RELACE: RETROSPECTIVE LIKELIHOOD-BASED ACTION CREDIT ESTIMATION FOR LONG-HORIZON LANGUAGE AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Group Relative Policy Optimization (GRPO) avoids a separate critic by estimating advantages from rollout groups. For multi-turn agents, however, trajectory-level supervision provides coarse, noisy credit: terminal rewards do not locate errors and can penalize useful actions alongside mistakes. Group-in-Group Policy Optimization (GiGPO) and subsequent methods refine supervision through state-conditioned comparisons, but their credit estimates remain sensitive to downstream decisions and outcomes. We introduce RELACE (Retrospective Likelihood-based Action Credit Estimation), a critic-free framework that integrates retrospective action assessment with state-conditioned advantage estimation. RELACE evaluates executed actions through teacher-forced likelihood scoring under both their original contexts and outcome-augmented contexts. Comparing these likelihoods yields a trajectory-normalized retrospective factor that captures outcome-dependent changes in action plausibility, rather than hindsight plausibility alone. We use this factor to reweight discounted task returns and construct local advantages by comparing weighted returns among actions from equivalent states within a task. This couples retrospective relevance with observed reward, producing fine-grained credit that complements trajectory-level GRPO supervision. Temporal smoothing and success-protecting masking further stabilize the local signal. RELACE requires neither auxiliary value nor reward models nor additional autoregressive rollouts for credit estimation. Experiments on ALFWorld and WebShop with Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct demonstrate substantial improvements over GRPO, GiGPO, and HCAPO. With the 1.5B model, RELACE achieves 96.35% success on ALFWorld and 79.43% on WebShop, surpassing GiGPO by 5.47 and 5.60 percentage points, respectively.

1 INTRODUCTION

Large language models (LLMs) can act as autonomous agents, making sequential decisions in partially observable environments (Yao et al., 2023). Tasks such as web navigation and simulated household interaction require coordinated actions to achieve user goals (Yao et al., 2022; Shridhar et al., 2021). Training agents for these tasks often involves sparse or delayed rewards, including terminal success signals, destabilizing training by obscuring intermediate actions' contributions (Feng et al., 2025). This highlights the temporal credit-assignment problem: an outcome evaluates the trajectory as a whole but provides limited guidance about individual decisions (Harutyunyan et al., 2019).

Group Relative Policy Optimization (GRPO) avoids a separate value critic by estimating relative advantages from samples for the same task (Shao et al., 2024). With outcome-level supervision, however, GRPO assigns the same scalar advantage to all actions within a trajectory. This coarse signal does not distinguish useful decisions from incidental or redundant actions, and a late error can obscure useful earlier behavior (Feng et al., 2025). The challenge is to obtain informative action-level credit while retaining critic-free simplicity.

Group-in-Group Policy Optimization (GiGPO) compares discounted downstream returns among decisions from matched environment states (Feng et al., 2025). Such comparisons account for the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

decision context, but their targets still reflect subsequent actions and transitions. Moreover, limited rollout sampling can make local estimates unreliable, assigning disproportionately large advantages to rare actions with favorable continuations (Li et al., 2026). State-level comparison supports local credit assignment, while leaving room for information about an action's relationship to the outcome.

Hindsight learning offers complementary information. Hindsight Experience Replay reuses achieved outcomes as alternative goals (Andrychowicz et al., 2017), whereas Hindsight Credit Assignment uses outcome-conditioned action distributions to inform credit allocation (Harutyunyan et al., 2019). For language agents, this perspective motivates a practical question: *how does outcome information change the model's assessment of an executed action?* An action may score highly in hindsight because it was already plausible in its original context. Comparing both contexts can distinguish baseline plausibility from outcome sensitivity.

HCAPO applies hindsight-conditioned action scoring to long-horizon language-agent optimization (Tan et al., 2026). Although its theoretical formulation motivates a hindsight-to-original probability ratio, its practical estimator normalizes each hindsight score by the mean hindsight score along the trajectory. It does not explicitly evaluate each executed action's original-context denominator. This raises a design question: how should explicit original-versus-hindsight scoring be combined with state-conditioned comparisons of downstream returns?

We propose RELACE, a critic-free framework that combines explicit retrospective scoring with state-conditioned advantage estimation. After termination, RELACE augments each intermediate context with the observed terminal outcome. The frozen behavior policy scores the same executed action under original and outcome-conditioned contexts using teacher forcing. We form a ratio of these length-normalized likelihood-based scores, normalize the ratios within each trajectory, and clip them before reweighting discounted terminal returns. The resulting weights represent retrospective relevance rather than causal contribution.

RELACE applies temporal smoothing to the weighted returns and computes local relative advantages within state-matched groups. Thus, outcome information enters the return targets *before* local comparison, while a trajectory-level GRPO advantage retains supervision from overall task performance. Following HCAPO, we use temporal smoothing and an asymmetric mask that removes negative local corrections from successful trajectories (Tan et al., 2026). The environment reward is unchanged; RELACE modifies advantage-estimation targets without a separate additive reward. Teacher-forced scoring adds likelihood evaluation without autoregressive rollouts or auxiliary value or reward models.

Our primary contributions are:

- We develop an explicit original-versus-hindsight scoring mechanism for executed language actions. Ratios of length-normalized action scores yield trajectory-normalized, clipped weights for retrospective credit estimation using the behavior policy itself.
- We integrate these weights into a state-conditioned advantage estimator by reweighting discounted task returns before local group comparison. The resulting local signal complements trajectory-level GRPO supervision within critic-free optimization.
- We evaluate RELACE on ALFWorld and WebShop using Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct. With the 1.5B model, RELACE achieves the highest reported mean success rates among the compared methods: 96.35% on ALFWorld and 79.43% on WebShop. These exceed GiGPO by 5.47 and 5.60 percentage points, HCAPO by 9.35 and 10.93 points, and GraphGPO by 3.64 and 0.78 points, respectively.

