but its estimates can be unreliable because observed returns also depend on subsequent actions, environment transitions, and trajectory length. We propose **AdaStep**, an **Ada**ptive **Step**-credit weighting method that controls how strongly each group-derived local advantage modifies the trajectory-level signal. We formulate this weighting as a mean-squared-error estimation problem for the latent step advantage and, under an explicit conditional sampling assumption, derive an optimal per-state shrinkage coefficient. The coefficient admits a **signal-to-total-variance** interpretation: it preserves local credit when return variation is attributable to the selected action and suppresses it when variation is dominated by downstream randomness. AdaStep requires only lightweight scalar computation, with no critic, additional rollouts, or extra model inference. Experiments with three model backbones on ALFWorld, WebShop, and ScienceWorld show **consistent improvements** over baselines at low computational cost.

1 INTRODUCTION

Large language models (LLMs) are increasingly moving from chat-based interaction to *agentic* decision making (Yao et al., 2023; Zhou et al., 2024; Wang et al., 2026; Zheng et al., 2026). Unlike a chat model, which primarily maps a prompt to a response, an agent repeatedly interacts with an external environment by observing states, taking actions, and receiving subsequent states and rewards. Completing a task therefore requires coordinating a sequence of decisions whose consequences may unfold over a long horizon. Meanwhile, group-based reinforcement learning has become an important paradigm for LLM post-training (Yu et al., 2025; Min et al., 2026; Zhang et al., 2026; Liu et al., 2026), offering a simple, critic-free learning signal. Given multiple rollouts of the same task, the mean return serves as an empirical estimate of the task-level value, and each rollout's return relative to this mean provides an advantage estimate, without requiring a separately trained critic (Shao et al., 2024; Yu et al., 2025). Under binary success rewards, successful trajectories receive nonnegative advantages and failed trajectories receive nonpositive advantages. Sparse, delayed rewards make it difficult to identify which intermediate actions contributed to an eventual outcome (Arjona-Medina et al., 2019). For agentic tasks, however, the long-horizon nature of decision making exposes a fundamental lim-

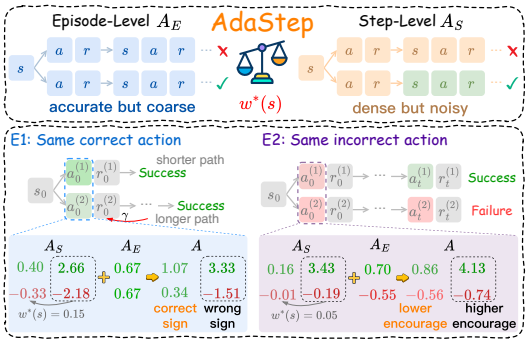

Figure 1: The episode-level advantage provides an accurate but coarse signal of trajectory outcomes, while the step-level advantage offers finer-grained but noisy local credit. In GiGPO, combining these signals with equal weights can assign negative advantages to useful actions or excessively large positive advantages to actions unrelated to success. AdaStep retains the episode-level signal and adaptively weights the local correction for each step-level group.

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.

itation of trajectory-level credit assignment in Group Relative Policy Optimization (GRPO) (Shao et al., 2024). By broadcasting the same outcome-based advantage to every state–action pair along a trajectory, GRPO provides a global but coarse learning signal: it does not distinguish actions that contributed to success from those that were irrelevant or erroneous, nor does it identify which decisions were responsible for failure. Consequently, useful actions in failed trajectories may be penalized, while unhelpful actions in successful trajectories may be reinforced.

To address this coarse credit assignment, Group in GRPO (GiGPO) (Feng et al., 2025) augments the trajectory-level advantage with a local, step-level correction. It groups state–action pairs sharing the same anchor state across rollouts and uses the group's mean discounted return as an empirical state-value baseline. Each sampled return is then compared with this baseline to obtain a step-level relative advantage. This construction provides finer-grained credit assignment while preserving critic-free estimation. However, these local advantages are estimated from finite samples, and their credibility can vary across step-level groups (Li et al., 2026). Step-level credit provides finer supervision, but its estimates are affected by downstream randomness. Even for the same state–action pair, stochastic transitions, subsequent actions, and discounted reward timing can produce advantage estimates with opposite signs. Strong local corrections may therefore amplify estimation noise, whereas weak corrections may suppress useful action information. This trade-off connects to variance control in advantage estimation (Schulman et al., 2015) and shrinkage-based reward-baseline estimation with limited rollouts (Zeng et al., 2026).

GiGPO (Feng et al., 2025) adds a local step-level correction to the global trajectory-level advantage using a fixed coefficient. However, this correction can differ in credibility across anchor-state groups. In some groups, return differences mainly reflect differences in action quality, making the local estimates useful for guiding policy updates. In others, much of the return variation arises from subsequent decisions or stochastic transitions, even when the same action is taken. Such variation makes the local estimates less credible. A fixed coefficient cannot adapt to these differences: a large weight may amplify noisy corrections, while a small weight may weaken useful local signals. We propose **AdaStep**, which retains the global advantage as a reference and adapts the local step-level correction strength to each group's estimation credibility, as illustrated in Figure 1. We derive the coefficient under a mean-squared-error objective and estimate it from existing rollout statistics, requiring no critic, additional rollouts, or extra model forward passes.

Our main contributions are as follows:

1. **An estimation formulation of local credit assignment.** Taking the trajectory-level advantage as a reference, we formulate local advantage weighting as an estimation problem for the true step-level advantage. This formulation makes explicit the trade-off between preserving useful local signals and suppressing sampling error.
2. **A principled correction coefficient for each group.** Under the proposed objective and stated assumptions, we derive the optimal correction coefficient and obtain a computationally tractable expression that adapts the strength of local correction to the estimation credibility of each step-level group.
3. **Empirical gains across three benchmarks.** Experiments on three agent-task benchmarks demonstrate that the proposed correction improves performance, supporting the value of accounting for local estimation credibility in group-relative credit assignment. AdaStep tops all trained baselines, gaining up to +9.36 points over GiGPO.
4. **Minimal computational overhead.** The method requires only a lightweight coefficient computation for each step-level group and integrates directly into the existing advantage calculation with minimal additional time and memory costs, adding about 1% credit-computation time.

2 Related Work

Reinforcement learning for large language models. Beyond preference alignment through RLHF, reinforcement learning is increasingly used to enhance LLM reasoning and decision making. Group-based methods, including RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), Dr. GRPO (Liu et al., 2025), DAPO (Yu et al., 2025), and CPPO (Lin et al., 2025), provide scalable alternatives to PPO (Schulman et al., 2017). These methods estimate relative advantages from multiple responses

to the same prompt, eliminating the need for a separately trained value function and reducing training costs. Recent work further advances this framework through hybrid token–sequence optimization in DHPO (Min et al., 2026), adaptive rollout selection in AERO (Zhang et al., 2026), and length-unbiased sequence-level optimization in LUSPO (Liu et al., 2026), emphasizing efficiency, robustness, and bias reduction in LLM post-training.

Step-level advantage assignment. GiGPO estimates step-level advantages by grouping state–action pairs with shared anchor states across rollouts. HGPO (He et al., 2026) and ProGPO (Fan & Liu, 2026) extend this idea through additional grouping mechanisms, introducing further computational costs. Critic-based approaches such as CAPO (Xie et al., 2025) and SAPO (Liang et al., 2026) incur the time and memory costs of training a critic, while EnvRL (Wang et al., 2026) adds auxiliary supervision through state prediction and inverse dynamics. PGPO (Zheng et al., 2026) refines step-level advantage signals but does not explicitly address how to control their correction strength. ECPO (Li et al., 2026) constructs heuristically motivated calibrated action-level advantages, whereas AdaStep retains the occurrence-level estimator and derives its state-specific scaling from an explicit estimation objective. Our analysis accounts for the action-conditional bias induced by the shared group baseline, while exploiting the cancellation of this bias after marginalizing over actions.

3 PRELIMINARIES

3.1 PROBLEM SETUP

We consider a general setting in which an LLM agent interacts with an environment to complete multi-step tasks described by a task prompt $x \sim p(X)$. We model the environment as a Markov decision process (MDP) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$, where $\mathcal{S}$ is the state space of the environment, $\mathcal{A}$ is the action space, $P(s_{t+1} \mid s_t, a_t)$ is the state transition model, r is the reward function, and γ is the discount factor. At each discrete time step $t = 1, 2, \ldots, T$, the environment is in a state $s_t \in \mathcal{S}$ and the agent generates a textual action $a_t \in \mathcal{A}_t$, where in this text world the action space is $\mathcal{A}_t = \mathcal{V}^n$, with $\mathcal{V}$ the token vocabulary and n the maximum generation length. In text worlds, $\mathcal{S}$ is typically a finite discrete set whose elements consist of text. After executing a_t, the environment returns a scalar reward $r_t \in \mathbb{R}$ and transitions to the next state s_{t+1} according to P. A full episode consists of a trajectory $\tau = \big\{(s_1, a_1, r_1), (s_2, a_2, r_2), \ldots, (s_T, a_T, r_T)\big\}$ with horizon T. The agent's behavior is governed by an LLM policy $\pi_\theta(a_t \mid s_t, x)$ parameterized by θ, which defines a distribution over textual outputs conditioned on the task prompt x and the current state s_t.

In many realistic scenarios the environment provides sparse or delayed rewards (e.g., success and failure signals at the end of an episode). Because the agent generates T consecutive textual actions $(a_1, \ldots, a_T)$, each potentially spanning thousands of tokens, assigning credit to individual tokens over the course of an episode is particularly challenging. In this paper we focus on tasks whose quality is measured by a binary *success* signal: the agent receives no reward before the final step ($r_t = 0$ for all $t < T$), and $r_T = 1$ only upon success in the final state, $r_T = 0$ otherwise. Therefore, for a trajectory τ_i, the episode return is given by $R_E(\tau_i) = \sum_{t=1}^{T} r_{i,t}$, taking the value 1 if and only if the trajectory succeeds, and 0 otherwise.

3.2 GRPO AND GIGPO

GRPO (Shao et al., 2024) removes the value network that PPO relies on and instead computes advantages by contrasting returns within a group of rollouts that share the same prompt. For a prompt x it samples n trajectories $\{\tau_1, \tau_2, \cdots, \tau_n\}$ and sets

$$A_E(\tau_i) = \frac{R_E(\tau_i) - \text{mean}\left(\{R(\tau_j)\}_{j=1}^{N}\right)}{\text{std}(\{R(\tau_j)\}_{j=1}^{N})}. \tag{1}$$

In GRPO, the episode return $R_E(\tau_i)$ is the ground-truth signal, and the group-relative advantage A_E computed for the trajectory τ_i is a single scalar applied to the entire trajectory τ_i; hence, for every state-action pair $(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)}) \in \tau_i$, we have $A_E(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = A_E(\tau_i)$. However, this advantage with respect to the state-action pair only captures the global information of the episode, namely, whether it succeeded and is therefore relatively coarse.

