GRANULARITY-ADAPTIVE CREDIT ASSIGNMENT FOR LONG-HORIZON LLM AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-horizon language-model agents trained with reinforcement learning often receive sparse outcome rewards that do not reveal which decisions along a trajectory deserve credit. Episode-level advantages provide coarse trajectory-wide credit, while step-level comparisons offer finer resolution with context-dependent estimation noise. We propose Granularity-Adaptive Credit Assignment (GACA), a critic-free method that adaptively mixes episode- and step-level credit for each decision during policy optimization. GACA normalizes the sampled response's mean per-token negative log-likelihood (NLL) within each trajectory and uses the resulting criticality score to determine the step-specific mixture. The computation reuses rollout log-probabilities without additional training rollouts or model evaluations. Our analysis characterizes optimal score-dependent mixing and derives conditions linking expected NLL to a lower bound on the preferred step-level weight. Across ALFWorld and WebShop with 1.5B and 7B backbones, GACA achieves the highest reported mean success rates among the compared methods, while introducing negligible additional computation.

1 INTRODUCTION

Language-model agents act over many steps to complete household tasks, browse web stores, and use external tools (Yao et al., 2022b; Shridhar et al., 2020; Yao et al., 2022a). Reinforcement learning can improve these agents using task outcomes (Ouyang et al., 2022; Guo et al., 2025), but a sparse terminal reward leaves a difficult credit-assignment problem: which decisions should receive the learning signal? A trajectory can contain both consequential choices and actions whose alternatives would have little effect on the outcome (Arjona-Medina et al., 2019).

Critic-free methods avoid training a value model by comparing several rollouts of the same task. GRPO (Shao et al., 2024) computes an episode-level relative advantage and broadcasts it to every action. GiGPO (Feng et al., 2026) adds local resolution by grouping steps with the same anchor observation and comparing their continuation returns. Its fixed combination of episode- and step-level signals leaves a further question: *should every decision rely on these two credit estimates in the same proportion?*

The two granularities offer a decision-dependent estimation trade-off. Episode credit spreads the outcome over all decisions, while step credit separates local continuations using finite, context-dependent comparison groups. We propose Granularity-Adaptive Credit Assignment (GACA), which adapts their relative contribution during policy optimization.

For each completed rollout, GACA computes the sampled response's mean per-token negative log-likelihood (NLL) from old-policy log-probabilities. Dividing by the trajectory mean gives a criticality score that measures how surprising the response is relative to other responses in that interaction. A bounded mapping converts this score into a step-specific mixing weight, placing more weight on step-level credit at higher scores. These computations reuse quantities already available for the policy update, requiring no additional model or training rollouts.

Our analysis characterizes how a score-dependent mixture balances the errors of the two credit estimates. It then identifies conditions under which greater variation in action values favors step-

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.

level credit and expected NLL provides a lower bound on that variation. These results motivate increasing the weight of local credit with the NLL-based score.

GACA complements methods that refine the step estimator, including history-consistent grouping in HGPO (He et al., 2026), behavior-aware grouping in BiPACE (Wang et al., 2026a), and progress-conditioned credit in ProGPO (Yang et al., 2026). Uncertainty-based modulation in AEM and STARE rescales an existing advantage (Zhao et al., 2026; Luo et al., 2026); GACA uses uncertainty to control the relative contribution of two credit granularities. Appendix A discusses these connections.

Our evaluation covers ALFWorld and WebShop with Qwen2.5-Instruct at 1.5B and 7B scales. On ALFWorld 1.5B, matched ablations compare GACA with five fixed mixtures and within-trajectory random score assignment. Adaptive mixing reaches 96.88% success, compared with 93.05% for the best tested fixed mixture and 88.54% for random assignment. These comparisons examine both the choice of mixing weights and their assignment to individual decisions.

2 Preliminaries

2.1 Sequential interaction and policy optimization

At step t, the full pre-action state s_t contains the task, interaction history, and current observation. The agent samples a response $a_t \sim \pi_\theta(\cdot \mid s_t)$ containing reasoning and an action command. The environment executes its parsed action $U_t = \phi(a_t)$, returns reward r_t, and continues until termination at T. The trajectory return is $R(\tau) = \sum_t r_t$, and the discounted step return is $\hat{g}_t = \sum_{k=t}^{T} \gamma^{k-t} r_k$. Each task produces a group of G trajectories for relative advantage estimation.

A per-step advantage A_t^{GACA} is broadcast to the response's valid tokens. With token importance ratio $\rho_{t,j} = \pi_\theta(a_t^j \mid s_t, a_t^{<j}) / \pi_{\theta_{\mathrm{old}}}(a_t^j \mid s_t, a_t^{<j})$, training uses the clipped surrogate (Schulman et al., 2017)

$$\mathcal{L}_{\mathrm{PG}} = -\mathbb{E}_t \left[\frac{1}{|a_t|} \sum_j \min\left(\rho_{t,j} A_t^{\mathrm{GACA}}, \mathrm{clip}(\rho_{t,j}, 1-\epsilon, 1+\epsilon) A_t^{\mathrm{GACA}}\right) \right] + \beta_{\mathrm{KL}} \widehat{\mathrm{KL}}(\pi_\theta \| \pi_{\mathrm{ref}}). \tag{1}$$

GACA supplies the advantage used in this update. For ALFWorld 1.5B, the expectation uses response weights $w_t = |a_t| / \sum_{k \in \mathcal{B}} |a_k|$ in optimization batch $\mathcal{B}$, giving a mean over valid tokens. Appendix E specifies loss reduction and regularization.

2.2 Episode- and step-level credit

The episode-side advantage compares the outcome score $\widetilde{R}_t = R(\tau) - p_t$ with its prompt group g, where p_t is the configured local invalid-action penalty:

$$A_t^{\mathrm{E}} = \frac{\widetilde{R}_t - \mu_g}{\sigma_g + \varepsilon}. \tag{2}$$

The trajectory outcome is shared across decisions, with the validity adjustment applied before normalization. Group statistics are computed over step rows; Appendix E gives the conventions.

Following GiGPO, steps with identical anchor observations form a group $\mathcal{G}$. Their discounted returns give the step advantage

$$A_t^{\mathrm{S}} = \frac{\hat{g}_t - \mu_{\mathcal{G}}}{\sigma_{\mathcal{G}} + \varepsilon}, \qquad \mu_{\mathcal{G}} = |\mathcal{G}|^{-1} \sum_{k \in \mathcal{G}} \hat{g}_k. \tag{3}$$

The standard-deviation division in both equations is used for ALFWorld and omitted for WebShop. A singleton group uses prompt-group normalization of its step return. Anchor matching uses observations and can group steps with different histories; the resulting estimation error is included in the analysis below. GiGPO combines the two signals as $A^{\mathrm{E}} + \omega A^{\mathrm{S}}$ with a fixed coefficient ω.

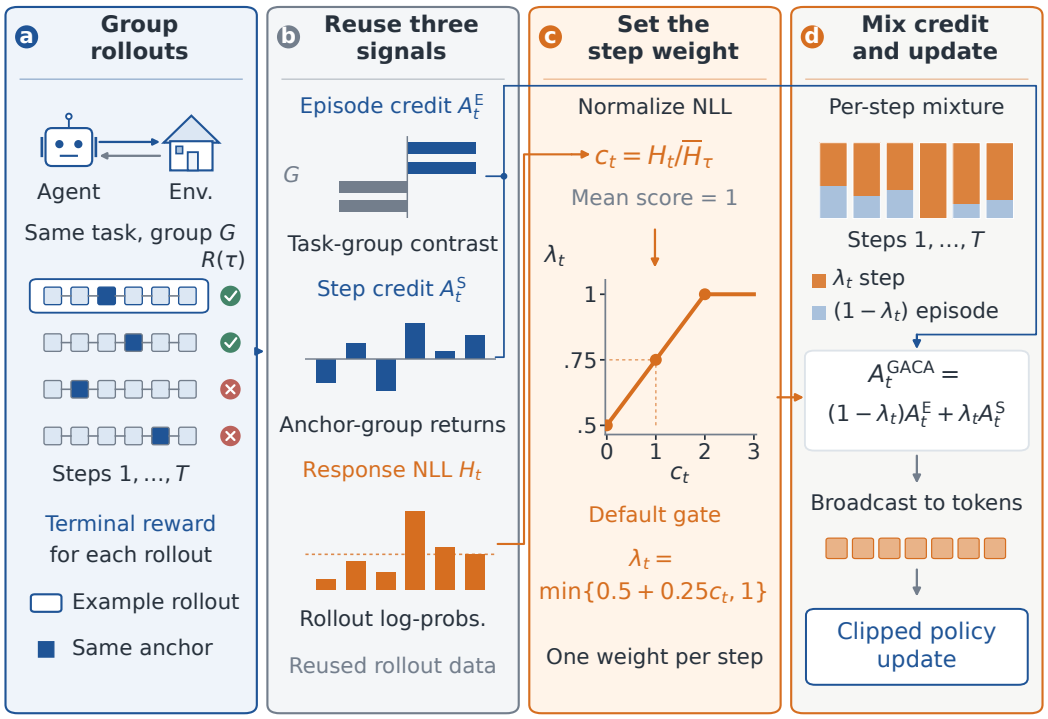


Figure 1: GACA during policy optimization. Completed rollout groups provide episode credit, anchor-grouped step credit, and sampled-response NLL. Trajectory normalization gives c_t, and the default gate assigns the step-level weight $\lambda_t = \min\{0.5+0.25c_t, 1\}$. The mixed advantage is broadcast to response tokens for the clipped policy update. Trajectories and signal values are schematic.