2 Related work

Credit Assignment in Agentic Reinforcement Learning. Long-horizon language-agent tasks require learning from rewards that provide limited feedback about individual decisions (Feng et al., 2025). Actor–critic implementations of PPO estimate advantages using learned value functions, introducing additional model training and value-estimation error (Schulman et al., 2017). Process supervision offers an alternative source of intermediate feedback through learned step-level evaluators, as demonstrated in mathematical reasoning (Lightman et al., 2023). Critic-free methods such as REINFORCE Leave-One-Out (RLOO) and Group Relative Policy Optimization (GRPO) instead

construct advantages from sampled returns without fitting a separate value network (Ahmadian et al., 2024; Shao et al., 2024). When used with outcome-level supervision, however, a trajectory-level scalar advantage does not distinguish the roles of individual actions, motivating finer-grained credit estimators for multi-turn interaction (Feng et al., 2025).

State-Conditioned and Structured Credit Assignment. GiGPO constructs local relative advantages by grouping decisions with matching environment states and comparing their discounted downstream returns (Feng et al., 2025). HGPO addresses historical-context inconsistency within step-level groups by constructing a hierarchy of groups with different degrees of context agreement (He et al., 2026). GraphGPO aggregates sampled trajectories into a state-transition graph and assigns credit according to estimated progress toward task goals (Cheng et al., 2026). ECPO focuses on the reliability of local comparisons, shrinking low-count action estimates and down-weighting groups dominated by within-action noise (Li et al., 2026). These approaches refine the comparison structure, progress signal, or statistical reliability of step-level credit. RELACE builds on state-conditioned grouping and investigates how outcome-conditioned action scores can inform the return targets used in those comparisons.

Exploration and Training Stability. Complementary work studies the learning dynamics of multi-turn agents. RAGEN examines training instability and the effects of rollout diversity, interaction granularity, and sampling frequency (Wang et al., 2025b). EPO regulates policy entropy across training phases to balance exploration and exploitation (Xu et al., 2025). EMPG modulates policy-gradient signals using step-wise uncertainty and final task outcomes, with an additional signal favoring predictable future interactions (Wang et al., 2025a). These methods provide complementary perspectives on how exploration and update weighting affect learning from sparse feedback.

Hindsight Credit Assignment and Language Agents. Hindsight methods use information observed after an action to improve learning from collected experience. Hindsight Experience Replay (HER) relabels experience with achieved goals, enabling learning even when the intended goal is not reached (Andrychowicz et al., 2017). Hindsight Policy Gradients extends this idea to goal-conditioned policy-gradient estimation (Rauber et al., 2019). Hindsight Credit Assignment (HCA) uses outcome-conditioned action distributions to estimate the relationship between earlier actions and subsequent outcomes (Harutyunyan et al., 2019). Building on HCA, COCOA estimates counterfactual contributions to rewards or reward-relevant outcome representations, addressing limitations of conditioning on detailed future states (Meulemans et al., 2023).

HCAPO brings hindsight-based credit assignment to long-horizon language agents through outcome-conditioned action scoring and a combination of trajectory-level and step-level advantages (Tan et al., 2026). Although its theoretical formulation motivates a hindsight-to-original probability ratio, its practical estimator normalizes each hindsight score by the mean hindsight score along the trajectory. RELACE explicitly scores each executed action under both its original and outcome-conditioned contexts, then incorporates the resulting relevance weights into state-conditioned return comparisons. Its distinction lies in explicit original-context scoring, integration with local group baselines, teacher-forced evaluation, temporal smoothing, and protective masking. The complete estimator is presented in Section 3.

3 RETROSPECTIVE LIKELIHOOD-BASED ACTION CREDIT ASSIGNMENT

We introduce RELACE, a critic-free policy optimization algorithm for fine-grained credit assignment in partially observable multi-turn interactions. RELACE combines three components: (i) an episode-level relative advantage capturing global trajectory quality, (ii) a retrospective likelihood ratio estimating intermediate actions' relevance to the outcome, and (iii) a state-conditioned baseline comparing diverging actions from matched environment states. The term *retrospective* emphasizes that RELACE measures how outcome information changes the policy's assessment of an executed action, rather than the outcome-conditioned score alone.

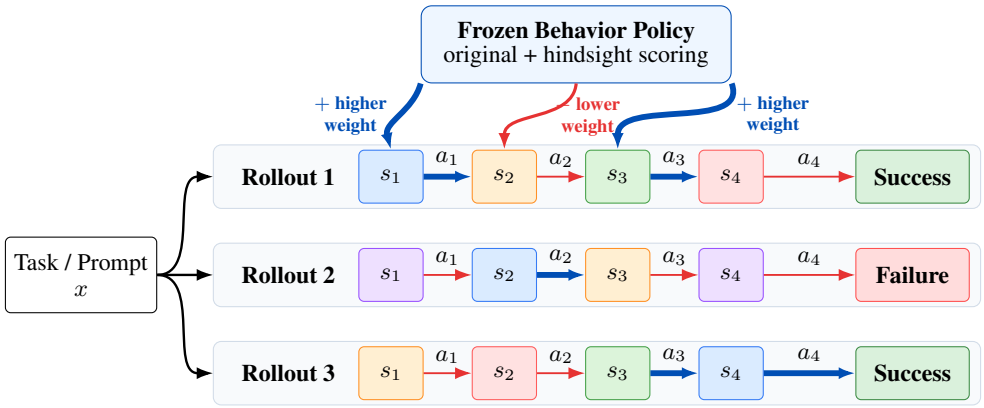


Figure 1: Overview of RELACE. Multiple trajectories are sampled for the same task, with colors indicating matched or similar environment states across rollouts. After the terminal outcome is observed, the frozen behavior policy scores the same executed actions under original and retrospective contexts using teacher forcing. The resulting hindsight reweighted returns are compared across matched states to construct the step-level credit signal. Arrow widths illustrate relative retrospective weights.