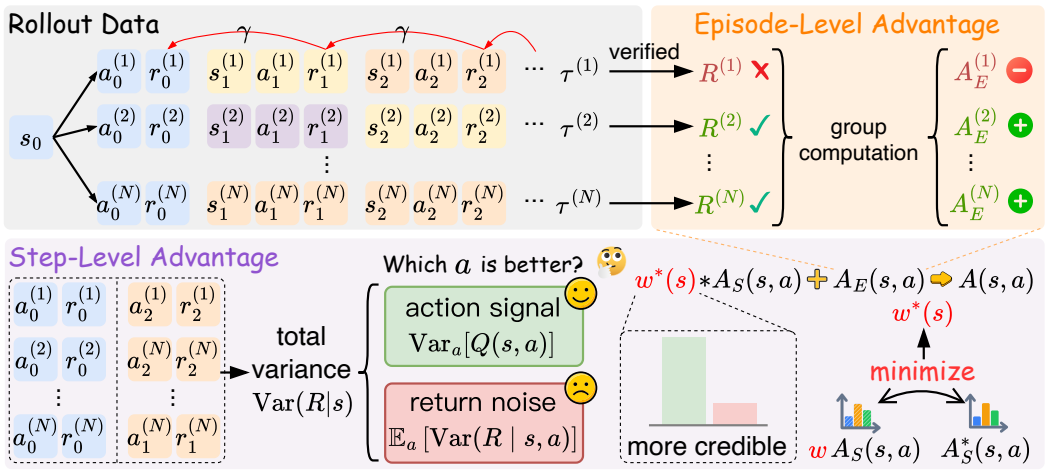

Figure 2: Overview of AdaStep. Rollout trajectories provide both episode-level returns and step-level groups. The episode-level relative advantage A_E supplies a global success/failure signal, while the step-level relative advantage A_S provides local credit but is affected by return noise. AdaStep estimates the reliability of A_S at each anchor state $\boldsymbol{s}$ using the fraction of return variance explained by action-dependent value differences, yielding the per-state weight $w^*(\boldsymbol{s})$. The final advantage combines the two signals as $A(\boldsymbol{s}, \boldsymbol{a}) = A_E(\boldsymbol{s}, \boldsymbol{a}) + w^*(\boldsymbol{s})A_S(\boldsymbol{s}, \boldsymbol{a})$.

GiGPO aims to address the problem that GRPO's signal is overly coarse, seeking fine-grained advantages that capture the contribution of individual actions to success. For the state $s_{i,t}$ in trajectory τ_i, its state return is the discounted return obtained from that state onward: $R_S(s_{i,t}) = \sum_{k=t}^{T} \gamma^{k-t} r_{i,k}$ Under the sparse setting, $r_{i,k} = 0$ for all $k < T$, while $r_{i,T} \in \{0, 1\}$, so $R_S(s_{i,t}) = \gamma^{T-t} r_{i,T}$.

Specifically, for a state set $\mathcal{S} = \{\tilde{s}_1, \cdots, \tilde{s}_K\}$ of size K, on top of GRPO's group rollouts within the same task, GiGPO matches states via a hash map and aggregates identical states together with their subsequent actions and returns. Within a cluster, the same state is referred to as an *anchor state*. The actions and returns immediately following the states in a cluster form a step-level group; that is,

$$\mathcal{G}(\tilde{s}_k) = \{(\boldsymbol{a}_t^{(i)}, R_t^{(i)}) : (\boldsymbol{s}_t, \boldsymbol{a}_t^{(i)}) \in \tau_i, i \in [1:N], t \in [1:T]\}$$

Based on this, the step-level group advantage of an action is computed as

$$A_S(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = \frac{R_t^{(i)} - \text{mean}(\{R_t^{(j)} | (\boldsymbol{a}_t^{(j)}, R_t^{(j)}) \in \mathcal{G}(\tilde{s}_k)\})}{\text{std}(\{R_t^{(j)} | (\boldsymbol{a}_t^{(j)}, R_t^{(j)}) \in \mathcal{G}(\tilde{s}_k)\})} \tag{2}$$

Therefore, in GiGPO, the advantage of an action is

$$A_G(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)}) = A_E(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)}) + \omega A_S(\boldsymbol{s}_t^{(i)}, \boldsymbol{a}_t^{(i)})$$

where $\omega \geqslant 0$ is the coefficient that controls the overall strength of the step-level relative advantage. In summary, the episode-level relative advantage provides a global reference, while the step-level relative advantage supplies an additional local correction.

4 Methods

4.1 Motivation

In policy-gradient methods, for a state-action pair with positive advantage, the gradient update increases its probability under the new policy relative to the old policy, and decreases it for pairs with negative advantage. The problem is that GRPO's global trajectory-level signal is too coarse: it reflects only whether the entire trajectory succeeds and carries no information about which decisions caused the task to succeed or fail. GiGPO's results show that adding a local advantage correction

to the global advantage does improve performance. However, step-level advantages are based on the state-action return of the current trajectory, which is affected by randomness in the subsequent trajectory and therefore cannot always faithfully reflect the true advantage of a state-action pair.

Example 1. *A task requires purchasing meat snacks with the* `three meat` *flavor and the* `3 pack` *option. At the same product-detail-page state, two trajectories both execute* `click[three meat]`*, correctly selecting the requested flavor. The shorter trajectory then selects the pack size and completes the purchase in 8 steps. The longer trajectory also selects the correct pack size but returns to the search results and browses additional pages before succeeding in 23 steps. Due to return discounting and within-group comparisons, GiGPO assigns advantages of* 3.33 *and* -1.51 *to this same correct action in the shorter and longer trajectories, respectively. This negative advantage is misleading: the action is correct in both cases, and the difference arises from stochastic subsequent behavior rather than from the action itself. Our method assigns advantages of* 1.07 *and* 0.35*, respectively, correcting the sign in the longer trajectory and reducing the discrepancy between the advantages assigned to the same state–action pair.*

Example 2. *A task requires placing a cooled pot on a stove. At the state described by "You arrive at sinkbasin 1. On the sinkbasin 1, you see a apple 1, a egg 1, and a lettuce 1", multiple trajectories execute* `clean pot 1 with sinkbasin 1`*. However, cleaning the pot does not cool it. Within one group, this action occurs four times. One trajectory succeeds through subsequent correct actions and receives a positive return, whereas the other three have zero return. GiGPO assigns the cleaning action an advantage of* 4.13 *in the successful trajectory and* -0.74 *in each of the other three. This positive credit is misleading because the cleaning action itself does not satisfy the cooling requirement; success results from later actions. Our method changes these advantages to* 0.86 *and* -0.56*, respectively.*

The local step-level signal is therefore valuable but not fully credible, and how strongly to trust it (i.e., how to control the strength of the step-level correction) is an important question. Throughout this section we write $\boldsymbol{s}$ for the *anchor state* of GiGPO, the state on which a step-level group $\mathcal{G}(\boldsymbol{s})$ is formed (Section 3.2). Figure 2 gives an overview of AdaStep, and the complete training procedure is summarized in Algorithm 1 (Appendix B).

4.2 Optimal per-state shrinkage

We assume that, before the advantage is normalized by its standard deviation, for each state–action pair in a task, there exists a ground-truth step-level advantage $A^*_S(\boldsymbol{s}, \boldsymbol{a})$ that we would like to recover but cannot obtain exactly. In practice, within the step-level group $\mathcal{G}(\boldsymbol{s})$ of an anchor state $\boldsymbol{s}$, we estimate it using the unnormalized form of Eq. (2), obtaining a noisy estimate

$$A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) + \eta_i, \tag{3}$$

where η_i is the estimation noise associated with the trajectory τ_i in the group. Because the step-level advantage is computed *within* a group, we seek a per-group weight that rescales the group's estimate so as to best approximate the true advantage. For each anchor state $\boldsymbol{s}$ we therefore solve

$$w^*(\boldsymbol{s}) = \arg\min_{w \in \mathbb{R}} \mathbb{E}_{\boldsymbol{a}^{(i)} \sim \pi(\cdot|\boldsymbol{s})} \Big[\big(w\, A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) \big)^2 \;\Big|\; \boldsymbol{s} \Big]. \tag{4}$$

This objective asks: among all scalars w applied to the *estimate* A_S, which one lands closest, in mean square over the action distribution at $\boldsymbol{s}$, to the *truth* A^*_S? The solution is a standard projection, requiring only that the second moment of the estimate be strictly positive.

Theorem 1 (Optimal per-state weight)**.** *Fix an anchor state $\boldsymbol{s}$ and a step-level group $\mathcal{G}(\boldsymbol{s})$ of size $N_{\boldsymbol{s}} \geqslant 2$ with the noise model in Eq.* (3)*, with a strictly positive second moment of the estimate, $\mathbb{E}[A_S(\boldsymbol{s}, \boldsymbol{a})^2 \mid \boldsymbol{s}] > 0$. Then the minimizer of Eq.* (4) *is*

$$w^*(\boldsymbol{s}) = \frac{\mathbb{E}[A_S(\boldsymbol{s}, \boldsymbol{a})\, A^*_S(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}]}{\mathbb{E}[A_S(\boldsymbol{s}, \boldsymbol{a})^2 \mid \boldsymbol{s}]}. \tag{5}$$

The numerator measures how well the estimate A_S aligns with the truth A^*_S; the denominator is the second moment of the estimate and is the sole condition needed for the solution to be well defined. The proof is deferred to Appendix A. Theorem 1 is a population-level projection: it is exact but, as written, *not yet computable*, since it requires the unknown A^*_S. We next replace it by an equivalent form that depends only on group statistics GiGPO already collects.