3 Granularity-Adaptive Credit Assignment

During advantage computation, GACA assigns a mixing weight to each environment step using a criticality score computed from the completed rollout (Figure 1). It reuses the two group-relative advantages and the logged token probabilities, adding only scalar score and mixing operations.

3.1 An NLL-based criticality score

For the full response sampled at step t, including reasoning and the action command, define its mean per-token NLL and trajectory-normalized criticality as

$$H_t = -\frac{1}{|a_t|}\sum_{j=1}^{|a_t|}\log \pi_{\theta_{\text{old}}}(a_t^j \mid s_t, a_t^{<j}), \tag{4}$$

$$c_t = \frac{H_t}{\overline{H}_\tau}, \qquad \overline{H}_\tau = \frac{1}{T}\sum_{k=1}^{T} H_k. \tag{5}$$

The criticality score is an NLL-based proxy for allocating credit: $c_t > 1$ identifies a response that is more surprising than the trajectory average. For $\overline{H}_\tau > 0$, the scores have mean one and preserve the ordering of H_t within the trajectory. Normalization puts decisions from the same interaction on a common relative scale. Scores are computed after rollout completion, when the full trajectory is available for advantage estimation.

Algorithm 1 GACA advantage computation

Require: group rollouts $\{\tau_i\}_{i=1}^{G}$ with returns, step returns, action-validity flags, anchor observations, and token log-probabilities; hyperparameters $\lambda_0, \alpha_\lambda$

1: compute episode advantages A_t^{E} using prompt-group statistics ▷ Eq. 2
2: group matching anchor observations and compute A_t^{S}, with singleton fallback ▷ Eq. 3
3: **for** each trajectory τ **do**
4: compute sampled-response NLL H_t and criticality scores c_t ▷ Eqs. 4–5
5: $\lambda_t \leftarrow \min\{\lambda_0(1+\alpha_\lambda c_t), 1\}$ for every step
6: $A_t^{\mathrm{GACA}} \leftarrow (1-\lambda_t)A_t^{\mathrm{E}} + \lambda_t A_t^{\mathrm{S}}$ for every step
7: **end for**
8: **return** $\{A_t^{\mathrm{GACA}}\}$ for the policy update in Eq. 1

3.2 Bounded adaptive mixing

We map criticality to a bounded step-level weight and combine the two advantages:

$$\lambda_t = \min\{\lambda_0(1+\alpha_\lambda c_t), 1\}, \qquad \lambda_0 \in [0,1], \quad \alpha_\lambda \geq 0, \tag{6}$$

$$A_t^{\mathrm{GACA}} = (1-\lambda_t)A_t^{\mathrm{E}} + \lambda_t A_t^{\mathrm{S}}. \tag{7}$$

The parameter λ_0 sets the step-weight floor, and α_λ controls the increase with criticality before saturation. Setting $\alpha_\lambda = 0$ gives a fixed mixture. The defaults $\lambda_0 = \alpha_\lambda = 0.5$ yield $\lambda_t = 0.75$ at $c_t = 1$ and reach the upper cap at $c_t = 2$. The gate retains a positive step-level coefficient at low criticality and assigns all weight to the step advantage once $c_t \geq 2$.

Appendix G reports sensitivity to the modulation strength. Rollout-derived scores and advantages are held fixed during each policy update. Algorithm 1 summarizes the computation; Appendix E gives the normalization and singleton conventions.

3.3 Why adapt the credit granularity?

We study the squared error of mixed credit estimates relative to the action advantage under a fixed rollout policy π. Let $U = \phi(X)$ be the environment action parsed from response X, and let $Q^\pi(s,u)$ be its expected discounted return from full state s under π. The action advantage is $A^\pi(s,u) = Q^\pi(s,u) - \mathbb{E}_{U|s}Q^\pi(s,U)$.

We use the normalized target $Y = A^\pi/\sigma_A$, where $\sigma_A^2 = \mathbb{E}[(A^\pi)^2] > 0$ is computed over decision steps sampled from π. The inputs $E = A^{\mathrm{E}}$ and $S = A^{\mathrm{S}}$ retain their implemented normalizations. Their errors relative to Y include grouping effects, scale mismatch, and shared rollout noise. Full statements and proofs are in Appendix B–C.

Conditional mixing balances the two estimation errors. At a given score c, let $M_E = \mathbb{E}[(E-Y)^2 \mid c]$ and $M_S = \mathbb{E}[(S-Y)^2 \mid c]$ be the two mean squared errors, and $\chi = \mathbb{E}[(E-Y)(S-Y) \mid c]$ their error cross-moment. With finite second moments and $q = M_E + M_S - 2\chi = \mathbb{E}[(S-E)^2 \mid c] > 0$, the optimal conditional weight is

$$f^\star(c) = \Pi_{[0,1]}\left(\frac{M_E(c) - \chi(c)}{M_E(c) + M_S(c) - 2\chi(c)}\right), \tag{8}$$

where $\Pi_{[0,1]}$ projects onto the unit interval. This oracle improves on the best constant mixture whenever its unique optimum varies over scores with $q > 0$ (Proposition 1). For an implemented gate f, let $u(c) = (M_E - \chi)/q$, $\lambda_{\mathrm{fix}}^\star = \Pi_{[0,1]}\{\mathbb{E}[M_E - \chi]/\mathbb{E}[q]\}$, and $\mathcal{R}(f) = \mathbb{E}[((1-f(c))E + f(c)S - Y)^2]$. Proposition 1 gives

$$\mathcal{R}(f) - \mathcal{R}(\lambda_{\mathrm{fix}}^\star) = \mathbb{E}\left[q(c)\left\{[f(c)-u(c)]^2 - [\lambda_{\mathrm{fix}}^\star - u(c)]^2\right\}\right]. \tag{9}$$

Terms with $q = 0$ contribute zero. The score is useful when its gate tracks the conditional error trade-off more closely than the best constant mixture. Here, closeness is measured by expected squared distance to $u(c)$, weighted by $q(c)$, so weight assignment matters more where the estimators disagree. This motivates testing both adaptive weights and their assignment to steps.

Action-value variation can favor finer credit. Consider the coarse-to-fine model

$$E = (1-\eta)Y + \xi_E, \qquad S = Y + \xi_S, \qquad 0 \le \eta \le 1, \tag{10}$$

with $\mathbb{E}[Y\xi_E \mid s] = \mathbb{E}[Y\xi_S \mid s] = 0$. The episode-side signal attenuates action-specific variation by η, while the step-side signal retains it. Define $v_E = \mathbb{E}[\xi_E^2 \mid s]$, $v_S = \mathbb{E}[\xi_S^2 \mid s]$, and $\chi = \mathbb{E}[\xi_E\xi_S \mid s]$. The action-value heterogeneity $C(s) = \mathrm{Var}_{U|s} Q^\pi(s,U)$ measures how much the expected outcome changes across available actions. Proposition 2 yields

$$\lambda^\star_{\mathrm{cf}}(C) = \Pi_{[0,1]}\left(\frac{\eta^2 C/\sigma_A^2 + v_E - \chi}{\eta^2 C/\sigma_A^2 + v_E + v_S - 2\chi}\right). \tag{11}$$

For a positive denominator, holding η, the residual moments, and σ_A fixed, this weight is nondecreasing in C when $v_S > \chi$, and strictly increasing in the interior when $\eta > 0$. Greater action-value variation then increases the cost of coarse credit and favors the finer estimator. Appendix B gives a shared-noise interpretation of the residual condition.

Expected NLL links the score to this regime. Let $h(s) = \mathbb{E}[-L^{-1}\log\pi(X \mid s) \mid s]$, where L is the full response length in tokens. Suppose the distribution over parsed environment actions has finite support of size $K_s \ge 2$, $L \ge L_{\min} > 0$, and linguistic variation among responses executing the same action satisfies $H(X \mid U, s) \le B_s$. The entropy chain rule gives $H(U \mid s) \ge L_{\min}h(s) - B_s$. Thus expected NLL exceeding the within-action variation implies uncertainty over distinct environment actions.

For independent actions U, U', define diversity as $D(s) = \Pr(U \ne U' \mid s)$. Proposition 3 constructs a lower bound $\underline{D}_s$ such that

$$D(s) \ge \underline{D}_s(h(s)). \tag{12}$$

For fixed $K_s, B_s, L_{\min}$, this bound is nondecreasing in h and positive when $L_{\min}h > B_s$. If distinct actions also have mean squared value separation at least $\kappa_0 > 0$, then

$$C(s) = \tfrac{1}{2}D(s)\mathbb{E}[(Q^\pi(s,U) - Q^\pi(s,U'))^2 \mid U \ne U', s] \;\ge\; \tfrac{\kappa_0}{2}\underline{D}_s(h(s)). \tag{13}$$

Composing this bound with Eq. 11 gives a nondecreasing lower bound on the preferred step-level weight at fixed regularity and residual-noise parameters. This establishes the conditional route from expected NLL to action diversity, action-value heterogeneity, and finer credit.