3.1 Outcome-conditioned hindsight relabeling

For each task x, we sample K trajectories from the behavior policy $\pi_{\theta_{\text{old}}}$. Trajectory τ_i contains T_i actions, receives terminal reward R_i, and ends with observed outcome s_i^{final}. At step t, $s_{i,t}$ is the observable representation used for grouping, $c_{i,t}$ is the policy's interaction context, and $a_{i,t} = (y_{i,t,1}, \ldots, y_{i,t,L_{i,t}})$ is the executed token sequence. The latent environment state need not be observable. Appendix A gives the full setting and background.

Following hindsight learning (Andrychowicz et al., 2017; Harutyunyan et al., 2019), we construct a retrospective context after termination, using the observed outcome for both successful and unsuccessful trajectories:

$$c_{i,t}^{\text{H}} = \text{Relabel}\left(c_{i,t}, s_i^{\text{final}}\right). \tag{1}$$

The outcome information is the final observation s_i^{final}, used as the hindsight target. The operator Relabel constructs this prompt using the templates in Appendix D.3, retaining the current observation and available interaction history.

We then evaluate the same executed action $a_{i,t}$ under both $c_{i,t}$ and $c_{i,t}^{\text{H}}$, using the frozen behavior policy $\pi_{\theta_{\text{old}}}$ and teacher forcing. No new action is sampled. Holding the executed action fixed measures the score change between the two contexts. The terminal outcome is used only for training-time credit estimation, never for rollout generation or evaluation.

3.2 Retrospective credit reweighting via likelihood ratios

To estimate retrospective relevance, we compare likelihood-based scores for the same executed action under its original and retrospective contexts. For $L_{i,t} > 0$, we use a temperature-scaled geometric mean of token probabilities to account for action length. The two scores are:

$$p_{i,t} = \exp\left(\frac{1}{\tau_{\text{temp}} L_{i,t}} \sum_{j=1}^{L_{i,t}} \log \pi_{\theta_{\text{old}}}\left(y_{i,t,j} \mid c_{i,t}, y_{i,t,<j}\right)\right), \tag{2}$$

$$p_{i,t}^{\text{H}} = \exp\left(\frac{1}{\tau_{\text{temp}} L_{i,t}} \sum_{j=1}^{L_{i,t}} \log \pi_{\theta_{\text{old}}}\left(y_{i,t,j} \mid c_{i,t}^{\text{H}}, y_{i,t,<j}\right)\right), \tag{3}$$

where $\tau_{\text{temp}} > 0$ controls score sharpness. These length-normalized scores are not normalized probabilities over variable-length actions, and this score temperature is distinct from decoding temperature. We refer to $p_{i,t}^{\text{H}}$ as the *retrospective likelihood score* of the executed action.

With numerical stabilizer $\epsilon > 0$, we first compute the raw retrospective score ratio:

$$u_{i,t} = \frac{p^{\mathrm{H}}_{i,t}}{p_{i,t} + \epsilon}. \tag{4}$$

A ratio $u_{i,t} > 1$ indicates a higher score under the retrospective context; its magnitude depends on the stabilizer when $p_{i,t}$ is small. To account for varying magnitudes across trajectories, we normalize the ratio by its trajectory mean, $\bar{u}_i = \frac{1}{T_i} \sum_{k=1}^{T_i} u_{i,k}$, yielding a clipped relative relevance:

$$\rho_{i,t} = \mathrm{clip}\left(\frac{u_{i,t}}{\bar{u}_i + \epsilon}, \rho_{\min}, \rho_{\max}\right), \tag{5}$$

where $0 < \rho_{\min} \leq 1 \leq \rho_{\max}$. We distinguish raw ratio $u_{i,t}$ from normalized, clipped weight $\rho_{i,t}$ used in credit estimation. Writing $Q_{i,t} = \gamma^{T_i - t} R_i$ for the original discounted return, we obtain:

$$Q^{\mathrm{H}}_{i,t} = \rho_{i,t} \gamma^{T_i - t} R_i. \tag{6}$$

This reweights the return target without changing the environment reward. Before clipping and ignoring ϵ, the weights have trajectory mean one; this does not in general preserve the sum of discounted return targets. In particular, if $R_i = 0$, every $Q^{\mathrm{H}}_{i,t}$ is zero, regardless of the hindsight scores.

3.3 State-conditioned grouping and asymmetric masking

To compare an action with sampled alternatives at a decision node, RELACE groups actions originating from matched observable environment states across the rollout batch (Feng et al., 2025). For a given task x and anchor state s, the *anchor state group* is defined as

$$\mathcal{G}_x(s) = \{(i,t) \mid x_i = x,\ s_{i,t} = s\},$$

which contains all batch occurrences from the same observed state.

State matching determines which decisions share a local baseline. Here, $s_{i,t}$ denotes the observable representation used for grouping, whereas $c_{i,t}$ denotes the full policy context used for likelihood scoring. Matched observations need not identify the same latent state under partial observability. The matching procedure appears in Appendix C.