4.3 COMPUTABLE FORM FROM GROUP STATISTICS

To make the optimal weight in Eq. (5) tractable, we use the definition of step-level group relative advantage in Eq. (2). Within a group $\mathcal{G}(\boldsymbol{s})$ of size N_s, the step-level advantage without standard-deviation normalization is:

$$A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = R^{(i)} - \hat{V}(\boldsymbol{s}), \qquad \hat{V}_{N_s}(\boldsymbol{s}) = \frac{1}{N_s} \sum_{j \in \mathcal{G}(\boldsymbol{s})} R^{(j)}, \tag{6}$$

while, under a fixed policy π, the ground truth it estimates is

$$A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - V(\boldsymbol{s}), \qquad V(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}[Q(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}] = \mathbb{E}_\pi[R \mid \boldsymbol{s}], \tag{7}$$

Therefore, in the computation, GiGPO uses the mean of the returns within a group as the estimate of the state-value function, and only a single return as the estimate of the state-action value, where $Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_\pi[R \mid \boldsymbol{s}, \boldsymbol{a}]$ is the action-value function. Therefore, the estimation noise of the advantage arises from two sources: the estimation of Q and the estimation of V. Furthermore, by computing the estimation error $\eta_i = A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})$, we obtain

$$\eta_i = \frac{N_s - 1}{N_s}\big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) \; - \; \frac{A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})}{N_s} \; - \; \frac{1}{N_s} \sum_{j \neq i} \big(R^{(j)} - V(\boldsymbol{s})\big). \tag{8}$$

The left term is the sampling noise of the current rollout—the realized return minus its conditional mean; the middle term is a $1/N_s$ shrinkage of the true step advantage, an artifact of estimating the baseline with a finite group that includes sample i itself; the right term is the average deviation of the *other* group members from the state value. Taking conditional expectations, the left and right terms vanish and only the middle term survives: $\mathbb{E}_i[\eta_i \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] = -A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})/N_s$. Conditioned on both $\boldsymbol{s}$ and the action, the estimate carries a $1/N_s$ bias; and the marginalize over the action this bias disappears, since $\mathbb{E}_{\boldsymbol{a}}[A^*_S \mid \boldsymbol{s}] = 0$ by definition of $V(\boldsymbol{s})$,$\mathbb{E}_{i,\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}[\eta_i \mid \boldsymbol{s}] = 0$. Thus within a step group the step-level estimate is *unbiased* for the true step advantage—reweighting, rather than sign-flipping or discarding, is the right intervention.

Furthermore, we assume that within a group $\mathcal{G}(\boldsymbol{s})$, the action–return pairs $(\boldsymbol{a}^{(j)}, R^{(j)})$ are independent and identically distributed conditional on $\boldsymbol{s}$, and that the return variance within a group is finite. Under these two practical assumptions, the numerator and denominator of the optimal weight in Eq. (5) admit the following closed forms:

Lemma 1. *Assume the action–return pairs $(\boldsymbol{a}^{(j)}, R^{(j)})$ are i.i.d. conditional on $\boldsymbol{s}$ with finite return variance, Then the numerator and denominator of* (5) *admit the closed forms*

$$\mathbb{E}[A_S \, A^*_S \mid \boldsymbol{s}] = \frac{N_s - 1}{N_s} \, \mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big], \mathbb{E}[A^2_S \mid \boldsymbol{s}] = \frac{N_s - 1}{N_s} \, \mathrm{Var}(R \mid \boldsymbol{s}).$$

The common factor $(N_s - 1)/N_s$ cancels in the ratio. A restatement and proof are given in Lemma A.2 of Appendix A.

We first express the optimal coefficient as a signal-to-(signal-plus-noise) ratio to make its credibility interpretation explicit. By the law of total variance, at a fixed state $\boldsymbol{s}$, the return variance decomposes into the variance of $Q(\boldsymbol{s}, \boldsymbol{a})$ across actions $\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})$ and the conditional return variance after fixing both $\boldsymbol{s}$ and $\boldsymbol{a}$, averaged over the same action distribution:

$$\mathrm{Var}(R \mid \boldsymbol{s}) = \mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big] + \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big]. \tag{9}$$

The decomposition is proved in Lemma A.3 of Appendix A.

Theorem 2. *Under the assumptions of Theorem 1 and Lemma 1, the optimal per-state weight* (5) *admits two equivalent forms. The first expresses the weight as the fraction of return variance due to differences in expected return across actions:*

$$w^*(\boldsymbol{s}) = \frac{\mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big]}{\mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big] + \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big]} \; \in [0, 1]. \tag{10}$$

By the law of total variance, the equivalent computable form in terms of return variances is

$$w^*(\boldsymbol{s}) = 1 - \frac{\mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot|\boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big]}{\mathrm{Var}(R \mid \boldsymbol{s})}. \tag{11}$$

Table 1: **Main results.** Performance across three benchmarks. Results are averaged over 3 random seeds. The best and second-best values within each model block are shown in **bold** and underlined, respectively. Δ vs. GiGPO reports AdaStep minus GiGPO for the same model.

Type	Method	ALFWorld		WebShop		ScienceWorld
		In-Success	Out-Success	Test Score	Success	Success
Frontier Models						
Prompting	DeepSeek-V4.1-Flash (ReAct)	80.60±0.2	88.83±1.9	28.54±1.8	25.13±1.5	58.15±4.3
Prompting	GLM-5.2 (ReAct)	96.36±0.9	95.05±1.2	28.62±2.1	23.47±1.8	82.03±1.4
Model: Qwen3-1.7B						
Prompting	Pure model	9.64±1.2	11.46±2.4	53.38±3.1	5.34±0.8	9.38±2.1
Prompting	ReAct (Yao et al., 2023)	19.79±1.6	20.05±1.8	33.09±0.8	8.72±1.6	9.90±1.6
Prompting	Reflexion (Shinn et al., 2023)	52.02±1.9	51.30±3.1	37.26±1.8	22.66±3.6	18.49±0.7
RL Training	PPO (Schulman et al., 2017)	45.57±1.6	42.44±3.2	76.77±3.7	61.10±4.4	21.61±5.5
RL Training	RLOO (Ahmadian et al., 2024)	66.41±8.2	53.90±2.1	63.23±0.4	37.60±3.6	18.23±1.2
RL Training	GRPO (Shao et al., 2024)	71.09±0.8	64.58±4.3	71.00±0.6	38.83±3.1	19.79±1.2
RL Training	GiGPO (Feng et al., 2025)	71.10±3.4	64.06±0.0	80.61±0.8	64.58±0.8	20.57±1.8
RL Training	HGPO (He et al., 2026)	71.35±3.9	63.80±1.8	75.56±2.6	43.35±2.7	25.78±1.4
RL Training	**AdaStep (Ours)**	**73.70**±3.5	**65.79**±3.0	**81.36**±2.5	**65.24**±2.2	**29.93**±1.9
Δ *vs. GiGPO*		+2.60	+1.73	+0.75	+0.66	+9.36
Model: Qwen3-4B						
Prompting	Pure model	23.96±3.1	17.19±2.0	25.70±1.7	1.95±0.3	14.59±2.5
Prompting	ReAct (Yao et al., 2023)	77.61±1.6	80.99±2.5	29.55±0.6	16.93±0.9	15.10±1.9
Prompting	Reflexion (Shinn et al., 2023)	79.95±4.0	74.42±1.5	35.87±1.3	29.04±1.4	37.76±1.1
RL Training	PPO (Schulman et al., 2017)	38.54±1.6	40.63±3.6	74.23±4.4	61.37±2.1	30.97±0.4
RL Training	RLOO (Ahmadian et al., 2024)	76.04±0.9	76.82±0.5	79.02±1.9	67.58±2.4	37.76±1.7
RL Training	GRPO (Shao et al., 2024)	77.61±3.7	73.96±2.4	83.60±1.6	70.43±0.2	42.97±2.7
RL Training	GiGPO (Feng et al., 2025)	88.02±0.9	83.07±4.6	87.34±0.6	78.28±1.8	46.67±2.3
RL Training	HGPO (He et al., 2026)	91.93±1.6	85.42±0.5	88.18±0.4	76.43±1.6	47.63±1.3
RL Training	**AdaStep (Ours)**	**93.01**±0.9	**86.58**±1.2	**88.76**±0.5	**80.27**±0.4	**48.70**±2.7
Δ *vs. GiGPO*		+4.99	+3.51	+1.42	+1.99	+2.03
Model: Qwen2.5-7B-Instruct						
Prompting	Pure model	14.80	-	26.40	7.80	13.00
Prompting	ReAct (Yao et al., 2023)	31.20	-	46.20	19.50	12.50
Prompting	Reflexion (Shinn et al., 2023)	42.70	-	58.10	28.80	23.70
RL Training	PPO (Schulman et al., 2017)	43.23±1.8	42.19±4.8	81.40±3.1	68.7±5.1	24.90±3.6
RL Training	RLOO (Ahmadian et al., 2024)	73.70±1.2	67.19±0.8	80.3±3.2	65.7±4.0	37.70±1.0
RL Training	GRPO (Shao et al., 2024)	80.47±1.6	80.99±0.9	79.3±2.8	66.1±3.7	42.73±2.5
RL Training	GiGPO (Feng et al., 2025)	92.71±0.9	89.06±1.4	88.93±1.5	77.60±1.7	46.63±1.2
RL Training	HGPO (He et al., 2026)	94.79±0.5	93.23±1.2	88.96±1.0	78.51±1.4	46.87±1.5
RL Training	**AdaStep (Ours)**	**97.66**±0.8	**93.30**±1.3	**90.46**±0.4	**82.42**±1.2	**47.17**±0.5
Δ *vs. GiGPO*		+4.95	+4.24	+1.53	+4.82	+0.54

See Theorem A.2 in Appendix A for the proof. Equation (10) expresses the weight as the fraction of return variance explained by differences in expected return across actions. We regard these differences as signal, and the remaining variability for fixed state–action pairs, averaged over actions, as noise. The weight approaches 1 when signal dominates and 0 when noise dominates. This group-specific weight replaces the global coefficient ω in GiGPO.

Equation (11) allows us to estimate the weight from rollout return statistics without explicitly estimating action values. Section 5.3 specifies the sample requirements and fallback weights for groups with insufficient data.

We compute $w^*(\boldsymbol{s})$ from the unnormalized returns and multiply the local advantage by this weight. If standard-deviation normalization is used, we first normalize the local advantage within $\mathcal{G}(\boldsymbol{s})$ and then apply the same weight.

5 EXPERIMENTS

We evaluate AdaStep using three backbone models, Qwen3-1.7B, Qwen3-4B, and Qwen2.5-7B-Instruct, on three benchmarks: ALFWorld, WebShop, and ScienceWorld. We compare against closed-source model baselines, prompting methods such as ReAct, and reinforcement learning algorithms including GRPO, GiGPO, and HGPO. Detailed benchmark descriptions, baselines, and implementation settings are provided in Appendix C.

5.1 MAIN RESULTS

Table 1 compares AdaStep with prompting-based and RL baselines across three models and three benchmarks. AdaStep consistently outperforms HGPO and GiGPO across all model–benchmark combinations, with more pronounced gains over GRPO and PPO. These results support the effectiveness of reliability-aware step-level credit assignment across different model scales and task environments.