GACA follows this direction using the NLL of one sampled response per decision. Trajectory normalization preserves the ordering of NLL within each trajectory, and the gate increases with the normalized score. The ablations below evaluate the resulting allocation and modulation strength. Appendix B gives the exact diversity bound, and Appendix C.6 relates expected NLL to the implemented score.

4 Experiments

Our evaluation measures attained task success on two agent benchmarks, progress during training, and per-update computational cost. Matched ablations examine stepwise credit allocation, followed by diagnostics of the NLL signal.

4.1 Experimental setup

Tasks and training. We use the verl-agent stack (Feng et al., 2026; Sheng et al., 2025) with Qwen2.5-Instruct 1.5B and 7B. ALFWorld involves household tasks with a maximum of 50 steps; WebShop involves product search and purchase within 15 steps. GACA samples groups of 8 trajectories and applies an invalid-action penalty of 0.1. For ALFWorld 1.5B, we use a two-step history window, rollout temperature 1, and 16 task groups per update. We train this model for 300 parameter updates. We report its update-150, best, and final evaluations to distinguish performance at a shorter budget from gains under continued training. Appendix D gives training settings and benchmark details.

Evaluation and checkpoint selection. For ALFWorld, we follow HGPO's evaluation-seed convention (He et al., 2026): each selected checkpoint receives three stochastic evaluations, summarized by their mean and population SD. Each backbone uses one trained model. The 1.5B run uses the in-distribution evaluation split, 128 tasks, and temperature 0.4. It evaluates every 5 updates and selects the update with the highest mean validation success. The selected ALFWorld updates are 275 (1.5B) and 265 (7B). Figure 2 also reports progress at fixed updates. Appendix D.1 gives the seed values, selection rule, and published-reference protocols.

Reference methods and internal controls. The benchmark comparison focuses on attained task success, using selected GACA checkpoints and the strongest published configurations of the listed reference methods. GRPO and GiGPO results are quoted from Feng et al. (2026); HGPO and its accompanying GiGPO references are quoted from He et al. (2026). Table 1 retains the $K = 4$ HGPO configurations, the stronger reported setting in each framework version, and distinguishes their WebShop horizons. The complete $K = 2, 4$ comparison appears in Table 7. Internal ALFWorld 1.5B ablations retain the same advantage definitions, normalization, and singleton fallback under a common training and selection protocol. The fixed-weight controls retain GACA's convex-mixture scale; Appendix E distinguishes this scale from the original GiGPO update.

4.2 Overall Performance

Table 1 reports success rates across four backbone–benchmark settings. GACA has the highest reported mean among the listed methods in each setting, reaching 96.88% and 98.18% on ALFWorld and 80.99% and 83.59% on WebShop at 1.5B and 7B, respectively. Relative to updated-framework HGPO ($K = 4$), the differences are 2.03 and 2.22 percentage points on ALFWorld and 2.87 and 4.30 points on WebShop. The margins over the original GiGPO reference range from 7.4 to 13.6 points.

Table 1: Success rates (%, mean ± SD). External rows are quoted from the indicated sources. HGPO's ALFWorld entries are in-distribution results; K is the history-window length. The complete history-window comparison is in Table 7. WebShop horizons and framework versions are stated per block. GACA's ALFWorld deviations are across three evaluation seeds at one selected checkpoint.

Method	ALFWorld		WebShop	
	1.5B	7B	1.5B	7B
Feng et al. (2026); WebShop: 15 steps				
GRPO	72.8 ±3.6	77.6 ±5.2	56.8 ±3.8	66.1 ±3.7
GiGPO	86.7 ±1.7	90.8 ±1.3	67.4 ±4.5	75.2 ±3.8
He et al. (2026), Table 7: earlier framework; WebShop: 15 steps				
HGPO ($K = 4$)	92.45 ±0.81	94.79 ±0.90	75.91 ±1.19	77.21 ±0.22
He et al. (2026), Table 1: updated framework; WebShop: 30 steps				
GiGPO ($K = 4$)	93.29 ±1.07	95.63 ±0.49	73.24 ±2.25	77.79 ±0.92
HGPO ($K = 4$)	94.85 ±0.92	95.96 ±0.49	78.12 ±2.06	79.29 ±1.17
This work; WebShop: 15 steps				
GACA	**96.88** ±0.64	**98.18** ±0.37	**80.99** ±2.24	**83.59** ±3.38

Comparison with history-aware grouping. HGPO improves the local advantage by grouping steps at different history depths. GACA retains GiGPO's anchor grouping and changes the relative weight of episode and step credit. The two approaches act on different parts of credit assignment: construction of the local estimate and its contribution to the final update. The earlier-framework HGPO rows provide a 15-step WebShop reference, while the updated rows retain the stronger published results with their 30-step horizon. These comparisons preserve each source's training and evaluation protocol; the internal ablations below isolate GACA's allocation choices.

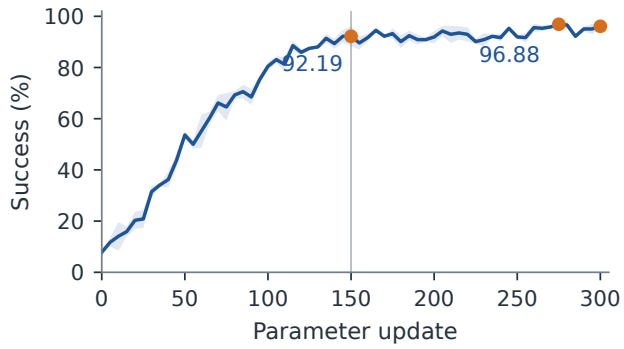


Figure 2: ALFWorld 1.5B validation success over 300 parameter updates. The curve and band show the mean and population SD across three evaluation seeds for one training run. Markers identify update 150, the selected update 275, and the final update 300.

Performance over training. ALFWorld 1.5B reaches $92.19 \pm 3.88\%$ by update 150. Continued training raises success by 4.69 percentage points to $96.88 \pm 0.64\%$ at update 275, with $96.09 \pm 1.10\%$ retained at the final update 300 (Figure 2). This trajectory shows strong performance within the first 150 updates and further gains with continued optimization. Table 3 reports the individual evaluation scores at these updates.

Computational cost. GACA reuses log-probabilities already computed for the policy update and adds scalar NLL, score, and mixing operations. These operations take 0.09 seconds per iteration in the ALFWorld 1.5B timing breakdown, against approximately 372 seconds for the complete iteration. The reported peak memory remains 62.4 GB. These timings average 20 iterations on the same hardware and quantify the incremental cost of the adaptive rule. Appendix F gives the full breakdown.

4.3 Effect of Adaptive Credit Allocation

We use ALFWorld 1.5B to test three aspects of credit allocation: combining episode and step signals, adapting their mixture beyond a constant weight, and assigning the resulting weights to individual decisions. Figure 3 presents the fixed-weight sweep and random assignment control. The historical step-weighted mean is $\bar{\lambda}_{\text{train}} \approx 0.748$, motivating fixed $\lambda = 0.75$ as a near-mean comparator. Appendix G gives the complete numerical grid.

Adaptive versus fixed mixing. With adaptation disabled ($\alpha_\lambda = 0$), we vary the constant weight over $\{0, 0.25, 0.5, 0.75, 1\}$. The episode-only and step-only endpoints reach 85.94% and 84.60%, while every tested interior mixture performs better than both endpoints. Combining the two granularities therefore helps even with a fixed weight. The best tested constant, $\lambda = 0.5$, reaches 93.05%; fixed $\lambda = 0.75$, close to GACA's mean weight, reaches 92.41%. With the same episode- and step-level estimators, GACA reaches 96.88%, exceeding these controls by 3.83 and 4.47 points, respectively. The results support adapting the relative contribution of the two credit signals across decisions.

Does the assignment to steps matter? Randomly permuting criticality scores within each trajectory reduces success from 96.88% to 88.54%. This control retains the trajectory's collection of gate values and changes their pairing with the episode and step advantages. The 8.34-point drop shows that performance depends on where the weights are assigned. Together with the fixed-weight sweep, it supports using the score to allocate the two credit signals at individual decisions.

The realized step weights span 0.623–0.900 at the 10th–90th percentiles in frozen-policy rollouts; Appendix H.1 gives their distribution and the training mean.

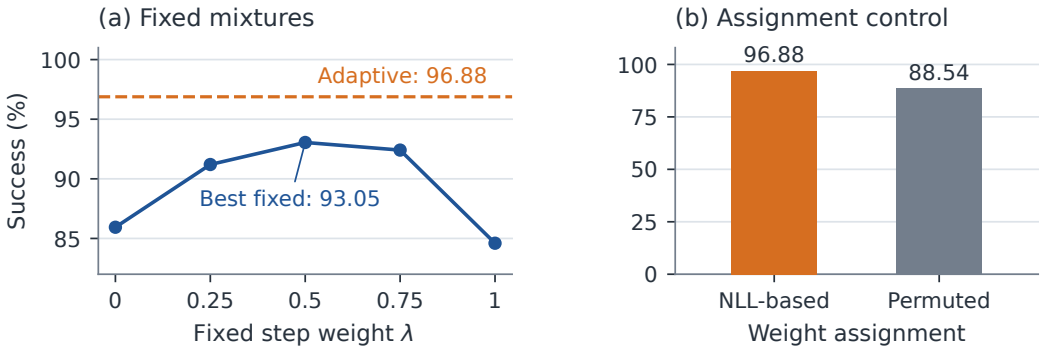


Figure 3: Matched ALFWorld 1.5B ablations, using means over three evaluation seeds. **Left:** adaptive mixing exceeds the best tested fixed weight by 3.83 points. **Right:** permuting NLL scores within each trajectory preserves its weight multiset but reduces success by 8.34 points. All values also appear in Table 9.