Following HCAPO (Tan et al., 2026), we use adjacent-step smoothing to share the weighted return signal with the preceding action:

$$\widetilde{Q}^{\mathrm{H}}_{i,t} = \alpha Q^{\mathrm{H}}_{i,t} + (1 - \alpha) Q^{\mathrm{H}}_{i,\min(t+1,T_i)}, \qquad \alpha \in [0,1]. \tag{7}$$

At $t = T_i$, this leaves the terminal target unchanged; $\alpha = 1$ disables smoothing. Both the focal target and the local baseline use the same smoothed, weighted quantity. For a nonempty group, its mean is

$$\mu_V(x,s) = \frac{1}{|\mathcal{G}_x(s)|} \sum_{(j,k) \in \mathcal{G}_x(s)} \widetilde{Q}^{\mathrm{H}}_{j,k}, \tag{8}$$

and its sample standard deviation is

$$\sigma_V(x,s) = \sqrt{\frac{1}{|\mathcal{G}_x(s)| - 1} \sum_{(j,k) \in \mathcal{G}_x(s)} \left(\widetilde{Q}^{\mathrm{H}}_{j,k} - \mu_V(x,s)\right)^2}, \tag{9}$$

when $|\mathcal{G}_x(s)| \geq 2$. Singleton groups receive zero local advantage without evaluating their sample standard deviation. Constant-target groups also yield zero local advantage. For groups of size at least two, the state-relative advantage is:

$$A^{\mathrm{H}}_{i,t} = \frac{\widetilde{Q}^{\mathrm{H}}_{i,t} - \mu_V(x_i, s_{i,t})}{\sigma_V(x_i, s_{i,t}) + \epsilon}. \tag{10}$$

A successful trajectory can receive a negative local advantage when its weighted target falls below its matched-group mean; finite-sample noise can also affect that comparison. Following HCAPO (Tan

et al., 2026), we apply a conservative *do-no-harm mask*. Let $d_i \in \{0, 1\}$ indicate trajectory success; the protected micro-advantage becomes:

$$\widehat{A}^{\mathrm{H}}_{i,t} = \begin{cases} 0, & d_i = 1 \wedge A^{\mathrm{H}}_{i,t} < 0, \\ A^{\mathrm{H}}_{i,t}, & \text{otherwise.} \end{cases} \tag{11}$$

The mask removes every negative local correction on successful trajectories, whether or not caused by noise, and leaves the macro-advantage unchanged.

3.4 Macro advantage and policy objective

RELACE also computes an episode-wise macro advantage A^{E}_i within each task group to preserve preference for high-reward trajectories:

$$A^{\mathrm{E}}_i = \frac{R_i - \mu_R(x_i)}{\sigma_R(x_i) + \epsilon}. \tag{12}$$

The final advantage combines the episode-level and protected local terms:

$$A^{\mathrm{RELACE}}_{i,t} = A^{\mathrm{E}}_i + \omega \widehat{A}^{\mathrm{H}}_{i,t}, \tag{13}$$

where $\omega \geq 0$ balances global and local preferences. Credit estimates are computed with the frozen behavior policy and detached during policy optimization. The scalar $A^{\mathrm{RELACE}}_{i,t}$ is broadcast to all valid tokens of action $a_{i,t}$. Let $r_{i,t,j}(\theta) = \pi_\theta(y_{i,t,j} \mid c_{i,t}, y_{i,t,<j}) / \pi_{\theta_{\mathrm{old}}}(y_{i,t,j} \mid c_{i,t}, y_{i,t,<j})$ be the policy-update ratio, evaluated under the original context. This token-level ratio is distinct from the retrospective score ratio $u_{i,t}$. The policy is updated using the clipped PPO surrogate objective (Schulman et al., 2017):

$$\begin{aligned} \mathcal{J}_{\mathrm{RELACE}}(\theta) = \mathbb{E}\Big[\min \big(& r_{i,t,j}(\theta) A^{\mathrm{RELACE}}_{i,t}, \\ & \mathrm{clip}(r_{i,t,j}(\theta), 1 - \epsilon_{\mathrm{ppo}}, 1 + \epsilon_{\mathrm{ppo}}) A^{\mathrm{RELACE}}_{i,t} \big) \Big], \end{aligned} \tag{14}$$

Here, the expectation uses the task, trajectory, and valid-token aggregation specified in Eq. (16), replacing A^E_i with $A^{\mathrm{RELACE}}_{i,t}$. We retain the KL regularization against the reference policy from that objective. The hindsight context is used for credit estimation, not as the policy-update input. Algorithm 1 in Appendix B gives the training procedure.

4 Experiments and analysis

4.1 Experimental setup

Benchmarks and metrics. We evaluate RELACE on two multi-turn environments with sparse outcome feedback: ALFWorld (Shridhar et al., 2021) and WebShop (Yao et al., 2022). ALFWorld requires household planning across six task categories: Pick, Clean, Heat, Cool, Examine, and Pick 2. We report overall and category-specific success rates. WebShop requires product search, attribute inspection, and purchases satisfying natural-language requests. We report mean task score (0–100) and success rate (SR), measuring complete task satisfaction. These distinguish partial from full completion. Episodes are capped at 50 environment steps in ALFWorld and 15 in WebShop.