5.2 ROBUSTNESS TO ADVANTAGE NORMALIZATION

To examine whether AdaStep's gains depend on a particular normalization setting, Table 2 compares GiGPO and AdaStep both with and without standard-deviation normalization on Qwen3-4B and Qwen2.5-7B-Instruct. We add reruns of ALFWorld without std normalization and WebShop with std normalization. In both settings, the coefficient is computed from unnormalized group statistics before any normalization of the step-level advantages.

AdaStep consistently achieves higher mean performance than GiGPO both with and without standard-deviation normalization (Table 2), supporting its effectiveness under both settings.

Table 2: Results with and without std normalization. Δ denotes gains over GiGPO and best results **bolded**.

Std	Method	ALFWorld		WebShop	
		In-Suc.	Out-Suc.	Score	Suc.
Model: Qwen3-4B					
w/	GiGPO	88.02±0.9	83.07±4.6	88.77±0.3	77.17±0.3
w/	**AdaStep**	**93.01**±0.9	**86.58**±1.2	**89.90**±1.1	**79.27**±0.5
Δ		+4.99	+3.51	+1.13	+2.10
w/o	GiGPO	88.20±1.2	85.17±0.7	87.34±0.6	78.28±1.8
w/o	**AdaStep**	**91.73**±1.8	**85.40**±0.8	**88.76**±0.5	**80.27**±0.4
Δ		+3.53	+0.23	+1.42	+1.99
Model: Qwen2.5-7B-Instruct					
w/	GiGPO	92.71±0.9	89.06±1.4	88.70±2.9	76.53±2.7
w/	**AdaStep**	**97.66**±0.8	**93.30**±1.3	**89.93**±0.1	**79.97**±0.6
Δ		+4.95	+4.24	+1.23	+3.44
w/o	GiGPO	92.93±0.3	88.00±1.6	88.93±1.5	77.60±1.7
w/o	**AdaStep**	**96.00**±0.1	**90.63**±1.0	**90.46**±0.4	**82.42**±1.2
Δ		+3.07	+2.63	+1.53	+4.82

5.3 FALLBACK WEIGHTS FOR NON-ESTIMABLE GROUPS

In practice, some step-level groups contain only one distinct action, only one return observation per action, or identical returns, as illustrated in Table 3(a). We therefore estimate the coefficient in Theorem 2 only for groups containing at least two distinct actions, at least two return observations per action, and positive total return variance.

Groups with identical returns have zero step-level advantages and require no correction. For other groups that do not meet the estimation criteria, we compare three default weights: $w = 0$, $w = \bar{w}$, and $w = 1$, where $\bar{w}$ is the mean coefficient over eligible groups in the current batch when such groups exist. Eligible groups use their estimated coefficients in all three variants.

On WebShop and ALFWorld with Qwen2.5-7B-Instruct, $w = 1$ achieves the highest observed success rate among the three choices (Table 3(b)), supporting retention of the original step-level signal when evidence is insufficient for coefficient estimation.

5.4 SENSITIVITY TO THE STEP-LEVEL COEFFICIENT

Figure 3(a,b) compares fixed global weights in $A = A_E + wA_S$ with AdaStep using Qwen2.5-7B-Instruct on ALFWorld and WebShop. On both benchmarks, moderate local weighting improves performance, whereas excessive weighting degrades it. AdaStep achieves higher success rates than every tested fixed coefficient, supporting group-specific adaptation beyond global weight tuning.

Table 3: Left: groups outside the estimation criteria; right: Qwen2.5-7B-Instruct success rates (%) under different default weights, with the best results in **bold**.

(a) Group configurations

Observed group	Handling
One action; identical returns	$A_S = 0$; no correction.
One action; varying returns	Cannot compute $\mathbb{E}_{\boldsymbol{a}}[\cdot]$.
Multiple actions; one return observation per action	Cannot compute $\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})$.

(b) Fallback ablation

Fallback	WebShop	ALFWorld
$w = 0$	80.1	92.2
$w = \bar{w}$ (mean)	77.0	96.1
$w = 1$	**82.4**	**97.6**

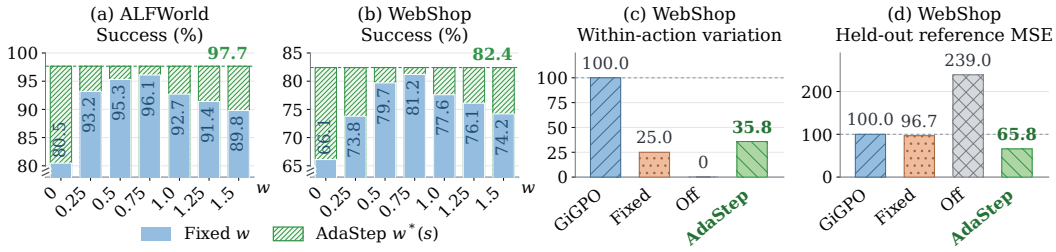

Figure 3: **Coefficient sensitivity and step-credit diagnostics.** (a,b) Success rates for fixed weights (blue) and AdaStep (hatched green). (c,d) WebShop within-action variation and held-out reference MSE, each normalized separately to GiGPO $= 100$; Fixed uses $c = 0.5$ and Off uses $c = 0$. Appendix E gives the diagnostic protocol.

5.5 Analysis of Step-Credit Estimation on WebShop

We examine step-credit estimation on 3,666 historical WebShop anchor-state groups using two complementary diagnostics. Within-action variation measures the remaining fluctuation among weighted credits assigned to the same action. Held-out reference MSE measures their squared deviation from action-level advantages estimated using a separate subset of samples. Appendix E describes the coefficient estimation, sample splitting, and metric definitions.

Figure 3(c,d) shows that AdaStep reduces within-action variation while achieving the lowest held-out reference MSE. This suggests a better balance between suppressing fluctuations and preserving action-level information, as reducing variation alone does not guarantee better estimation.

5.6 Computational Efficiency

Table 4 compares per-iteration return and advantage computation on WebShop with Qwen2.5-7B-Instruct. AdaStep achieves the highest mean success rate with approximately 1.0% more credit-computation time than GiGPO, while also requiring less time than HGPO. These results show that adaptive step-credit weighting improves performance with minimal additional computation.

Table 4: Credit-computation time per iteration and success rates on WebShop.

Method	Time (s)	Success (%)
GiGPO	$0.4968_{\pm 0.0324}$	$77.60_{\pm 1.7}$
HGPO	$0.6340_{\pm 0.0711}$	$78.51_{\pm 1.4}$
AdaStep	$0.5018_{\pm 0.0114}$	$\mathbf{82.42}_{\pm 1.2}$

6 Conclusion

We introduced AdaStep, which derives an optimal per-state shrinkage coefficient from a mean-squared-error objective on the latent step advantage and reads it as the fraction of return variance explained by the selected action. Across three backbones and three benchmarks, AdaStep consistently improves over GRPO, GiGPO, and HGPO, with gains that persist with and without advantage normalization, at negligible computational cost. Extending AdaStep to continuous or partially observable environments, where anchor states cannot be matched directly, remains open and is discussed in Appendix F.

REPRODUCIBILITY STATEMENT

We present AdaStep in Section 4, with theoretical proofs in Appendix A and pseudocode in Appendix B. Experimental settings, prompt templates, and diagnostic protocols are provided in Appendices C–E. We will publicly release all code and configurations upon acceptance.

AI USE STATEMENT

In this work, we used large language models to help edit the text, prepare plotting code, and check derivations. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

Our work studies credit assignment in reinforcement learning for LLM agents and does not involve human subjects, personally identifiable information, or the collection of new user data. All experiments are conducted in publicly available simulated environments (ALFWorld, WebShop, and ScienceWorld), and the language models used are publicly released. We foresee no ethical concerns or conflicts of interest, and we note that the proposed method optimizes training signals only, without introducing new data collection or deployment practices.

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.

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

DeepSeek-AI, Anyi Xu, B. Li, Bangcai Lin, Bing Xue, BingCheng Xian, Bingzheng Xu, Bochao Wu, et al. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026.

Mingxuan Fan and Peiyang Liu. Progress- and reliability-oriented group policy optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.04242*, 2026.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 46375–46408, 2025.

GLM-5-Team, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, et al. GLM-5: From vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

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

Yuanfan Li, Qi Zhou, Wenjing Duan, and Lu Chen. When denser credit is not enough: Evidence-calibrated policy optimization for long-horizon LLM agent training. *arXiv preprint arXiv:2606.05885*, 2026.

Dayang Liang, Lang Feng, Bo An, and Yunlong Liu. SAPO: Single-rollout autoregressive policy optimization for agentic reinforcement learning. *arXiv preprint arXiv:2608.19842*, 2026.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. CPPO: Accelerating the training of group relative policy optimization-based reasoning models. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 61043–61068, 2025.

Fanfan Liu, Youyang Yin, Peng Shi, Siqi Yang, Zhixiong Zeng, and Haibo Qiu. Length-unbiased sequence policy optimization: Revealing and controlling response length variation in RLVR. *arXiv preprint arXiv:2602.05261*, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025.

Zijun Min, Bingshuai Liu, Ante Wang, Long Zhang, Anxiang Zeng, Haibo Zhang, and Jinsong Su. Orchestrating tokens and sequences: Dynamic hybrid policy optimization for RLVR. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 37782–37795, 2026.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

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, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, 2022.

Zhitong Wang, Songze Li, Hao Peng, Shuzheng Si, Yi Wang, Maosong Sun, and Juanzi Li. EnvRL: Learn from environment dynamics in agentic reinforcement learning. *arXiv preprint arXiv:2606.17680*, 2026.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. CAPO: Towards enhancing LLM reasoning through generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, 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. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244, 2025.

Guanning Zeng, Zhaoyi Zhou, Daman Arora, and Andrea Zanette. Shrinking the variance: Shrinkage baselines for reinforcement learning with verifiable rewards. In *International Conference on Machine Learning*, 2026.

Zhi Zhang, Zhen Han, Costas Mavromatis, Qi Zhu, Yunyi Zhang, Sheng Guan, Dingmin Wang, Xiong Zhou, Shuai Wang, Soji Adeshina, Vassilis Ioannidis, and Huzefa Rangwala. Train less, learn more: Adaptive efficient rollout optimization for group-based reinforcement learning. *arXiv preprint arXiv:2602.14338*, 2026.

Yuyao Zheng, Haipeng Sun, Junwei Bao, Lemao Liu, Hongfei Jiang, Yang Song, and Dejing Dou. PGPO: Potential-guided policy optimization for multi-turn agentic tasks. *arXiv preprint arXiv:2609.02236*, 2026.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning*, volume 235, pp. 62178–62209, 2024.