Modulation strength. All three tested nonzero modulation strengths have higher mean success than the modulation-off control, with $\alpha_\lambda = 0.5$ giving the highest mean. Appendix G reports the sensitivity results.

4.4 Within-trajectory analysis of the NLL score

We examine whether the score used by GACA identifies decisions with greater action diversity within a trajectory. Using the frozen ALFWorld 1.5B final@300 checkpoint, we select 32 training tasks stratified by task type and collect 8 trajectories per task. From one uniformly sampled trajectory per task, we draw two decisions uniformly without replacement, giving 64 states. Each pair is ordered by its original rollout criticality c_t. At each state, 32 independently sampled responses estimate action disagreement D, the probability that two responses execute different parsed actions.

Higher-score states have mean disagreement 0.1789, compared with 0.1041 for lower-score states (Figure 4). The mean paired difference is 0.0747, with a 95% task-bootstrap interval of $[0.0248, 0.1334]$.

Higher-score decisions thus exhibit greater action diversity on average within the same trajectory. Together, the allocation controls and paired analysis show that the assignment of weights affects training performance and that higher weights are associated with more action-diverse decisions. Appendix H.2 gives the sampling and estimation details.

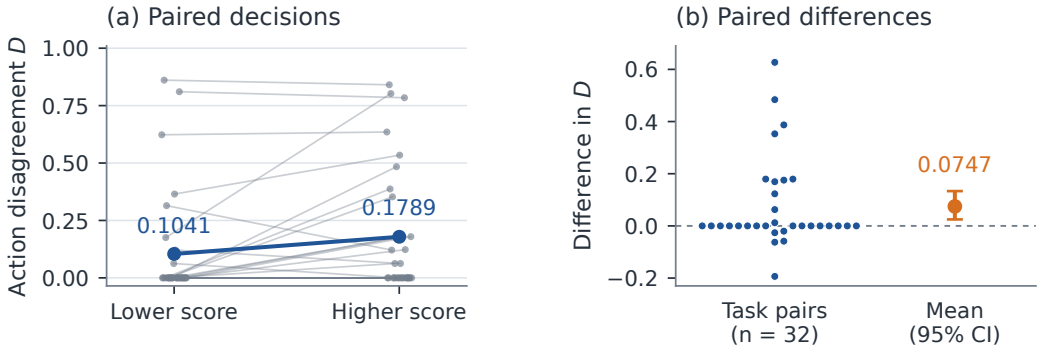


Figure 4: Within-trajectory action diversity for the frozen ALFWorld 1.5B policy. **Left:** paired states ordered by original c_t; blue markers show group means. **Right:** all 32 higher-minus-lower differences (blue), with their mean and 95% task-bootstrap interval (orange). Horizontal offsets separate overlapping observations. Diversity is estimated from fresh responses.

5 Conclusion

GACA uses rollout NLL to adapt the mixture of episode- and step-level credit during policy optimization, without additional training rollouts or model evaluations. Conditional risk analysis characterizes score-dependent mixing and conditions favoring finer credit. Across ALFWorld and WebShop with two backbone sizes, GACA achieves the highest reported mean success rates among the compared methods. In matched ALFWorld ablations, GACA improves over the tested fixed weights and within-trajectory random assignment. Together, these results support step-dependent credit mixing as a practical design choice for critic-free agent training.

AI Use Statement

Generative AI tools assisted with manuscript editing, theoretical exposition, figure preparation, and diagnostic experiment design. The authors are responsible for verifying the arguments, implementation descriptions, references, and reported results.

Reproducibility Statement

Appendix B–C contain the theoretical statements and proofs. Appendix D–E specify training settings, advantage normalization, singleton fallback, and criticality construction. Appendix G gives the full numerical ablation grid, and Appendix H defines the diagnostic estimators. Section 4.1 records the checkpoint-selection protocol and the source of external baseline results.

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.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. *Advances in neural information processing systems*, 30, 2017.

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

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018.

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

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

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

Haipeng Luo, Qingfeng Sun, Songli Wu, Can Xu, Wenfeng Deng, Han Hu, and Yansong Tang. Stare: Surprisal-guided token-level advantage reweighting for policy entropy stability. *arXiv preprint arXiv:2606.19236*, 2026.

Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Icml*, volume 99, pp. 278–287. Citeseer, 1999.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

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

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*, 2020.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Hanyang Wang, Weijieying Ren, Yuxiang Zhang, Ding Cao, Zhizhao Zeng, Ke Zeng, and Tianxiang Zhao. Bipace: Bisimulation-guided policy optimization with action counterfactual estimation for llm agents. *arXiv preprint arXiv:2606.25556*, 2026a.

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

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486, 2026b.

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

Kaibing Yang, Guangfeng Cai, Shengtian Yang, Shuo He, Yu Li, Mengyi Liu, Pengwei Chen, Jun Xu, and Lei Feng. Progress-conditioned group policy optimization for long-horizon agentic tasks. *arXiv preprint arXiv:2607.22724*, 2026.

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

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Haotian Zhao, Songlin Zhou, Yuxin Zhang, Stephen S-T Yau, Wenyu Zhang, Lun Tian, Tianshu Zhu, Yifeng Huang, Yucheng Zeng, Jingnan Gu, et al. Aem: Adaptive entropy modulation for multi-turn agentic reinforcement learning. *arXiv preprint arXiv:2605.00425*, 2026.

A Extended Related Work

Critic-free RL for LLMs and agents. Replacing the learned value function with a group baseline has become a common recipe for RL on language models, from GRPO (Shao et al., 2024; Guo et al., 2025) and REINFORCE-style variants (Ahmadian et al., 2024) to refinements of clipping and normalization (Yu et al., 2026). Carrying this from single-turn reasoning to multi-turn agents required new infrastructure and recipes (Sheng et al., 2025; Wang et al., 2025). GiGPO (Feng et al., 2026) is the direct methodological precursor from which our reported implementation inherits anchor-observation step grouping. Subsequent work has exposed limitations of observation-only step groups: HGPO uses hierarchical history-consistent groups to address context inconsistency (He et al., 2026), while BiPACE replaces surface observation hashing with behavior-aware grouping and action-conditioned peer baselines (Wang et al., 2026a). ProGPO addresses a complementary long-horizon credit trap by introducing progress-conditioned relative advantages in failure-dominated groups (Yang et al., 2026). These methods modify the construction of local credit estimates. GACA addresses a complementary choice: the relative weight assigned to local and episode-level credit at each step. The mixing analysis accommodates alternative step estimators; the reported experiments use GiGPO-style anchor grouping.

Credit assignment and reward redistribution. Assigning delayed outcomes to individual decisions is a central problem in reinforcement learning. Approaches include return decomposition (Arjona-Medina et al., 2019), hindsight relabeling (Andrychowicz et al., 2017), generalized advantage estimation (Schulman et al., 2015), and potential-based shaping (Ng et al., 1999). For LLMs, process reward models supply dense step-level supervision where intermediate correctness is checkable (Lightman et al., 2024; Wang et al., 2024; Uesato et al., 2022), and VinePPO replaces the critic with Monte-Carlo value estimates from extra rollouts branched at intermediate states (Kazemnejad et al., 2024). These approaches use additional process supervision or intermediate-state rollouts. GACA reuses rollout log-probabilities to condition the mixture of existing credit estimates. GAE trades bias and variance through a mixture of temporal estimates; GACA instead mixes two critic-free group-relative estimates using a per-step score. Proposition 1 characterizes score-dependent mixing relative to local credit, while Proposition 2 isolates a regime in which finer credit becomes increasingly valuable as local action values separate.

Uncertainty-guided advantage modulation. Recent work also uses surprisal to reshape RL updates. AEM (Zhao et al., 2026) analyzes response-level entropy dynamics in multi-turn agents and rescales a base response-level advantage using a response-level uncertainty proxy. STARE (Luo et al., 2026) identifies high-surprisal tokens and selectively reweights their effective advantages to stabilize policy entropy. These methods rescale a pre-existing credit signal. GACA uses the score to construct a mixture of episode- and step-level credit estimates.

Entropy and the value of uncertain decisions. Entropy has long shaped exploration and stability in RL (Haarnoja et al., 2018), and recent analyses of RL for LLM reasoning show that a small, high-entropy minority of tokens drives most of the policy improvement while entropy collapse degrades learning (Wang et al., 2026b; Cui et al., 2025). GACA uses uncertainty at the environment-step level to decide how much to rely on the two credit granularities.

B Detailed Theory Statements

This appendix gives the full formulations underlying the three results summarized in Section 3.3. Throughout, $\pi = \pi_{\theta_{\text{old}}}$ is fixed.