Models and comparisons. We initialize policies from Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Qwen Team, 2024). The comparison includes prompting-based agents, ReAct and Reflexion (Yao et al., 2023; Shinn et al., 2023); actor–critic training with PPO (Schulman et al., 2017); and critic-free methods, RLOO, GRPO, GiGPO, HCAPO, and GraphGPO (Ahmadian et al., 2024; Shao et al., 2024; Feng et al., 2025; Tan et al., 2026; Cheng et al., 2026). Table 1 combines our RELACE results with published baseline configurations: GRPO, GiGPO, and GraphGPO follow Cheng et al. (2026), while HCAPO follows Tan et al. (2026). We separately compare RELACE and GiGPO validation histories at shared checkpoints. Thus, the benchmark table and learning curves characterize task performance and training dynamics.

Table 1: Benchmark performance of RELACE and the listed baseline configurations. WebShop Score is the mean task success rate (SR) on a 0–100 scale and all ALFWorld entries are success percentages. Baseline values follow Cheng et al. (2026) and Tan et al. (2026); Boldface and underlining indicate the highest and second-highest means within each model-size block, respectively.

Method	WebShop		ALFWorld						
	Score	SR	Pick	Clean	Heat	Cool	Examine	Pick 2	Avg
Closed-Source Models									
GPT-5.6-Sol-High	28.0	25.3	**98.7**	**76.6**	85.7	**83.6**	**89.8**	**80.2**	**86.0**
GPT-5.6-Sol-Medium	26.7	22.3	95.4	64.3	**87.6**	79.8	88.2	73.1	80.8
GPT-4o	31.8	23.7	75.3	60.2	21.6	31.2	56.7	49.8	48.0
Gemini-2.5-Pro	**42.5**	**35.9**	92.8	63.2	26.6	62.1	69.0	58.7	60.3
Prompting (Qwen2.5-1.5B-Instruct)									
ReAct	40.1	11.3	17.4	20.5	7.7	15.7	6.2	2.0	12.8
Reflexion	55.8	21.9	35.3	22.2	19.4	21.7	13.6	3.7	21.8
RL Training (Qwen2.5-1.5B-Instruct)									
PPO	73.8 ± 3.0	51.5 ± 2.9	64.8 ± 3.5	40.5 ± 6.9	46.4 ± 4.0	57.1 ± 4.9	60.6 ± 6.6	47.4 ± 1.9	54.4 ± 3.1
RLOO	73.9 ± 5.6	52.1 ± 6.7	88.3 ± 3.0	52.8 ± 8.6	66.4 ± 5.5	71.0 ± 5.9	62.8 ± 8.7	56.9 ± 4.7	69.7 ± 2.5
GRPO	84.73 ± 0.49	71.35 ± 2.05	82.89 ± 3.62	82.14 ± 6.37	77.78 ± 4.54	73.86 ± 6.84	78.57 ± 0.00	71.43 ± 3.89	77.86 ± 1.33
HCAPO	83.8 ± 0.7	68.5 ± 1.0	88.6 ± 7.0	97.6 ± 1.8	90.7 ± 6.9	84.2 ± 0.0	75.0 ± 0.0	74.2 ± 6.9	87.0 ± 4.1
GiGPO	87.94 ± 0.43	73.83 ± 2.30	98.81 ± 1.68	95.16 ± 3.89	94.44 ± 0.00	81.46 ± 0.56	78.57 ± 0.00	$\mathbf{93.65 \pm 5.94}$	90.88 ± 0.97
GraphGPO	89.29 ± 1.48	78.65 ± 3.86	95.15 ± 1.62	$\mathbf{100.0 \pm 0.00}$	$\mathbf{96.30 \pm 2.61}$	85.26 ± 2.58	85.71 ± 5.83	$\mathbf{93.65 \pm 2.24}$	92.71 ± 1.32
RELACE	$\mathbf{90.90 \pm 1.04}$	$\mathbf{79.43 \pm 2.74}$	$\mathbf{100.00 \pm 0.00}$	98.41 ± 2.75	94.45 ± 9.61	$\mathbf{96.49 \pm 6.08}$	$\mathbf{87.46 \pm 11.38}$	93.33 ± 3.12	$\mathbf{96.35 \pm 0.45}$
RL Training (Qwen2.5-7B-Instruct)									
GRPO	84.31 ± 1.27	75.00 ± 2.78	88.98 ± 5.30	91.98 ± 4.43	90.74 ± 5.24	77.89 ± 4.58	78.57 ± 0.00	71.43 ± 3.89	83.33 ± 2.05
HCAPO	85.1 ± 1.3	73.8 ± 2.8	99.10 ± 1.30	97.30 ± 1.90	81.80 ± 8.80	90.80 ± 6.60	90.30 ± 2.00	81.90 ± 10.00	91.40 ± 2.30
GiGPO	86.72 ± 1.44	78.38 ± 1.94	97.53 ± 1.75	$\mathbf{100.0 \pm 0.00}$	$\mathbf{94.44 \pm 0.00}$	83.98 ± 1.32	90.48 ± 6.73	$\mathbf{100.0 \pm 0.00}$	94.27 ± 1.33
GraphGPO	86.94 ± 0.68	80.31 ± 1.33	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$	$\mathbf{94.44 \pm 0.0}$	91.40 ± 1.5	92.86 ± 5.8	92.06 ± 2.2	95.31 ± 1.10
RELACE	$\mathbf{90.34 \pm 0.80}$	$\mathbf{82.29 \pm 1.80}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{100.00 \pm 0.00}$	94.11 ± 5.37	$\mathbf{94.71 \pm 5.03}$	$\mathbf{94.44 \pm 9.62}$	$\mathbf{100.00 \pm 0.00}$	$\mathbf{97.66 \pm 0.78}$

Training and reporting. Our implementation extends the `verl-agent` framework (Feng et al., 2025).[1] For the 1.5B experiments, each training batch contains 16 prompts with $K = 8$ rollouts per prompt. The actor learning rate is 10^{-6}, and the KL regularization coefficient is 0.01. RELACE uses the retrospective weighting, temporal smoothing, and asymmetric masking defined in Section 3, without an auxiliary value or reward model. Validation is logged every five training steps; Figure 2 covers steps 0–200. Hyperparameters, seed coverage, and aggregation details appear in Appendix C; agent and hindsight prompts appear in Appendix D.