A PROOFS OF MAIN RESULTS

We collect here the proofs of the results stated in the main text. Throughout, we fix an anchor state $\boldsymbol{s}$ and write $N_s = |\mathcal{G}(\boldsymbol{s})| \geqslant 2$ for the size of its step-level group. We assume the action–return samples are i.i.d. conditional on $\boldsymbol{s}$,

$$(\boldsymbol{a}^{(j)}, R^{(j)}) \overset{\text{i.i.d.}}{\sim} \pi(\boldsymbol{a} \mid \boldsymbol{s})\, p(R \mid \boldsymbol{s}, \boldsymbol{a}),$$

and that $\mathrm{Var}[R \mid \boldsymbol{s}] < \infty$. The state and action values are

$$V(\boldsymbol{s}) = \mathbb{E}_\pi[R \mid \boldsymbol{s}], \qquad Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_\pi[R \mid \boldsymbol{s}, \boldsymbol{a}], \qquad A^*_S(\boldsymbol{s}, \boldsymbol{a}) = Q(\boldsymbol{s}, \boldsymbol{a}) - V(\boldsymbol{s}),$$

and the return variance decomposes by the law of total variance as

$$\mathrm{Var}(R \mid \boldsymbol{s}) = \mathrm{Var}_{\boldsymbol{a}}[Q(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}] + \mathbb{E}_{\boldsymbol{a}}[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}].$$

Two identities follow immediately from $V(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a}}[Q(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}]$: $\mathbb{E}_{\boldsymbol{a}}[A^*_S(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}] = 0$ and $\mathbb{E}_{\boldsymbol{a}}[A^*_S(\boldsymbol{s}, \boldsymbol{a})^2 \mid \boldsymbol{s}] = \mathrm{Var}_{\boldsymbol{a}}[Q(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}]$. The step-level estimate and its target are

$$A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = R^{(i)} - \hat{V}_{N_s}, \qquad \hat{V}_{N_s} = \frac{1}{N_s} \sum_{j=1}^{N_s} R^{(j)}, \qquad A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - V(\boldsymbol{s}),$$

where the group mean $\hat{V}_{N_s}$ includes sample i itself.

Theorem A.1 (Theorem 1 in the main text)**.** *Fix an anchor state $\boldsymbol{s}$ and a step-level group $\mathcal{G}(\boldsymbol{s})$ of size $N_s \geqslant 2$ with the noise model in Eq.* (3)*, with a strictly positive second moment of the estimate, $\mathbb{E}[A_S(\boldsymbol{s}, \boldsymbol{a})^2 \mid \boldsymbol{s}] > 0$. Then the minimizer of Eq.* (4) *is*

$$w^*(\boldsymbol{s}) = \frac{\mathbb{E}[A_S(\boldsymbol{s}, \boldsymbol{a})\, A^*_S(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}]}{\mathbb{E}[A_S(\boldsymbol{s}, \boldsymbol{a})^2 \mid \boldsymbol{s}]}. \tag{12}$$

Proof. We minimize $\mathcal{L}(w) = \mathbb{E}\big[(w\, A_S - A^*_S)^2 \mid \boldsymbol{s}\big]$ over $w \in \mathbb{R}$, the expectation taken over i and $\boldsymbol{a}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s})$. Expanding the square,

$$\mathcal{L}(w) = w^2\, \mathbb{E}[A_S^2 \mid \boldsymbol{s}] - 2w\, \mathbb{E}[A_S\, A^*_S \mid \boldsymbol{s}] + \mathbb{E}[A^{*2}_S \mid \boldsymbol{s}].$$

This is a convex quadratic in w (its curvature $2\,\mathbb{E}[A_S^2 \mid \boldsymbol{s}]$ is strictly positive by assumption), so its minimizer is the stationary point. Differentiating and setting to zero,

$$\frac{\partial \mathcal{L}}{\partial w} = 2w\, \mathbb{E}[A_S^2 \mid \boldsymbol{s}] - 2\, \mathbb{E}[A_S\, A^*_S \mid \boldsymbol{s}] = 0 \quad \Longrightarrow \quad w^*(\boldsymbol{s}) = \frac{\mathbb{E}[A_S\, A^*_S \mid \boldsymbol{s}]}{\mathbb{E}[A_S^2 \mid \boldsymbol{s}]}.$$

This is the orthogonal projection of the truth A^*_S onto the one-dimensional span of the estimate A_S; it requires only $\mathbb{E}[A_S^2 \mid \boldsymbol{s}] > 0$, and no unbiasedness assumption is needed for this projection to hold. The role of unbiasedness, established next, is to make the closed form of the numerator computable. □

Lemma A.1. *Fix an anchor state $\boldsymbol{s}$ and a step-level group $\mathcal{G}(\boldsymbol{s})$ of size $N_s \geqslant 2$, with action–return samples $(\boldsymbol{a}^{(j)}, R^{(j)})$ i.i.d. conditional on $\boldsymbol{s}$, and define $\hat{V}_{N_s} = \frac{1}{N_s} \sum_{j=1}^{N_s} R^{(j)}$ (the group mean, which includes sample i itself), $A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = R^{(i)} - \hat{V}_{N_s}$, $A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - V(\boldsymbol{s})$, and $\eta_i = A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})$. Then η_i admits the three-term decomposition*

$$\eta_i = \frac{N_s - 1}{N_s}\big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) \;-\; \frac{A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})}{N_s} \;-\; \frac{1}{N_s} \sum_{j \neq i} \big(R^{(j)} - V(\boldsymbol{s})\big), \tag{13}$$

i.e. Eq. (8) *of the main text, where term left is the sampling noise of the current rollout, term middle is a $1/N_s$ shrinkage of the true step advantage, and term right is the average deviation of the other group members from the state value. Its conditional and marginal means are*

$$\mathbb{E}_i[\eta_i \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] = -\frac{A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})}{N_s}, \tag{14}$$

$$\mathbb{E}_{i, \boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}[\eta_i \mid \boldsymbol{s}] = 0. \tag{15}$$

Proof. From $A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = R^{(i)} - \hat{V}_{N_s}$ and $A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - V(\boldsymbol{s})$,

$$\eta_i = A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}) = \big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) - \big(\hat{V}_{N_s} - V(\boldsymbol{s})\big).$$

Split the group-mean residual into the i-th and the remaining terms,

$$\hat{V}_{N_s} - V(\boldsymbol{s}) = \frac{1}{N_s}\sum_{j=1}^{N_s}(R^{(j)} - V(\boldsymbol{s})) = \frac{1}{N_s}(R^{(i)} - V(\boldsymbol{s})) + \frac{1}{N_s}\sum_{j\neq i}(R^{(j)} - V(\boldsymbol{s})),$$

so that

$$\eta_i = \big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) - \frac{1}{N_s}(R^{(i)} - V(\boldsymbol{s})) - \frac{1}{N_s}\sum_{j\neq i}(R^{(j)} - V(\boldsymbol{s})).$$

Grouping the first two terms,

$$\begin{aligned} R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - \frac{1}{N_s}(R^{(i)} - V(\boldsymbol{s})) &= \frac{N_s - 1}{N_s}\big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) + \frac{1}{N_s}\big(V(\boldsymbol{s}) - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) \\ &= \frac{N_s - 1}{N_s}\big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) - \frac{1}{N_s}A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)}), \end{aligned}$$

which yields the three-term decomposition (8):

$$\eta_i = \frac{N_s - 1}{N_s}\big(R^{(i)} - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) - \frac{A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})}{N_s} - \frac{1}{N_s}\sum_{j\neq i}\big(R^{(j)} - V(\boldsymbol{s})\big).$$

We now take conditional expectations. Given $(\boldsymbol{s}, \boldsymbol{a}^{(i)})$:

- Term left has mean $\frac{N_s-1}{N_s}\big(\mathbb{E}[R^{(i)} \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) = \frac{N_s-1}{N_s}\big(Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})\big) = 0$.
- For term right, each $R^{(j)}$ with $j \neq i$ is independent of $\boldsymbol{a}^{(i)}$ and has conditional mean $\mathbb{E}[R^{(j)} \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] = \mathbb{E}[R^{(j)} \mid \boldsymbol{s}] = V(\boldsymbol{s})$, so $\mathbb{E}[R^{(j)} - V(\boldsymbol{s}) \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] = 0$, and term (iii) vanishes.

Only term middle survives, giving the conditional bias (14):

$$\mathbb{E}_i[\eta_i \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] = -\frac{A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})}{N_s}.$$

Marginalizing once more over $\boldsymbol{a}^{(i)} \sim \pi(\cdot \mid \boldsymbol{s})$ and using $\mathbb{E}_{\boldsymbol{a}}[A^*_S(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}] = 0$,

$$\mathbb{E}_{i,\boldsymbol{a}}[\eta_i \mid \boldsymbol{s}] = -\frac{1}{N_s}\,\mathbb{E}_{\boldsymbol{a}}[A^*_S(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}] = 0,$$

which is the state-level unbiasedness (15). In words: the estimate is biased by a $1/N_s$ factor at a fixed action, but unbiased once averaged over the action distribution at $\boldsymbol{s}$. □

Lemma A.2 (Lemma 1 in the main text)**.** *Assume the action–return pairs $(\boldsymbol{a}^{(j)}, R^{(j)})$ are i.i.d. conditional on $\boldsymbol{s}$ with finite return variance, in addition to the assumptions of Theorem 1. For the step-level estimate and its target defined in* (6) *and* (7)*, the numerator and denominator of* (5) *satisfy*

$$\mathbb{E}[A_S\, A^*_S \mid \boldsymbol{s}] = \frac{N_s - 1}{N_s}\,\mathrm{Var}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big],$$

$$\mathbb{E}[A_S^2 \mid \boldsymbol{s}] = \frac{N_s - 1}{N_s}\,\mathrm{Var}(R \mid \boldsymbol{s}).$$

Proof. Fix any sample index i and write $A_S = A_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})$ and $A^*_S = A^*_S(\boldsymbol{s}, \boldsymbol{a}^{(i)})$. All expectations below are over the sampled group, conditional on the indicated state or state–action pair. Since the group mean includes sample i itself,

$$A_S = R^{(i)} - \hat{V}_{N_s} = \frac{N_s - 1}{N_s}R^{(i)} - \frac{1}{N_s}\sum_{j\neq i}R^{(j)}.$$

The two terms on the right are independent conditional on $\boldsymbol{s}$.