Local-credit target and conditional mixing. Let X denote the full sampled response, including reasoning and the action command, and let $U = \phi(X)$ be the corresponding environment action after parsing. At a full pre-action state s, define

$$Q^\pi(s,u) = \mathbb{E}[\hat{g}_t \mid s_t = s, U_t = u], \qquad V^\pi(s) = \mathbb{E}_{U\sim\pi(\cdot|s)}[Q^\pi(s,U)],$$

and the action advantage $A^\pi(s,u) = Q^\pi(s,u) - V^\pi(s)$. Let the old-policy step-row population be the reference measure and assume $0 < \sigma_A^2 < \infty$ and finite second moments for E and S. Define

$$\sigma_A^2 = \mathbb{E}\big[(A^\pi(s_t,U_t))^2\big] > 0,$$

and use the dimensionless local-credit target

$$Y_t = \frac{A^\pi(s_t, U_t)}{\sigma_A}. \tag{14}$$

The target scale is fixed by the old-policy population. Differences between this normalization and the implemented group normalizations enter the estimator error. Let $E = A^{\mathrm{E}}$ and $S = A^{\mathrm{S}}$ denote the two *implemented* inputs to equation 7, after their actual group normalization and any singleton fallback. Normalization mismatch, grouping bias, and shared rollout noise are included in

$$\begin{gathered} e_E = E - Y, \qquad e_S = S - Y, \\ M_E(c) = \mathbb{E}[e_E^2 \mid c], \qquad M_S(c) = \mathbb{E}[e_S^2 \mid c], \qquad \chi(c) = \mathbb{E}[e_E e_S \mid c], \\ q(c) = M_E(c) + M_S(c) - 2\chi(c) = \mathbb{E}[(S-E)^2 \mid c]. \end{gathered} \tag{15}$$

The score c may be statistically dependent on the sampled action, the trajectory, and both estimator errors.

Proposition 1 (Optimal conditional mixing relative to local credit)**.** *For $\lambda \in [0,1]$, the conditional squared-credit risk of $(1-\lambda)E + \lambda S$ is*

$$R_c(\lambda) = (1-\lambda)^2 M_E(c) + \lambda^2 M_S(c) + 2\lambda(1-\lambda)\chi(c). \tag{16}$$

On $\{q(c) > 0\}$, define

$$u(c) = \frac{M_E(c) - \chi(c)}{q(c)}.$$

The risk-minimizing score-dependent weight is

$$f^\star(c) = \Pi_{[0,1]}\big(u(c)\big). \tag{17}$$

If $\mathbb{E}[q(c)] > 0$, the best constant weight is

$$\lambda^\star_{\mathrm{fix}} = \Pi_{[0,1]}\left(\frac{\mathbb{E}[M_E(c) - \chi(c)]}{\mathbb{E}[q(c)]}\right). \tag{18}$$

Moreover, every measurable gate $f(c) \in [0,1]$ satisfies

$$\mathcal{R}(f) - \mathcal{R}(\lambda^\star_{\mathrm{fix}}) = \mathbb{E}\big[q(c)\left([f(c) - u(c)]^2 - [\lambda^\star_{\mathrm{fix}} - u(c)]^2\right)\big], \tag{19}$$

where $\mathcal{R}(f) = \mathbb{E}[((1-f(c))E + f(c)S - Y)^2]$. If $q(c) = 0$, then $S = E$ conditionally almost surely and the weight is irrelevant. The oracle satisfies $\mathcal{R}(f^\star) \le \min_{\lambda\in[0,1]} \mathcal{R}(\lambda)$. For $\mathbb{E}[q] > 0$, the inequality is strict exactly when no constant agrees with $f^\star(c)$ almost surely on $\{q(c) > 0\}$.

Why local action-value heterogeneity can favor finer credit. To connect the generic risk to credit granularity, consider the following explanatory special case. At a fixed full state s, let the coarse episode-side signal attenuate action-specific credit while the fine step-side signal retain it:

$$E = (1-\eta)Y + \xi_E, \qquad S = Y + \xi_S, \qquad 0 \le \eta \le 1. \tag{20}$$

Assume

$$\mathbb{E}[Y\xi_E \mid s] = \mathbb{E}[Y\xi_S \mid s] = 0,$$

and write

$$v_E = \mathbb{E}[\xi_E^2 \mid s], \qquad v_S = \mathbb{E}[\xi_S^2 \mid s], \qquad \chi = \mathbb{E}[\xi_E\xi_S \mid s].$$

Because $U \sim \pi(\cdot \mid s)$,

$$\mathbb{E}[Y \mid s] = 0, \qquad \mathbb{E}[Y^2 \mid s] = \frac{C(s)}{\sigma_A^2}, \qquad C(s) := \mathrm{Var}_{U\sim\pi(\cdot|s)}[Q^\pi(s,U)]. \tag{21}$$

Thus $C(s)$ measures how much the expected continuation value changes across actions available at the same full state.

Proposition 2 (Coarse-to-fine comparative statics)**.** *Under equation 20, the unprojected risk-minimizing fine-credit weight is*

$$u_{\mathrm{cf}}(C) = \frac{\eta^2 C/\sigma_A^2 + v_E - \chi}{\eta^2 C/\sigma_A^2 + v_E + v_S - 2\chi}, \tag{22}$$

whenever the denominator is positive, and the constrained optimum is $\lambda^\star_{\mathrm{cf}} = \Pi_{[0,1]}(u_{\mathrm{cf}})$*. Holding* η, v_E, v_S, χ*, and* σ_A *fixed,*

$$\frac{\partial u_{\mathrm{cf}}}{\partial C} = \frac{\eta^2 (v_S - \chi)}{\sigma_A^2 \big(\eta^2 C/\sigma_A^2 + v_E + v_S - 2\chi\big)^2}. \tag{23}$$

Hence, if $v_S > \chi$*, the preferred fine-credit weight is nondecreasing in local action-value heterogeneity* C*; it is strictly increasing in the interior whenever* $\eta > 0$*.*

The condition $v_S > \chi$ has a simple shared-noise interpretation. If

$$\xi_E = \zeta + \varepsilon_E, \qquad \xi_S = \zeta + \varepsilon_S,$$

with conditionally zero-mean components, the shared component ζ uncorrelated with the two residuals, and the residuals mutually uncorrelated, then

$$v_S - \chi = \mathrm{Var}(\varepsilon_S) \geq 0,$$

with strict inequality whenever the fine estimator has nonzero idiosyncratic residual noise. Proposition 2 describes variation in C within this model at fixed residual moments. Applying it across states also requires accounting for changes in those moments.

Expected NLL as a structural certificate. We next relate expected response surprisal to variation in the executed action. Condition on the *full* pre-action state s, including the interaction history. Let $X \sim \pi(\cdot \mid s)$ be the full response of token length L, and let $U = \phi(X)$ be its environment-executed parsed action identity. Define

$$\ell = -L^{-1} \log \pi(X \mid s), \qquad h(s) = \mathbb{E}[\ell \mid s]. \tag{24}$$

Let the canonical action distribution at s have finite support size $K_s \geq 2$, let $L \geq L_{\min} > 0$, and suppose the within-action textual entropy satisfies

$$H(X \mid U, s) \leq B_s.$$

The finite-support requirement is natural under a finite tokenizer vocabulary and bounded response length; the substantive quantity is B_s, which separates environment-action uncertainty from linguistic variation among texts that execute the same action.

Define

$$\begin{aligned} \underline{H}_s(h) &= \min\{[L_{\min} h - B_s]_+, \log K_s\}, \\ \underline{D}_s(h) &= 1 - F_{K_s}^{-1}\big(\underline{H}_s(h)\big), \\ F_K(p) = h_b(p) + (1-p)\log(K-1), &\qquad h_b(p) = -p\log p - (1-p)\log(1-p). \end{aligned} \tag{25}$$

Proposition 3 (Expected NLL certifies canonical-action diversity)**.** *Let* U, U' *be independent draws from the induced canonical-action law at* s*, and define*

$$D(s) = \Pr(U \neq U' \mid s).$$

Then

$$D(s) \geq \underline{D}_s(h(s)). \tag{26}$$

For fixed K_s, B_s*, and* $L_{\min}$*, the lower bound is nondecreasing in expected NLL* $h(s)$ *and is positive whenever* $L_{\min} h(s) > B_s$*.*

Corollary 4 (From action diversity to local value heterogeneity)**.** *Suppose in addition that, whenever* $D(s) > 0$*,*

$$\kappa_s := \mathbb{E}\Big[\big(Q^\pi(s,U) - Q^\pi(s,U')\big)^2 \mid U \neq U', s\Big] \geq \kappa_0 > 0. \tag{27}$$

Then

$$C(s) \geq \underline{C}_s(h(s)) := \frac{\kappa_0}{2}\, \underline{D}_s(h(s)). \tag{28}$$

The two conditions serve distinct roles: B_s bounds linguistic variation among responses executing the same action, while κ_0 links distinct actions to differences in expected outcome.

From NLL to a lower envelope on preferred granularity. Combining the two preceding pieces yields the population-level directional link used to motivate GACA.