4.2 Main results

RELACE achieves the highest reported mean on all three reported aggregate metrics in Table 1—ALFWorld overall success, WebShop score and success rate—at both model sizes. The improvements extend beyond trajectory-level GRPO to the listed step-level and hindsight-based baselines, showing that retrospective likelihood correction can complement structured local comparisons.

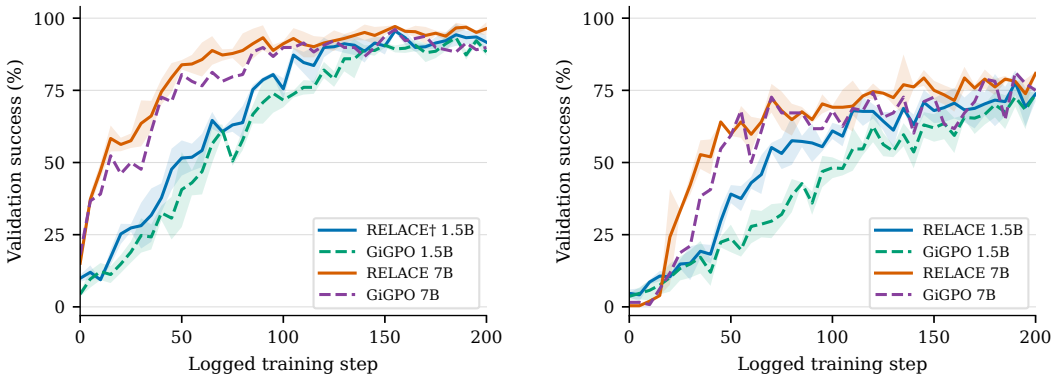
Strong performance with a 1.5B policy. On ALFWorld, RELACE reaches 96.35% overall success, exceeding GRPO, GiGPO, HCAPO, and GraphGPO by 18.49, 5.47, 9.35, and 3.64 percentage points, respectively. The subtask breakdown shows a large Cool gain: 96.49% versus 85.26% for GraphGPO and 81.46% for GiGPO. RELACE also reaches 100% on Pick and the highest Examine success among the listed 1.5B methods. The overall advantage includes gains on tasks requiring coordinated actions, while Clean, Heat, and Pick 2 remain strengths of other methods.

On WebShop, the policy achieves a score of 90.90 and a success rate of 79.43%. The success-rate gains over GRPO, GiGPO, and HCAPO are 8.08, 5.60, and 10.93 percentage points. Relative to GraphGPO, RELACE improves score by 1.61 points and success by 0.78 percentage points. Both gains indicate higher task rewards alongside more completed shopping tasks.

Benefits persist at 7B scale. With Qwen2.5-7B-Instruct, RELACE reaches 97.66% ALFWorld success, improving over GiGPO and GraphGPO by 3.39 and 2.35 percentage points. It achieves 100% on Pick, Clean, and Pick 2, and the highest Cool and Examine means among these methods. On WebShop, RELACE obtains a score of 90.34 and 82.29% success, exceeding GiGPO by 3.62 score points and 3.91 success percentage points, and GraphGPO by 3.40 and 1.98, respectively.

These results support the combination of original-context likelihood correction and state-conditioned return comparison across both policy sizes.

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



(a) ALFWorld Success Rate

(b) WebShop Success Rate

Figure 2: Validation success over steps 0–200 for RELACE (solid) and GiGPO (dashed), using Qwen2.5-1.5B/7B-Instruct. Lines show seed means; shading shows one sample standard deviation, omitted for single-seed baselines. Seed coverage and the † run-assignment note appear in Appendix C.

Table 2: Component comparison on WebShop. Columns show the focal target and group-baseline input before smoothing.

Variant	Focal target	Baseline input	Reported SR (%)
GiGPO	Q	Q	73.83
Hindsight-score weighting	$p^H Q$	$p^H Q$	75.78
Focal-only ratio weighting	ρQ	Q	60.94*
RELACE	ρQ	ρQ	**79.43**

4.3 Learning dynamics

Figure 2 compares validation success over the first 200 logged training steps. The curves compare RELACE's learning progress with GiGPO, complementing the benchmark results in Table 1.

The 1.5B curves separate most during early and intermediate training. On ALFWorld at step 50, RELACE reaches 51.56% success versus 40.63% for GiGPO, a 10.94-point advantage. On WebShop, the gap is larger: 39.06% versus 23.70%, or 15.36 points. The WebShop advantage remains 12.76 points at step 100 (60.94% versus 48.18%). The curves show faster early learning and higher success for RELACE. RELACE's computational overhead appears in Appendix C.

4.4 Ablation study

We examine two retrospective-estimator choices: correcting hindsight scores using the original action likelihood, and applying the resulting weights to the local comparison group. Table 2 reports four WebShop variants. Using $Q_{i,t} = \gamma^{T_i - t} R_i$, GiGPO compares unweighted returns Q. Hindsight-score weighting uses $p^H Q$ for both the focal action and its group baseline, without the original-context denominator. Focal-only ratio weighting uses ρQ for the focal action but retains a baseline constructed from Q. Full RELACE uses ρQ on both sides, with the smoothing and normalization defined in Section 3.3.