Denominator. Every return has conditional mean $V(\boldsymbol{s})$, so

$$\mathbb{E}[A_S \mid \boldsymbol{s}] = \frac{N_s - 1}{N_s} V(\boldsymbol{s}) - \frac{N_s - 1}{N_s} V(\boldsymbol{s}) = 0.$$

Thus the second moment equals the variance. Conditional independence and the identical return distributions give

$$\begin{aligned}
\mathbb{E}[A_S^2 \mid \boldsymbol{s}] &= \mathrm{Var}(A_S \mid \boldsymbol{s}) \\
&= \Big(\frac{N_s - 1}{N_s}\Big)^2 \mathrm{Var}(R \mid \boldsymbol{s}) + \frac{N_s - 1}{N_s^2} \mathrm{Var}(R \mid \boldsymbol{s}) \\
&= \frac{N_s - 1}{N_s} \mathrm{Var}(R \mid \boldsymbol{s}),
\end{aligned}$$

which proves denominator.

Numerator. Conditional on $(\boldsymbol{s}, \boldsymbol{a}^{(i)})$, the current return has mean $Q(\boldsymbol{s}, \boldsymbol{a}^{(i)})$, whereas each other return still has mean $V(\boldsymbol{s})$ by independence. Therefore,

$$\begin{aligned}
\mathbb{E}[A_S \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] &= \frac{N_s - 1}{N_s} Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - \frac{N_s - 1}{N_s} V(\boldsymbol{s}) \\
&= \frac{N_s - 1}{N_s} A_S^*.
\end{aligned}$$

Since A_S^* is determined by $(\boldsymbol{s}, \boldsymbol{a}^{(i)})$, the law of iterated expectations yields

$$\begin{aligned}
\mathbb{E}[A_S\, A_S^* \mid \boldsymbol{s}] &= \mathbb{E}\Big[A_S^*\, \mathbb{E}[A_S \mid \boldsymbol{s}, \boldsymbol{a}^{(i)}] \,\Big|\, \boldsymbol{s}\Big] \\
&= \frac{N_s - 1}{N_s}\, \mathbb{E}[(A_S^*)^2 \mid \boldsymbol{s}] \\
&= \frac{N_s - 1}{N_s}\, \mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big].
\end{aligned}$$

The last equality follows from $A_S^* = Q(\boldsymbol{s}, \boldsymbol{a}^{(i)}) - V(\boldsymbol{s})$ and $V(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}[Q(\boldsymbol{s}, \boldsymbol{a})]$. This proves and completes the proof of Lemma 1 in the main text. □

Lemma A.3 (Conditional return-variance decomposition)**.** *Fix a state $\boldsymbol{s}$ and a policy π, and assume $\mathbb{E}[R^2 \mid \boldsymbol{s}] < \infty$. Let $Q(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}[R \mid \boldsymbol{s}, \boldsymbol{a}]$ and $V(\boldsymbol{s}) = \mathbb{E}[R \mid \boldsymbol{s}]$. Then the total return variance at $\boldsymbol{s}$ satisfies* (9)*:*

$$\mathrm{Var}(R \mid \boldsymbol{s}) = \mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big] + \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big].$$

Proof. By the law of iterated expectations,

$$V(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big].$$

Decompose the centered return as

$$R - V(\boldsymbol{s}) = \big(Q(\boldsymbol{s}, \boldsymbol{a}) - V(\boldsymbol{s})\big) + \big(R - Q(\boldsymbol{s}, \boldsymbol{a})\big).$$

The first term depends only on the current action when $\boldsymbol{s}$ is fixed, whereas the second is the difference between the realized return R and its conditional expected return $Q(\boldsymbol{s}, \boldsymbol{a})$. By the definition of Q,

$$\mathbb{E}\big[R - Q(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}, \boldsymbol{a}\big] = 0.$$

Since $Q(\boldsymbol{s}, \boldsymbol{a}) - V(\boldsymbol{s})$ is fixed conditional on $(\boldsymbol{s}, \boldsymbol{a})$, conditioning on the action and then averaging shows that the cross term vanishes:

$$\mathbb{E}\big[\big(Q(\boldsymbol{s}, \boldsymbol{a}) - V(\boldsymbol{s})\big)\big(R - Q(\boldsymbol{s}, \boldsymbol{a})\big) \,\big|\, \boldsymbol{s}\big] = 0.$$

Squaring the centered-return decomposition and taking conditional expectations therefore gives

$$\begin{aligned}
\mathrm{Var}(R \mid \boldsymbol{s}) &= \mathbb{E}\Big[\big(R - V(\boldsymbol{s})\big)^2 \,\Big|\, \boldsymbol{s}\Big] \\
&= \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\Big[\big(Q(\boldsymbol{s}, \boldsymbol{a}) - V(\boldsymbol{s})\big)^2\Big] + \mathbb{E}\Big[\big(R - Q(\boldsymbol{s}, \boldsymbol{a})\big)^2 \,\Big|\, \boldsymbol{s}\Big] \\
&= \mathrm{Var}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big] + \mathbb{E}_{\boldsymbol{a} \sim \pi(\cdot \mid \boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big].
\end{aligned}$$

In the last equality, the first term is the variance of $Q(\boldsymbol{s}, \boldsymbol{a})$ across actions because its mean is $V(\boldsymbol{s})$. For the second term, $\big(R - Q(\boldsymbol{s}, \boldsymbol{a})\big)^2$ is the squared difference between the realized return and its expected value for a fixed state–action pair. Its expectation conditional on $(\boldsymbol{s}, \boldsymbol{a})$ is, by definition, $\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})$; averaging over actions gives the expected conditional return variance in (9). This completes the proof. □

Theorem A.2 (Theorem 2 in the main text)**.** *Under the assumptions of Theorem 1 and Lemma 1, the optimal per-state shrinkage weight* (5) *is*

$$w^*(\boldsymbol{s}) = \frac{\mathrm{Var}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big]}{\mathrm{Var}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\big[Q(\boldsymbol{s}, \boldsymbol{a})\big] + \mathbb{E}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big]} \in [0, 1],$$

or, equivalently,

$$w^*(\boldsymbol{s}) = 1 - \frac{\mathbb{E}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}\big[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})\big]}{\mathrm{Var}(R \mid \boldsymbol{s})}.$$

Proof. Existence and uniqueness. A nonzero step-level correction requires the group to contain at least two different return values. Thus, the observed advantages of an equal-return group cannot determine a unique weight.

The theorem concerns the population mean-square objective, whose expectation averages over sampled groups rather than conditioning on one realized group. For any fixed group member, the estimate A_S has zero conditional mean because every sampled return has mean $V(\boldsymbol{s})$. By conditional independence of the returns,

$$\mathbb{E}[A_S^2 \mid \boldsymbol{s}] = \left[\Big(\frac{N_s - 1}{N_s}\Big)^2 + \frac{N_s - 1}{N_s^2}\right] \mathrm{Var}(R \mid \boldsymbol{s}) = \frac{N_s - 1}{N_s}\ \mathrm{Var}(R \mid \boldsymbol{s}) > 0.$$

The strict inequality is an assumption of Theorem 1. Since $N_s \geqslant 2$, this identity also implies $\mathrm{Var}(R \mid \boldsymbol{s}) > 0$. The finite return variance also ensures finite second moments of A_S and A_S^*. Thus the mean-square objective is a finite, strictly convex quadratic in w, and Theorem A.1 gives its unique minimizer as the ratio in (5).

An individual sampled group may still have identical returns even when $\mathrm{Var}(R \mid \boldsymbol{s}) > 0$; this does not contradict population-level uniqueness. By contrast, if $\mathrm{Var}(R \mid \boldsymbol{s}) = 0$, then $R = V(\boldsymbol{s})$ almost surely and $Q(\boldsymbol{s}, \boldsymbol{a}) = V(\boldsymbol{s})$ for $\pi(\cdot \mid \boldsymbol{s})$-almost every action. Both A_S and A_S^* are then zero almost surely, every weight minimizes the objective, and the variance-ratio formulas are undefined. This degenerate case is excluded by the positive-second-moment assumption.

Interpretive form (10)*.* Substituting the two moment identities from Lemma A.2 into (5) cancels the common factor $\frac{N_s - 1}{N_s}$:

$$w^*(\boldsymbol{s}) = \frac{\mathrm{Var}_{\boldsymbol{a}}[Q(\boldsymbol{s}, \boldsymbol{a}) \mid \boldsymbol{s}]}{\mathrm{Var}(R \mid \boldsymbol{s})}.$$

Expanding the denominator by Lemma A.3 gives (10):

$$w^*(\boldsymbol{s}) = \frac{\mathrm{Var}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}[Q(\boldsymbol{s}, \boldsymbol{a})]}{\mathrm{Var}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}[Q(\boldsymbol{s}, \boldsymbol{a})] + \mathbb{E}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})]}.$$

The denominator equals the strictly positive total return variance, and both variance components are non-negative. Therefore $w^*(\boldsymbol{s}) \in [0, 1]$. It equals 1 exactly when the expected conditional return variance is zero, and 0 exactly when $Q(\boldsymbol{s}, \boldsymbol{a})$ is constant for $\pi(\cdot \mid \boldsymbol{s})$-almost every action.

Computable form (11)*.* By the law of total variance (9), the action-value variance equals the total return variance minus the expected conditional return variance. Substituting this identity yields

$$\begin{aligned} w^*(\boldsymbol{s}) &= \frac{\mathrm{Var}(R \mid \boldsymbol{s}) - \mathbb{E}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})]}{\mathrm{Var}(R \mid \boldsymbol{s})} \\ &= 1 - \frac{\mathbb{E}_{\boldsymbol{a}\sim\pi(\cdot|\boldsymbol{s})}[\mathrm{Var}(R \mid \boldsymbol{s}, \boldsymbol{a})]}{\mathrm{Var}(R \mid \boldsymbol{s})}. \end{aligned}$$

This proves the equivalent computable form (11) and completes the proof of Theorem 2 in the main text. □

B PSEUDO CODE

Algorithm 1 summarizes the AdaStep training procedure, following the rollout, grouping, and policy-update structure of GiGPO (Feng et al., 2025). AdaStep replaces the fixed coefficient ω with a coefficient estimated separately for each anchor-state group; the corresponding operations are italicized in the algorithm. Within each task, we use anchor states as hash-table keys to collect step-level groups, and further partition each group by the observed action to estimate within-action return variability. These statistics can be accumulated using counts, sums, and sums of squared returns, without an additional critic or additional rollouts.