Corollary 5 (NLL-induced lower envelope on fine-credit preference)**.** *Consider the coarse-to-fine family of Proposition 2 with fixed η, v_E, v_S, χ, and σ_A, and suppose $v_S > \chi$ and equation 27 holds. Define*

$$\underline{\lambda}(h) = \Pi_{[0,1]}\left(\frac{\eta^2 \underline{C}_s(h)/\sigma_A^2 + v_E - \chi}{\eta^2 \underline{C}_s(h)/\sigma_A^2 + v_E + v_S - 2\chi}\right). \tag{29}$$

Then

$$\lambda^\star_{\mathrm{cf}}(s) \geq \underline{\lambda}(h(s)), \tag{30}$$

and $\underline{\lambda}(h)$ is nondecreasing in h for fixed regularity and residual-noise parameters.

Alignment with the implemented trajectory-relative gate. GACA uses one sampled-response NLL per decision and sets $c_t = H_t/\overline{H}_\tau$ after rollout completion. For $\overline{H}_\tau > 0$, normalization preserves NLL ordering within the trajectory, and the gate in equation 6 assigns at least as much step-level weight to higher-scoring responses. The population analysis motivates this increasing direction; the gate floor and modulation strength are empirical design choices.

C Proofs and Technical Remarks

All results are fixed-policy statements under $\pi = \pi_{\theta_{\mathrm{old}}}$. The implemented inputs $E = A^{\mathrm{E}}$ and $S = A^{\mathrm{S}}$ may share rollout randomness, may be biased relative to the local-credit target, and may use random group normalizations; no independence between E, S, and the gate score is assumed in Proposition 1. Entropies use natural logarithms.

C.1 Proof of Proposition 1

Expanding equation 16,

$$R_c(\lambda) = M_E(c) - 2\lambda[M_E(c) - \chi(c)] + \lambda^2 q(c).$$

For $q(c) > 0$, completing the square gives

$$R_c(\lambda) = M_E(c) - \frac{[M_E(c) - \chi(c)]^2}{q(c)} + q(c)[\lambda - u(c)]^2.$$

Projection onto $[0, 1]$ therefore gives equation 17. If $q(c) = 0$, the identity $q(c) = \mathbb{E}[(S - E)^2 \mid c]$ implies $S = E$ conditionally almost surely. Taking expectation of the quadratic before minimizing gives equation 18. Subtracting the completed-square forms gives equation 19. Conditional Cauchy–Schwarz implies $[M_E - \chi]^2 \leq qM_E$, so the weighted square terms are integrable whenever the stated second moments are finite.

Finally, projection minimizes the conditional quadratic at every score. On $\{q(c) > 0\}$ the minimizer is unique; hence equality with a constant requires that constant to agree with $f^\star(c)$ almost surely there. Contributions from $\{q(c) = 0\}$ vanish. □

C.2 Proof of Proposition 2

Under equation 20,

$$e_E = -\eta Y + \xi_E, \qquad e_S = \xi_S.$$

The orthogonality conditions and equation 21 give

$$M_E(s) = \eta^2 \frac{C(s)}{\sigma_A^2} + v_E, \qquad M_S(s) = v_S, \qquad \chi(s) = \chi.$$

Substitution into the unprojected optimum $u = (M_E - \chi)/(M_E + M_S - 2\chi)$ yields equation 22. Differentiating with respect to C while holding η, v_E, v_S, χ, and σ_A fixed gives equation 23. Projection onto a closed interval is monotone, so the constrained optimum inherits the nondecreasing dependence on C. Strict increase holds wherever the unprojected optimum lies in the interior and $\eta > 0$, $v_S > \chi$. □

Shared-noise sufficient condition. If $\xi_E = \zeta + \varepsilon_E$ and $\xi_S = \zeta + \varepsilon_S$ with conditionally zero-mean components and all pairwise cross-covariances among $\zeta, \varepsilon_E, \varepsilon_S$ equal to zero, then

$$v_S = \mathrm{Var}(\zeta) + \mathrm{Var}(\varepsilon_S), \qquad \chi = \mathrm{Var}(\zeta),$$

and therefore

$$v_S - \chi = \mathrm{Var}(\varepsilon_S) \geq 0.$$

This shared-noise structure satisfies the second-moment condition in Proposition 2.

C.3 Proof of Proposition 3

Condition on a full pre-action state s. The textual action X follows the logged old policy, so

$$H(X \mid s) = \mathbb{E}[-\log \pi(X \mid s) \mid s] = \mathbb{E}[L\ell \mid s].$$

Because $L \geq L_{\min}$ and $\ell \geq 0$,

$$H(X \mid s) \geq L_{\min} h(s).$$

Since $U = \phi(X)$ is a deterministic function of X,

$$H(X \mid s) = H(U \mid s) + H(X \mid U, s),$$

and the bound $H(X \mid U, s) \leq B_s$ gives

$$H(U \mid s) \geq L_{\min} h(s) - B_s. \tag{31}$$

Truncating to the feasible range $[0, \log K_s]$ yields $H(U \mid s) \geq \underline{H}_s(h(s))$.

Let $p_s^{\max} = \max_u p_s(u)$. For two independent canonical actions,

$$D(s) = 1 - \sum_u p_s(u)^2 \geq 1 - p_s^{\max}.$$

Among distributions on K_s symbols with largest mass $p \geq 1/K_s$, entropy is maximized by assigning the remaining mass uniformly, giving

$$H(U \mid s) \leq F_{K_s}(p_s^{\max}),$$

where F_K is defined in equation 25. Since $F_K'(p) = \log \frac{1-p}{p(K-1)} < 0$ for $p > 1/K$, its inverse is decreasing on $[0, \log K]$. Hence

$$p_s^{\max} \leq F_{K_s}^{-1}\big(H(U \mid s)\big) \leq F_{K_s}^{-1}\big(\underline{H}_s(h(s))\big),$$

which proves

$$D(s) \geq \underline{D}_s(h(s)).$$

Monotonicity in h follows because $\underline{H}_s(h)$ is nondecreasing and $F_{K_s}^{-1}$ is decreasing. The lower bound is positive exactly when $L_{\min} h(s) > B_s$. □

C.4 Proof of Corollary 4

For independent U, U' from the canonical-action law at s,

$$\mathbb{E}[(Q^\pi(s,U) - Q^\pi(s,U'))^2 \mid s] = 2\mathrm{Var}_U[Q^\pi(s,U) \mid s] = 2C(s).$$

The squared difference is zero on $U = U'$, so when $D(s) > 0$,

$$2C(s) = D(s)\,\mathbb{E}[(Q^\pi(s,U) - Q^\pi(s,U'))^2 \mid U \neq U', s] = D(s)\kappa_s.$$

Using $\kappa_s \geq \kappa_0$ and Proposition 3,

$$C(s) \geq \frac{\kappa_0}{2} D(s) \geq \frac{\kappa_0}{2} \underline{D}_s(h(s)) = \underline{C}_s(h(s)).$$

If $D(s) = 0$, both sides are zero. □

C.5 Proof of Corollary 5

Under the fixed coarse-to-fine parameters in Corollary 5, Proposition 2 shows that the projected optimum is a nondecreasing function of C when $v_S > \chi$. Corollary 4 gives $C(s) \geq \underline{C}_s(h(s))$. Applying the nondecreasing mapping to both sides yields equation 30. The lower envelope is nondecreasing in h because $\underline{C}_s(h)$ is nondecreasing and the coarse-to-fine optimum is nondecreasing in its C argument. □

C.6 The implemented NLL score

The population analysis uses $h(s) = \mathbb{E}[H_t \mid s]$, the expected mean per-token NLL of the full response. GACA uses one sampled H_t at each decision and computes $c_t = H_t/\overline{H}_\tau$ from the completed rollout. The score can therefore depend on the sampled action and subsequent continuation. During policy optimization, scores and advantages are treated as fixed quantities, with no gradients propagated through them. Proposition 1 allows their statistical dependence.

D Training Configuration and Hyperparameters

Table 2 describes the two benchmarks. Table 4 lists training settings; the archived ALFWorld 1.5B configuration sets 300 epochs and its log records 300 parameter updates. Table 5 gives gate defaults and search ranges. Appendix E specifies advantage normalization and singleton handling. Appendix G reports the parameter sensitivity results.

Table 2: Benchmarks used in our evaluation. Both use the agent-environment interface of Section 2.1 with a sparse, episode-level success reward and a small invalid-action penalty.

Benchmark	**Setting**	**Max steps**	**Eval tasks**	**Metric**
ALFWorld	text-based embodied household tasks	50	128	success rate
WebShop	grounded web shopping and navigation	15	128	success rate

Table 3: ALFWorld validation success (%) at fixed and selected parameter updates. Evaluation columns enumerate the three sampling seeds at the same checkpoint; the summary uses their population standard deviation. Updates 275 (1.5B) and 265 (7B) are selected by validation mean.

Backbone	Update	Eval. 1	Eval. 2	Eval. 3	Mean ± SD
1.5B	150	89.844	89.063	97.656	92.19 ± 3.88
1.5B	275	96.875	96.094	97.656	96.88 ± 0.64
1.5B	300	95.313	95.313	97.656	96.09 ± 1.10
7B	265	98.44	97.66	98.44	98.18 ± 0.37

Table 4: Training settings. The archived epoch count is available for ALFWorld 1.5B. NR denotes a setting not specified in the archived configuration. The frozen-policy diagnostics use success reward 10. Section 4.1 describes evaluation and published references.