Original-context correction. Hindsight-score weighting reaches 75.78%, compared with 73.83% for GiGPO. Full RELACE reaches 79.43%, a further 3.65-point increase. This ordering supports comparing an action's hindsight and original plausibility: a high hindsight score may reflect an already probable action.

Consistency of the local baseline. The focal-only variant peaks at 60.94%, indicating that focal-return weighting alone is insufficient here. An unweighted baseline compares differently trans-

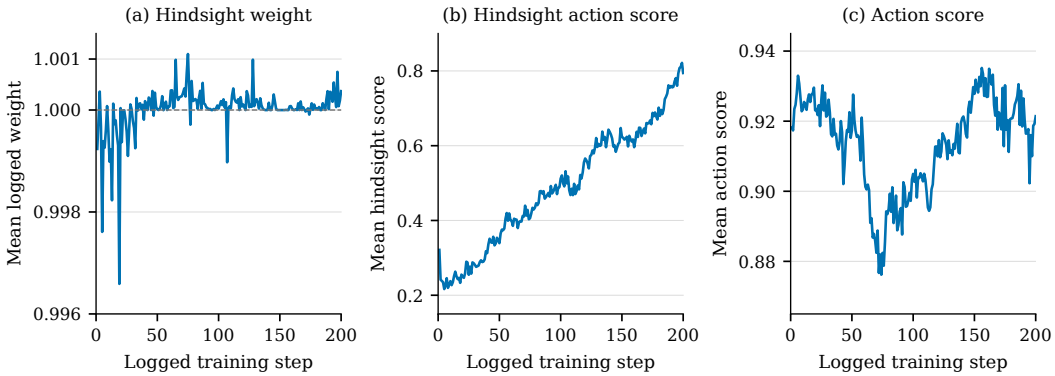


Figure 3: RELACE scoring dynamics on WebShop (Qwen2.5-1.5B-Instruct, steps 1–200): (a) hindsight weight, with a dashed reference at one; (b) hindsight action score; and (c) action score.

formed quantities. In RELACE, both focal targets and group statistics use the same hindsight-weighted, smoothed returns. The ordering supports this design motivation.

4.5 Dynamics of retrospective scoring

Complementing validation, we examine 200 training steps of Qwen2.5-1.5B-Instruct on WebShop. Figure 3 tracks three run diagnostics: mean retrospective weight, mean hindsight score, and action score. These curves characterize scoring during training.

Stability of the mean retrospective weight. The logged mean weight stays near one, ranging from 0.9966 to 1.0011. Its temporal standard deviation falls from 6.84×10^{-4} over steps 1–50 to 1.58×10^{-4} over steps 151–200, a reduction of about 77%. This indicates greater late-stage stability of the batch mean.

Increasing hindsight scores. The mean hindsight score rises from 0.241 over steps 1–10 to 0.785 over steps 191–200. Successive 50-step averages are 0.279, 0.434, 0.563, and 0.687, showing an upward trend despite batch fluctuations. Thus, later actions score higher in outcome-conditioned contexts, consistent with greater compatibility with the final-observation hindsight targets.

Non-monotonic action-score evolution. The action score follows a different trajectory. Its 50-step average decreases from 0.921 over steps 1–50 to 0.898 over steps 51–100, then recovers to 0.911 and 0.924 in the next two windows. Hindsight scores rise while action scores decline and recover, rather than both increasing uniformly. These patterns suggest changing action preferences during training.

5 Conclusion

Sparse terminal rewards provide limited credit-assignment guidance in long-horizon tasks. We introduced RELACE, a critic-free framework that uses teacher forcing to compare executed actions under original and outcome-conditioned contexts. The relevance weights refine discounted returns before state-conditioned comparison, combining local credit with GRPO supervision without auxiliary models or additional autoregressive rollouts.

Experiments on ALFWorld and WebShop show higher reported aggregate performance than compared baselines at 1.5B and 7B scales, with early gains over GiGPO at 1.5B. Training diagnostics show increasing hindsight scores and a more stable mean retrospective weight, supporting learning from sparse outcomes.

These scores represent retrospective relevance rather than causal contribution. Future work will examine their relationship to action utility, improve partially observable state matching, and evaluate generalization across environments.

AI USE STATEMENT

Generative AI tools assisted with sentence-level editing, literature lookup and citation formatting. Also assisted in implementing code experimentation, and LaTeX compilation checks. The authors remain responsible for the final text, analyses, and claims.

REPRODUCIBILITY STATEMENT

Our implementation is based on the publicly available codebase cited in the paper, Feng et al. (2025), from which we modified the relevant components to implement RELACE. The main modified training code file is included as supplementary material. We will release the complete source code, including all modifications, scripts, and configuration files, with the camera-ready version of the paper.

To facilitate reproduction, we provide all essential implementation details, including the training algorithm, model configurations, environment settings, hyperparameters, optimization details, and evaluation protocols. In particular, Appendix C provides the complete experimental configuration, including batch sizes, rollout settings, learning rates, KL coefficients, sampling temperatures, RELACE-specific hyperparameters, and state-matching details. The prompt templates and additional implementation details are also provided in the appendix.

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)*, 2024. URL https://arxiv.org/abs/2402.14740.

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems*, volume 30, pp. 5048–5058, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/453fadbd8a1a3af50a9df4df899537b5-Abstract.html.