Computing the adaptive coefficient. For a trajectory τ_i of length T_i, let $R_{i,t} = \sum_{k=t}^{T_i} \gamma^{k-t} r_{i,k}$ denote the discounted return from step t. Write $\tilde{s}_{i,t}$ for its anchor-state key, while $s_{i,t}$ denotes the state or context conditioning the policy. For each task, define the index group $\mathcal{G}(\tilde{s}) = \{(i,t) : \tilde{s}_{i,t} = \tilde{s}\}$. Let $\mathcal{A}_{\tilde{s}}$ be the set of distinct actions observed in the group, and define

$$\begin{aligned} \mathcal{G}(\tilde{s}, a) &= \{(i,t) \in \mathcal{G}(\tilde{s}) : a_{i,t} = a\}, \\ N_{\tilde{s}} = |\mathcal{G}(\tilde{s})|, \qquad n_{\tilde{s},a} &= |\mathcal{G}(\tilde{s}, a)|, \qquad \widehat{p}(a \mid \tilde{s}) = \frac{n_{\tilde{s},a}}{N_{\tilde{s}}}. \end{aligned} \tag{16}$$

Denote the empirical variance of returns in $\mathcal{G}(\tilde{s})$ by $\widehat{v}_{\tilde{s}}$, and the empirical variance within $\mathcal{G}(\tilde{s}, a)$ by $\widehat{v}_{\tilde{s},a}$. The variance denominators are $N_{\tilde{s}}$ for the full group and $n_{\tilde{s},a}$ for each action subgroup. Substituting these statistics into Eq. (11) gives the empirical coefficient

$$\widehat{w}(\tilde{s}) = 1 - \frac{\sum_{a \in \mathcal{A}_{\tilde{s}}} \widehat{p}(a \mid \tilde{s})\, \widehat{v}_{\tilde{s},a}}{\widehat{v}_{\tilde{s}}}. \tag{17}$$

Here $\widehat{w}$ is a finite-sample estimate of the population coefficient w^*, and all variance statistics use the unnormalized discounted returns. We evaluate Eq. (17) only if $|\mathcal{A}_{\tilde{s}}| \geq 2$, $n_{\tilde{s},a} \geq 2$ for every observed action, and $\widehat{v}_{\tilde{s}} > 0$. Otherwise, we set $\widehat{w}(\tilde{s}) = 1$, following the fallback choice in Section 5.3. If the group returns are identical, we set the step-level advantages to zero and skip the variance ratio. The combined advantage is

$$A_{i,t} = A_E(\tau_i) + \widehat{w}(\tilde{s}_{i,t}) A_S(s_{i,t}, a_{i,t}), \tag{18}$$

where A_S uses the benchmark-specific normalization in Appendix C. All coefficients and advantages are computed from the old-policy rollouts and held fixed during policy optimization.

Algorithm 1 Training LLM Agents with AdaStep

Require: Initial policy π_θ, fixed reference policy π_{ref}, task distribution $p(X)$
Require: Group size N, maximum episode length $T_{\max}$, discount γ, clipping parameter ϵ, KL coefficient β, and benchmark-specific step-normalization setting

for each training iteration **do**
  $\theta_{\text{old}} \leftarrow \theta$; sample a task batch from $p(X)$; initialize buffer $\mathcal{B} \leftarrow \varnothing$
  for each sampled task x **do**
    Multi-step rollout: initialize N identical environments for x
    for $t = 1, \ldots, T_{\max}$, until all environments terminate **do**
      Sample $a_{i,t} \sim \pi_{\theta_{\text{old}}}(\cdot \mid s_{i,t}, x)$ for each active environment i
      Execute actions, record $(s_{i,t}, a_{i,t}, r_{i,t}, s_{i,t+1})$, and stop terminated episodes
    end for
    Compute $R_E(\tau_i) = \sum_{t=1}^{T_i} r_{i,t}$ and $R_{i,t} = \sum_{k=t}^{T_i} \gamma^{k-t} r_{i,k}$
    Compute $A_E(\tau_i)$ using Eq. (1); set $A_E = 0$ if all episode returns agree
    Grouping: hash transitions by anchor state to build $\mathcal{G}(\tilde{s})$ within task x
    for each anchor-state group $\mathcal{G}(\tilde{s})$ **do**
      Compute the group return mean $\overline{R}_{\tilde{s}}$
      Initialize the fallback coefficient $\widehat{w}(\tilde{s}) \leftarrow 1$
      if all returns in $\mathcal{G}(\tilde{s})$ are identical **then**
        Set $A_S(s_{i,t}, a_{i,t}) \leftarrow 0$ for all $(i,t) \in \mathcal{G}(\tilde{s})$
      else
        Compute $\widehat{v}_{\tilde{s}}$ and set $A_S(s_{i,t}, a_{i,t}) \leftarrow R_{i,t} - \overline{R}_{\tilde{s}}$
        Divide A_S by $\sqrt{\widehat{v}_{\tilde{s}}}$ if the benchmark uses w/ std
        Partition by action to obtain $\mathcal{G}(\tilde{s}, a)$ *and counts* $n_{\tilde{s},a}$
        if $|\mathcal{A}_{\tilde{s}}| \geq 2$ and $n_{\tilde{s},a} \geq 2$ for every observed action a **then**
          Compute within-action variances $\widehat{v}_{\tilde{s},a}$ *from the raw returns*
          Set $\widehat{w}(\tilde{s}) \leftarrow 1 - \dfrac{\sum_a (n_{\tilde{s},a}/N_{\tilde{s}})\widehat{v}_{\tilde{s},a}}{\widehat{v}_{\tilde{s}}}$
        end if
      end if
      Combine $A_{i,t} \leftarrow A_E(\tau_i) + \widehat{w}(\tilde{s}) A_S(s_{i,t}, a_{i,t})$ *for each member*
    end for
    Store the task's valid transitions and fixed advantages in $\mathcal{B}$
  end for
  Policy update: optimize θ on $\mathcal{B}$ by maximizing Eq. (20)
end for
Ensure: Trained policy π_θ

B.1 Policy Optimization Objective

AdaStep uses the clipped policy objective of GiGPO with the adaptive combined advantage in Eq. (18). For an action sampled at state $s_{i,t}$ under task x, the importance sampling ratio is

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t} \mid s_{i,t}, x)}{\pi_{\theta_{\text{old}}}(a_{i,t} \mid s_{i,t}, x)}. \tag{19}$$

Let $M = \sum_{i=1}^{N} T_i$ be the number of valid environment steps in a task group. We maximize

$$\begin{aligned} \mathcal{J}_{\text{AdaStep}}(\theta) = \mathbb{E}_{\substack{x \sim p(X) \\ \{\tau_i\}_{i=1}^{N} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)}} \Bigg[\frac{1}{M} \sum_{i=1}^{N} \sum_{t=1}^{T_i} \Big(& \\ \min\{\rho_{i,t}(\theta) A_{i,t},\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, A_{i,t}\} & \\ - \beta D_{\text{KL}}(\pi_\theta(\cdot \mid s_{i,t}, x) \,\|\, \pi_{\text{ref}}(\cdot \mid s_{i,t}, x)) \Big) \Bigg]. & \end{aligned} \tag{20}$$

The parameter ϵ defines the importance-ratio clipping interval, and β controls the KL penalty toward the fixed reference policy π_{ref}. The old policy is refreshed before each rollout iteration. For a task batch, we average the per-task terms in Eq. (20). Only valid steps contribute; if every trajectory has length T, then $M = NT$. Replacing $\widehat{w}(\tilde{s}_{i,t})$ with a fixed ω in Eq. (18) recovers the GiGPO advantage and its corresponding clipped objective.

C EXPERIMENTAL DETAILS

Benchmarks. We evaluate on three long-horizon benchmarks: ALFWorld (Shridhar et al., 2021), WebShop (Yao et al., 2022), and ScienceWorld (Wang et al., 2022). ALFWorld is a text-based embodied environment in which agents complete multi-step household tasks, such as heating, cleaning, and relocating objects, with sparse rewards upon task completion; we set the maximum number of environment steps to 50. WebShop is a simulated shopping environment with 1.18 million products, requiring agents to search, select product options, and purchase items that satisfy natural-language instructions; we set the maximum number of environment steps to 30. ScienceWorld is a text-based scientific reasoning environment in which agents conduct virtual experiments such as measuring melting points and testing electrical conductivity; we set the maximum number of environment steps to 30. In our experiments, ScienceWorld evaluation is restricted to eight simple tasks.

Baselines. We compare against baselines across three categories: (1) *Frontier models:* DeepSeek-V4.1-Flash (DeepSeek-AI et al., 2026) and GLM-5.2 (GLM-5-Team et al., 2026), evaluated with ReAct prompting (Yao et al., 2023) as reference points for frontier model capabilities. (2) *Prompting methods:* ReAct (Yao et al., 2023), which interleaves reasoning and actions, and Reflexion (Shinn et al., 2023), which uses verbal self-reflection and episodic memory to improve subsequent attempts without updating model parameters. (3) *RL training methods:* PPO (Schulman et al., 2017), a standard actor–critic algorithm; GRPO (Shao et al., 2024) and GiGPO (Feng et al., 2025), which estimate relative advantages from trajectory groups without learned critics, with GiGPO additionally using step-level groups for fine-grained credit assignment; and HGPO (He et al., 2026), which incorporates historical context into anchor-state grouping and adaptively aggregates advantages across hierarchical groups to reduce bias in step-level advantage estimation.

Implementation. We use Qwen2.5-7B-Instruct (Qwen et al., 2025), Qwen3-4B, and Qwen3-1.7B (Yang et al., 2025) as backbone models. All Qwen3 models are run with their built-in thinking mode disabled throughout training and evaluation. For fair comparison, all RL methods share the same training pipeline and common hyperparameters specified below. We follow the hyperparameter configuration of GiGPO (Feng et al., 2025) for ALFWorld and WebShop, with the WebShop episode limit extended from 15 to 30 steps. AdaStep computes its adaptive weights from the unnormalized return statistics of step-level groups, replacing GiGPO's fixed coefficient ω without introducing additional tunable hyperparameters. For HGPO, we set the maximum historical context length to $K = 2$ throughout our experiments.

Hyperparameters for ALFWorld. All RL methods use a maximum prompt length of 2,048 tokens and a maximum response length of 512 tokens, with at most 50 environment steps per episode. Training uses a rule-based task reward of 10 for success and 0 for failure, together with a reward penalty of -0.1 for each invalid action. These practical rewards rescale the theoretical binary success signal and add an invalid-action penalty. For group-based methods (RLOO, GRPO, GiGPO, HGPO, and AdaStep), each rollout batch contains 16 distinct task groups with 8 trajectories per group, yielding $16 \times 8 = 128$ environment instances. PPO instead collects rollouts from 128 separate environment instances without arranging them into trajectory groups. The sampling temperature is 1.0 during training rollouts and 0.4 during validation.