Setting	**ALFWorld**	**WebShop**
Backbone	Qwen2.5-Instruct, 1.5B and 7B	
Group size G	8	8
Configured epochs	300 (1.5B)	150
Learning rate	1×10^{-6}	1×10^{-6}
KL loss (low-variance estimator)	0.01	0.01
Discount γ	0.95	0.95
Max environment steps	50	15
Max prompt length	2048	4096
Advantage normalization	mean-and-std-norm	mean-norm
Invalid-action penalty magnitude	0.1	0.1
Terminal success reward (training)	NR	1

Table 5: GACA default hyperparameters and search ranges.

Parameter	**Default**	**Search range**
criticality signal	normalized NLL	NLL and within-trajectory random permutation
fusion weight α_c	1.0	$\{1.0, 0.7, 0.5\}$
step-weight floor λ_0	0.5	$\{0, 0.25, 0.5, 0.75, 1.0\}$; fixed-weight sweep uses $\alpha_\lambda = 0$
modulation α_λ	0.5	$\{0, 0.25, 0.5, 1.0\}$; $\alpha_\lambda = 0$ turns adaptation off
advantage normalization	per-benchmark	{mean-norm, mean-and-std-norm}

D.1 Evaluation protocol and published references

Table 6 records the available evaluation settings. For the ALFWorld 1.5B run, each evaluated update supplies three success rates $s_{u,j}$, and the selected update is $u^\star = \arg\max_u \frac{1}{3}\sum_{j=1}^{3} s_{u,j}$. The reported mean and population SD use the three scores at that same update. The final-update result uses $u = 300$ independently of this maximization. This seed aggregation follows the testing convention stated by HGPO; the checkpoint selection described here is GACA's own protocol.

Table 6: Evaluation and selection settings for the archived GACA ALFWorld 1.5B run.

Setting	GACA, ALFWorld 1.5B
Environment evaluation split	`eval_in_distribution`
Evaluation batch size	128 tasks
Response sampling	temperature 0.4, top-$p = 1$, top-k disabled
Evaluation seeds	42, 123, 2024
Evaluation frequency	every 5 parameter updates
Checkpoint selection	maximum mean validation success over three seeds
Reported best / final update	275 / 300
Training runs underlying SD	one; SD is across evaluation seeds
History / episode limit	2 interactions / 50 environment steps

The original GiGPO study uses 150 updates for ALFWorld and WebShop. HGPO reports 160 training iterations; its updated-framework setting uses a 30-step WebShop horizon, and its earlier-framework results use 15 steps. Table 1 retains representative configurations from both versions, and Table 7 gives the full published-reference comparison, including both history-window lengths. Source results are carried over with their original mean and SD summaries. GACA's historical training log supplies the 150-update and final-300 evaluations in addition to the selected best result.

Table 7: Full success-rate comparison (%, mean ± SD), including all published reference configurations used in this work. External rows are quoted from their sources; K denotes the history-window length. HGPO's ALFWorld entries use its in-distribution setting. WebShop horizons and framework versions are indicated separately. GACA's ALFWorld deviations summarize three evaluation seeds at one selected checkpoint. These rows retain their source protocols.

Method	ALFWorld		WebShop	
	1.5B	7B	1.5B	7B
Feng et al. (2026); WebShop: 15 steps				
GRPO	72.8 ±3.6	77.6 ±5.2	56.8 ±3.8	66.1 ±3.7
GiGPO	86.7 ±1.7	90.8 ±1.3	67.4 ±4.5	75.2 ±3.8
He et al. (2026), Table 7: earlier framework; WebShop: 15 steps				
HGPO ($K = 2$)	89.58 ±0.45	91.15 ±1.19	72.66 ±1.78	76.43 ±1.47
HGPO ($K = 4$)	92.45 ±0.81	94.79 ±0.90	75.91 ±1.19	77.21 ±0.22
He et al. (2026), Table 1: updated framework; WebShop: 30 steps				
GiGPO ($K = 2$)	90.16 ±0.22	93.29 ±0.40	66.53 ±1.92	77.60 ±1.68
HGPO ($K = 2$)	92.77 ±1.08	95.44 ±0.62	71.54 ±4.00	78.51 ±1.40
GiGPO ($K = 4$)	93.29 ±1.07	95.63 ±0.49	73.24 ±2.25	77.79 ±0.92
HGPO ($K = 4$)	94.85 ±0.92	95.96 ±0.49	78.12 ±2.06	79.29 ±1.17
This work; WebShop: 15 steps				
GACA	**96.88** ±0.64	**98.18** ±0.37	**80.99** ±2.24	**83.59** ±3.38

E Implementation Details

E100 execution. The configured score mode is named `entropy`, but its statistic is the sampled response's mean-token NLL in equation 4. The retention parameter `criticality_top_p=1.0` assigns every step to the high branch. Each step therefore forms its own segment, segment aggregation is the identity, and the stored $\beta_\lambda = 0.5$ is unused. Equation 6 is the resulting gate. The equivalent centered parameterization is given below.

Normalization conventions. Group statistics are computed over step rows. In the reconstructed ALFWorld 1.5B computation, each episode-side row carries $\widetilde{R}_t = R(\tau) - 0.1\,\mathbf{1}\{\text{invalid action at } t\}$. For $N_g = \sum_{\tau \in g} T_\tau$, the statistics are $\mu_g = N_g^{-1} \sum_{\tau \in g} \sum_t \widetilde{R}_t$ and $\sigma_g^2 = (N_g - 1)^{-1} \sum_{\tau \in g} \sum_t (\widetilde{R}_t - \mu_g)^2$. Thus longer trajectories contribute more rows, and a local validity penalty can change the episode-side credit at the affected step. Discounted step returns are computed from environment rewards. Their anchor-group sample standard deviation uses divisor $|\mathcal{G}| - 1$ for non-singleton groups. These conventions are retained in the implemented estimator inputs of Section 3.3. The step advantage enters the mixture without additional multiplicative rescaling.

Loss reduction and regularization. The archived ALFWorld 1.5B configuration uses `token-mean`: each valid response token has equal weight across the optimization batch. In equation 1, this corresponds to response weights $w_t = |a_t| / \sum_k |a_k|$ inside $\mathbb{E}_t$. The KL coefficient is 0.01, using `low_var_kl`, and the full objective includes $-0.001\widehat{\mathcal{H}}$, where $\widehat{\mathcal{H}}$ is mean policy entropy over valid token contexts. This entropy regularizer is distinct from the sampled-response NLL score H_t.

Singleton step-group fallback. A singleton anchor group admits no within-group contrast. We normalize that step's discounted return against the statistics of its *prompt group*: $A_t^{\mathrm{S}} = \hat{g}_t - \mu_{\mathrm{prompt}}$ in mean-norm mode and $(\hat{g}_t - \mu_{\mathrm{prompt}})/(\sigma_{\mathrm{prompt}} + \varepsilon)$ in mean-and-std-norm mode, with the prompt-group statistics taken over all steps sharing the prompt index. This provides a group-relative reference for states observed only once, following the normalization conventions of equations 2–3.

Per-step advantage scalar. Every quantity entering equation 7 is a per-step scalar: A_t^{S} and c_t are computed per step, and the mixed A_t^{GACA} is broadcast back to the tokens of a_t through the response mask. Where an advantage has already been broadcast, its per-step scalar is recovered by averaging over the step's valid response tokens.

Ablation controls and centered gate. In the fixed-weight sweep, $\alpha_\lambda = 0$ gives $\lambda_t = \lambda_0$ and leaves the advantage definitions and singleton fallback unchanged. The modulation-off result at $\lambda_0 = 0.5$ is the same configuration as the 0.5 fixed sweep. With $\lambda_{\mathrm{ref}} = \lambda_0(1 + \alpha_\lambda)$ and $\kappa_\lambda = \lambda_0 \alpha_\lambda$, the gate can be written $\lambda_t = \min\{\lambda_{\mathrm{ref}} + \kappa_\lambda(c_t - 1), 1\}$. For $c_t \equiv 1$, the default gate has constant weight 0.75. The observed step-weighted training mean is denoted separately by $\bar{\lambda}_{\mathrm{train}}$. The random control permutes NLL scores within each trajectory, preserving its weight multiset while changing their pairing with the two advantage signals. The original GiGPO rule $A^{\mathrm{E}} + A^{\mathrm{S}}$ also has a different total advantage scale from the fixed $\lambda = 0.5$ GACA mixture.

F Computational Cost

GACA adds no parameters, no rollouts and no forward passes: the per-token log-probabilities that equation 4 averages are already materialized for the importance ratio in equation 1, and everything downstream of them is elementwise arithmetic on per-step scalars. Table 8 confirms this on the setting where the horizon is longest. The two components GACA adds over GiGPO cost 0.09 s per iteration together, against an iteration dominated by rollout and policy update, giving a reported overhead below 0.2% over GRPO in this component-level timing comparison.

Table 8: Per-iteration wall-clock breakdown on ALFWorld with Qwen2.5-1.5B-Instruct, averaged over 20 training iterations on the same hardware. Rows above the rule are shared by all three methods; rows below are the additions each method makes to the advantage computation. The reported peak memory is the same for all three configurations.