The actor learning rate is 1×10^{-6} for all RL methods. PPO additionally trains a critic with a learning rate of 1×10^{-5}; the group-based methods do not use a critic. We use a mini-batch size of 256 and a KL-divergence loss coefficient of 0.01. For the GiGPO baseline, the step-level advantage coefficient is fixed at $\omega = 1$ without further tuning, and the return discount factor is $\gamma = 0.95$. For the main results in Table 1, GiGPO, HGPO, and AdaStep compute step-level advantages with standard-deviation normalization (*w/ std*).

Hyperparameters for WebShop. All RL methods use a maximum prompt length of 4,096 tokens and a maximum response length of 512 tokens. Each episode is limited to 30 environment steps. We use the same rule-based reward as in ALFWorld: 10 for task success, 0 for failure, and a penalty of -0.1 for each invalid action. Each group-based method samples 16 task groups per rollout batch, with 8 trajectories in each group, for a total of 128 environment instances. PPO uses 128 separate environment instances. The rollout and validation temperatures are 1.0 and 0.4, respectively.

The actor learning rate is 1×10^{-6}, and the critic learning rate, applicable only to PPO, is 1×10^{-5}. The mini-batch size is 64, and the KL-divergence loss coefficient is 0.01. As in ALFWorld, the

GiGPO baseline uses a fixed step-level advantage coefficient $\omega = 1$ without additional tuning and a discount factor $\gamma = 0.95$. For the main results in Table 1, GiGPO, HGPO, and AdaStep compute step-level advantages without standard-deviation normalization (*w/o std*).

Hyperparameters for ScienceWorld. We evaluate on eight simple ScienceWorld tasks. All RL methods use a maximum prompt length of 2,048 tokens and a maximum response length of 512 tokens, with at most 30 environment steps per episode. Training uses a rule-based task reward of 10 for success and 0 for failure, together with a reward penalty of -0.1 for each invalid action. For group-based methods (RLOO, GRPO, GiGPO, HGPO, and AdaStep), each rollout batch contains 16 distinct task groups with 8 trajectories per group, yielding $16 \times 8 = 128$ environment instances. PPO instead collects rollouts from 128 separate environment instances without arranging them into trajectory groups. The sampling temperature is 1.0 during training rollouts and 0.4 during validation.

The actor learning rate is 1×10^{-6} for all RL methods. PPO additionally trains a critic with a learning rate of 1×10^{-5}; the group-based methods do not use a critic. We use a mini-batch size of 256 and a KL-divergence loss coefficient of 0.01. For the GiGPO baseline, the step-level advantage coefficient is fixed at $\omega = 1$ without further tuning, and the return discount factor is $\gamma = 0.95$. For the main results in Table 1, GiGPO, HGPO, and AdaStep compute step-level advantages with standard-deviation normalization (*w/ std*).

Computing details. All experiments were conducted on 8 NVIDIA H100 GPUs for 160 training steps, with each run taking no more than 16 hours. Results reported with means and standard deviations were computed over three independent runs with different random seeds.

D Prompt Templates

Figures 4–6 show the WebShop, ALFWorld, and ScienceWorld prompt templates. Braced placeholders are filled with task and interaction information at each step. All three prompts enclose reasoning within `<thought>` and `</thought>` tags, and the chosen action within `<action>` and `</action>` tags. These delimiters specify the response format. All Qwen3 models are run with their built-in thinking mode disabled, as stated in Appendix C.

Prompt Template for WebShop

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 <thought> </thought> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Figure 4: The WebShop prompt template with `<thought>` and `</thought>` reasoning delimiters.

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 <thought> </thought> tags. Once you've finished your reasoning, you should choose an admissible action for current step and present it within <action> </action> tags.

Figure 5: The ALFWorld prompt template with <thought> and </thought> reasoning delimiters.

Prompt Template for ScienceWorld

You are an expert agent in a text-based interactive science environment (ScienceWorld).

Your task is: {task_description}

Prior to this step, you have already taken {step_count} step(s). This reasoning process MUST be enclosed within <thought> </thought> tags.

Once you've finished your reasoning, write the single action you choose, enclosed within <action> </action> tags.

Figure 6: The ScienceWorld prompt template with <thought> and </thought> reasoning delimiters.

E Details of the WebShop Step-Credit Analysis

This appendix describes the offline diagnostics in Section 5.5. We compare coefficient rules on the same historical return samples, rather than collecting a new set of policy-training runs for this analysis. The comparison contains 3,666 WebShop anchor-state groups. We write G_i for the recorded discounted step return of sample i and a_i for its action, and suppress the split index below.

Fit and reference subsets. Within each anchor-state group s, the return observations for each action are divided into two halves: a fit subset $\mathcal{F}_{s,a}$ and a disjoint reference subset $\mathcal{H}_{s,a}$. This row-level splitting is repeated 32 times per group, and all four coefficient rules use the same splits. The fit subset is used to estimate group statistics and the adaptive coefficient. Here, "fit" denotes direct empirical estimation; no additional value model or critic is trained. The reference subset supplies action-level empirical targets and is not used to choose the adaptive coefficient.

Estimating the coefficient from the fit subset. Let $\mathcal{F}_s = \bigcup_a \mathcal{F}_{s,a}$, $n^{\text{fit}}_{s,a} = |\mathcal{F}_{s,a}|$, and $N^{\text{fit}}_s = |\mathcal{F}_s|$. The fit-subset action frequencies and return means are

$$\begin{aligned} \widehat{p}^{\text{fit}}_{s,a} &= \frac{n^{\text{fit}}_{s,a}}{N^{\text{fit}}_s}, & \widehat{\mu}^{\text{fit}}_{s,a} &= \frac{1}{n^{\text{fit}}_{s,a}} \sum_{i \in \mathcal{F}_{s,a}} G_i, \\ \widehat{\mu}^{\text{fit}}_{s} &= \sum_a \widehat{p}^{\text{fit}}_{s,a} \widehat{\mu}^{\text{fit}}_{s,a}, & X_i &= G_i - \widehat{\mu}^{\text{fit}}_{s}, \quad i \in \mathcal{F}_s. \end{aligned} \tag{21}$$

The signal X_i is the unnormalized, return-centered step credit. We estimate within-action return variation and total return variance using only fit rows:

$$\widehat{W}^{\text{fit}}_s = \sum_a \widehat{p}^{\text{fit}}_{s,a} \widehat{\text{Var}}_{\mathcal{F}_{s,a}}(G), \qquad \widehat{D}^{\text{fit}}_s = \widehat{\text{Var}}_{\mathcal{F}_s}(G). \tag{22}$$

For this diagnostic, we apply the empirical variance-ratio rule of Eq. (17) to the fit statistics and clip the resulting coefficient to $[0, 1]$:

$$\widehat{w}_s = \mathrm{clip}\left(1 - \frac{\widehat{W}_s^{\mathrm{fit}}}{\widehat{D}_s^{\mathrm{fit}}}, 0, 1\right). \tag{23}$$

We use the fallback coefficient $\widehat{w}_s = 1$ when the fit subset does not support coefficient estimation, following Section 5.3. Equal-return groups have zero centered step credit. The four rules apply $c_s X_i$ with $c_s = 1$ (GiGPO), $c_s = 0.5$ (fixed scaling), $c_s = 0$ (step off), or $c_s = \widehat{w}_s$ (AdaStep). Thus, fitting consists of computing empirical means, variances, and a scalar ratio; the coefficient is not optimized against the reference MSE.

Scaled within-action variation. Let $p_{s,a}$ denote the empirical action-frequency weights used in the diagnostic, $W_s = \sum_a p_{s,a} \widehat{\mathrm{Var}}(G \mid s, a)$ the within-action return variation, and $D_s > 0$ the common total-return variance scale. These diagnostic quantities are held fixed across all four coefficient rules within each split; the fit-only statistics above are used to estimate $\widehat{w}_s$. The first diagnostic is

$$\mathcal{V}_s(c_s) = \frac{c_s^2 W_s}{D_s}. \tag{24}$$

It measures the within-action fluctuation remaining after scaling the local credit. The factor c_s^2 follows from the scaling property of variance. Consequently, $c_s = 0.5$ retains one quarter of the unscaled variation and $c_s = 0$ removes it entirely. This metric measures the stability of the weighted credit, not a change in the environment's raw return variance; its reduction alone does not establish improved estimation.

Held-out action-level reference MSE. For each action, compute its empirical mean return using reference rows, then center these means using the diagnostic's action-frequency weights:

$$\overline{G}_{s,a}^{\mathrm{ref}} = \frac{1}{|\mathcal{H}_{s,a}|} \sum_{j \in \mathcal{H}_{s,a}} G_j, \qquad B_{s,a} = \overline{G}_{s,a}^{\mathrm{ref}} - \sum_b p_{s,b} \overline{G}_{s,b}^{\mathrm{ref}}. \tag{25}$$

The second diagnostic compares each fit-row credit with the reference for its action, using the same positive variance scale for all coefficient rules:

$$\mathcal{E}_s(c_s) = \frac{1}{D_s N_s^{\mathrm{fit}}} \sum_{i \in \mathcal{F}_s} \left(c_s X_i - B_{s,a_i}\right)^2. \tag{26}$$

Lower values indicate closer agreement with the held-out empirical action-level signal. Removing the step component can increase this error when the reference action advantages are nonzero. The reference itself is estimated from finitely many rows, so this is a proxy error, not an error against true action values or ground-truth advantages.

Aggregation and interpretation. For each metric, we average over the 32 splits within each group and then equally over the 3,666 groups. Figure 3(c,d) reports $100\,\overline{\mathcal{V}}(c)/\overline{\mathcal{V}}(1)$ and $100\,\overline{\mathcal{E}}(c)/\overline{\mathcal{E}}(1)$, so each metric has its own GiGPO baseline of 100. Before this GiGPO-relative rescaling, the aggregate diagnostic values for GiGPO and AdaStep are 0.1576 and 0.0564 for within-action variation, and 0.3313 and 0.2180 for reference MSE. These are descriptive results on historical samples. Rows may share trajectories, and groups may recur across training updates; the repeated splits are not independent training seeds. We therefore do not attach independent-seed confidence intervals or interpret the proxy error as a direct measure of downstream policy performance.

F Limitations and Future Work

AdaStep estimates its adaptive coefficient from the return statistics of the samples that share an anchor state, so it relies on constructing step-level state groups across multiple rollouts. In the three benchmarks studied here, the environment state is discrete and exactly reproduced, so identical states can be matched directly and the groups required for estimation are readily available. In more general settings, such as continuous, partially observable, or open-ended environments, the same situation is rarely revisited verbatim, and how to define and construct state groups that remain suitable for the variance-ratio estimate is an open question. We therefore leave the design of state abstractions or learned grouping criteria that preserve the reliability interpretation of $w^*(\boldsymbol{s})$ to future work.