Component (s / iteration)	**GRPO**	**GiGPO**	**GACA**
Environment rollout		318.6	
Old- and reference-policy log-probabilities		23.1	
Clipped policy update		29.5	
Anchor-state grouping (hash lookups)	—	0.01	0.01
Step-level advantage	—	0.53	0.53
sampled NLL and surprisal score	—	—	0.07
Mixing weight and advantage blend	—	—	0.02
Total	371.2	371.7	371.8
Overhead vs. GRPO	—	+0.15%	+0.17%
Peak GPU memory (GB)	62.4	62.4	62.4

G Complete Ablation Results

Table 9 gives the complete ALFWorld 1.5B ablation grid underlying the main allocation comparison and the additional sensitivity analyses below. The two fixed-0.5 entries are one configuration and one result.

Table 9: Ablation grid on ALFWorld (1.5B): each block varies one design choice with the remaining components at their defaults, under a single training and selection protocol and averaged over the same three evaluation seeds. Best per block in **bold**; the full default configuration is shaded. The two entries marked † are the same configuration: $\alpha_\lambda = 0$ and $\lambda_0 = 0.5$ give a fixed equally weighted mixture.

Variant	Success rate	Δ vs. default
Criticality score. *Does score-to-step assignment matter?*		
NLL proxy *(default)*	**96.88**	*reference*
random permutation	88.54	−8.34
Fixed mixing weight λ_0, adaptation off. *Which tested constant is best?*		
0.0 (episode-level only)	85.94	−10.94
0.25	91.20	−5.68
0.5†	**93.05**	−3.83
0.75	92.41	−4.47
1.0 (step-level only)	84.60	−12.28
Modulation strength α_λ. *Which strength is best?*		
0 (no modulation)†	93.05	−3.83
0.25	95.12	−1.76
0.5 *(default)*	**96.88**	*reference*
1.0	95.44	−1.44
Fusion weight α_c. *Does return change help?*		
1.0 *(default)*	**96.88**	*reference*
0.7	96.05	−0.83
0.5	95.20	−1.68

G.1 Modulation strength and the mixing law

At $\lambda_0 = 0.5$, the tested strengths $\alpha_\lambda = 0, 0.25, 0.5, 1$ give success rates of 93.05%, 95.12%, 96.88%, and 95.44%. All three nonzero strengths have higher mean success than the fixed mixture, with the default 0.5 giving the largest mean. Increasing α_λ raises the weight at a given positive score and lowers the criticality threshold for saturation. The sweep therefore varies the mean weight together with sensitivity to criticality and the extent of clipping.

G.2 Choice of criticality signal

We also test adding normalized changes in discounted return to the NLL score, keeping the gate parameters fixed.

NLL–return-change fusion. In the measured non-default control, let $\hat{e}_t = H_t/\overline{H}$ and $\hat{d}_t = \Delta_t/\overline{\Delta}$ with $\Delta_t = |\hat{g}_t - \hat{g}_{t-1}|$ and $\Delta_1 = 0$. When $\overline{\Delta} > 0$, the optional fusion is

$$\text{score}_t = \alpha_c \hat{e}_t + (1 - \alpha_c)\hat{d}_t, \qquad \alpha_c \in [0, 1]. \tag{32}$$

The fused score is normalized within the completed trajectory before gating; $\alpha_c = 1$ gives the main NLL-only method. The archived configuration does not specify the handling of $\overline{\Delta} = 0$.

Pure NLL ($\alpha_c = 1$) reaches 96.88%, compared with 96.05% and 95.20% for NLL shares 0.7 and 0.5, respectively (Table 9), supporting the NLL-only choice among these tested constructions.

H SUPPLEMENTARY DIAGNOSTICS

The diagnostics use the frozen ALFWorld 1.5B final@300 checkpoint with temperature 1, top-$p = 1$, top-k disabled, and at most 512 response tokens. The environment uses success reward 10, discount 0.95, a 50-step episode limit, and a two-step history window. The weight-distribution sample uses 16 fixed training tasks. The paired diagnostic uses 32 tasks stratified by training-task type, fixed before sampling with master seed 2026092602.

H.1 TRAINING MEAN AND FROZEN-POLICY WEIGHTS

Across 300 parameter updates, the logged mixing weights have a step-weighted mean of approximately 0.748; each update's mean is weighted by its number of environment steps. Trajectory-normalized scores have mean one, giving the default gate a pre-cap mean of 0.75, with clipping slightly reducing it. This motivates the fixed-0.75 training control in Section 4.3.

Stepwise weight distributions use newly collected complete rollout groups from the frozen final@300 checkpoint. Each of 16 fixed training tasks supplies 8 trajectories; 122 of the 128 trajectories succeed. The sample contains 1,269 environment steps, with non-singleton anchor groups covering 96.30% of the steps.

The sampled weight mean is 0.7484, its median is 0.7390, and its 10th–90th percentiles are 0.6228–0.8996; 2.68% of steps reach $\lambda_t = 1$ (Figure 5).

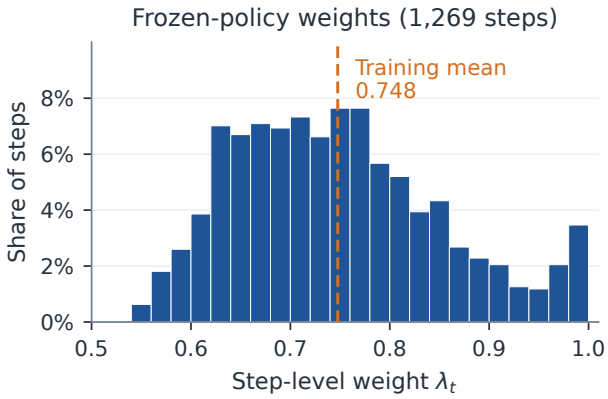


Figure 5: Step weights from the frozen ALFWorld 1.5B final@300 policy, using 128 trajectories and 1,269 decisions on training tasks. The vertical line marks the historical training mean.

H.2 WITHIN-TRAJECTORY SCORE AND ACTION DIVERSITY

A full pre-action state includes the task, model input, interaction history, and current environment state. Fresh responses sampled independently of the original rollout estimate action diversity. If n_u of the n samples execute canonical action u, the disagreement estimator is

$$\widehat{D}(s) = 1 - \frac{\sum_u n_u(n_u - 1)}{n(n-1)}. \tag{33}$$

The parser groups responses by environment-executed action identity. Invalid responses share an identity only when their environment behavior is identical.

Paired sampling. For each of the 32 stratified tasks, we collect eight trajectories, totaling 256 trajectories and 2,181 decisions. One trajectory per task is selected uniformly, followed by two decision steps drawn uniformly without replacement. A step in a trajectory of length T has selection probability $(1/8)(2/T)$. The two selected steps are ordered by their original rollout scores c_t; all 32 pairs have distinct scores and mixing weights. At each state, 32 fresh responses estimate diversity,

totaling 2,048 responses. Forty-one of the 64 states have zero observed disagreement. Figure 4 displays all 32 pairs: 11 diversity differences are positive, 5 negative, and 16 zero.

Task pairs are the units of analysis and resampling. The 95% percentile intervals in Table 10 use 2,000 task-bootstrap replicates with seed 2026092602, preserving both states in each sampled task. They describe uncertainty across this task panel conditional on the frozen checkpoint. The action-diversity difference remains positive after each leave-one-task-out exclusion (0.0569–0.0834). Since trajectory normalization divides both scores by the same positive value, original NLL and criticality induce the same within-trajectory ordering.

H.3 Action-value measurements

The paired diagnostic uses the same 64 states for continuation-value measurements. At each state, $m = 8$ independent response draws give parsed actions U_i, each followed by independent continuation blocks $b = 1, 2$. All 1,024 candidate branches pass restoration checks on observations, admissible actions, model-input tokens, and environment-state hashes. Discounted returns include the initial action and use the same remaining horizon. ALFWorld stores the parsed action in subsequent policy history, so the continuation context depends on the canonical action.

Within each block, returns from draws executing the same action are averaged. Let $\widetilde{G}_i^{(b)}$ be the resulting mean assigned to draw i, retaining the original action multiplicities, and let $\overline{\widetilde{G}}^{(b)} = m^{-1} \sum_i \widetilde{G}_i^{(b)}$. We estimate action-value variation by

$$\widehat{C}(s) = \frac{1}{m-1} \sum_{i=1}^{m} \big(\widetilde{G}_i^{(1)} - \overline{\widetilde{G}}^{(1)}\big)\big(\widetilde{G}_i^{(2)} - \overline{\widetilde{G}}^{(2)}\big). \tag{34}$$

Independent continuation blocks remove noise cross-products in expectation. Shared-action averaging gives identical estimates to repeated draws of the same action; states with a single sampled action therefore have zero estimated variation. The paired sample has 47 such states, and two of the remaining state estimates are negative. Negative finite-sample estimates are retained.

Table 10: Paired measurements ordered by original rollout criticality. Means use all 32 task pairs. Differences are higher-score minus lower-score; intervals use task resampling. The value estimates use same-action averaging in each continuation block.

Quantity	Lower c	Higher c	Difference	95% interval
Action diversity $\widehat{D}$	0.1041	0.1789	0.0747	[0.0248, 0.1334]
Action-value variation $\widehat{C}$	0.2502	0.1710	−0.0792	[−0.3599, 0.1954]

The interval for the action-value difference includes zero, so this panel provides no clear evidence of value separation between the higher- and lower-score states.