Xin Cheng, Shuo He, Lang Feng, HaiYang Xu, Ming Yan, Lei Feng, and Bo An. Beyond trajectory-level attribution: Graph-based credit assignment for agentic reinforcement learning. *arXiv preprint arXiv:2605.26684*, 2026. URL https://arxiv.org/abs/2605.26684.

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, 2025. doi: 10.52202/085713-1544. URL https://arxiv.org/abs/2505.10978.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado P. van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and Remi Munos. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, volume 32, pp. 12467–12476, 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/hash/195f15384c2a79cedf293e4a847ce85c-Abstract.html.

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*, 2026. URL https://arxiv.org/abs/2602.22817.

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. doi: 10.48550/arXiv.2606.05885. URL https://arxiv.org/abs/2606.05885.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Alexander Meulemans, Simon Schug, Seijin Kobayashi, Nathaniel Daw, and Gregory Wayne. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. In *Advances in Neural Information Processing Systems*, 2023. URL https://papers.neurips.cc/paper_files/paper/2023/hash/d8bd445c2abe1343cce0e14b361b2fb3-Abstract-Conference.html.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115.

Paulo Rauber, Avinash Ummadisingu, Filipe Mutz, and Jürgen Schmidhuber. Hindsight policy gradients. In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1711.06006.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347. URL https://arxiv.org/abs/1707.06347.

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. URL https://arxiv.org/abs/2402.03300.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2010.03768.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight credit assignment for long-horizon LLM agents. *arXiv preprint arXiv:2603.08754*, 2026. doi: 10.48550/arXiv.2603.08754. URL https://arxiv.org/abs/2603.08754.

Jiawei Wang, Jiacai Liu, Yuqian Fu, Yingru Li, Xintao Wang, Yuan Lin, Yu Yue, Lin Zhang, Yang Wang, and Ke Wang. Harnessing uncertainty: Entropy-modulated policy gradients for long-horizon LLM agents. *arXiv preprint arXiv:2509.09265*, 2025a. URL https://arxiv.org/abs/2509.09265.

Yu Wang. The dark room in the reward channel: Dense prediction rewards collapse GRPO-trained LLM agents – and the channel, not the content, decides what works. *arXiv preprint arXiv:2607.21273*, 2026. doi: 10.48550/arXiv.2607.21273. URL https://arxiv.org/abs/2607.21273.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Yiping Lu, Kyunghyun Cho, Jiajun Wu, Li Fei-Fei, Lijuan Wang, Yejin Choi, and Manling Li. RAGEN: Understanding self-evolution in LLM agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025b. URL https://arxiv.org/abs/2504.20073.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N. Metaxas. EPO: Entropy-regularized policy optimization for LLM agents reinforcement learning. *arXiv preprint arXiv:2509.22576*, 2025. URL https://arxiv.org/abs/2509.22576.

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*, 2022. URL https://arxiv.org/abs/2207.01206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.03629.

A PRELIMINARIES

Markov decision processes. A Markov decision process (MDP) is defined by $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P(s' \mid s, a)$ is the transition kernel, R is the reward function, and $\gamma \in [0, 1]$ is the discount factor. A policy $\pi(a \mid s)$ selects an action conditioned on the current state, and the objective is to maximize expected discounted return.

Partial observability. Long-horizon language agents generally do not have direct access to the underlying environment state. We therefore model interaction as a partially observable Markov decision process (POMDP),

$$\mathcal{M} = (\mathcal{Z}, \mathcal{A}, \mathcal{O}, P, \Omega, R, \gamma),$$

where $\mathcal{Z}$ denotes latent states, $\mathcal{O}$ observations, and $\Omega(o \mid z)$ the observation kernel. Rather than conditioning only on the current latent state, the policy acts from its available interaction history. For language agents, we represent this information by a context c_t, containing the task specification, previous observations and actions, and the current observation.

Language-agent setting. For each task x, we sample a group of K trajectories from the behavior policy $\pi_{\theta_{\text{old}}}$. Trajectory

$$\tau_i = (s_{i,1}, a_{i,1}, \ldots, s_{i,T_i}, a_{i,T_i}, s_i^{\text{final}})$$

contains T_i actions, terminates in observed outcome s_i^{final}, and receives terminal reward $R_i = R(\tau_i)$. At turn t, the context $c_{i,t}$ contains the task and observed interaction history, while the action $a_{i,t}$ is a variable-length token sequence

$$a_{i,t} = (y_{i,t,1}, \ldots, y_{i,t,L_{i,t}}).$$

We use $s_{i,t}$ to denote the observable environment representation used for state-conditioned grouping. Under partial observability, identical observed states need not correspond to identical latent environment states.

Group Relative Policy Optimization. GRPO (Shao et al., 2024) avoids training a separate value critic by constructing relative advantages from groups of rollouts sampled for the same task. For trajectory τ_i, the episode-level advantage is

$$A_i^E = \frac{R_i - \mu_R(x)}{\sigma_R(x) + \epsilon}, \qquad r_{i,t,j}(\theta) = \frac{\pi_\theta(y_{i,t,j} \mid c_{i,t}, y_{i,t,<j})}{\pi_{\theta_{\text{old}}}(y_{i,t,j} \mid c_{i,t}, y_{i,t,<j})}, \tag{15}$$

where $\mu_R(x)$ and $\sigma_R(x)$ are reward statistics computed over the rollout group for task x, and $\epsilon > 0$ ensures numerical stability. GRPO optimizes the clipped surrogate objective

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{K}\sum_{i=1}^{K}\frac{1}{N_i}\sum_{t,j}\left\{\ell\big(r_{i,t,j}(\theta), A_i^E\big) - \beta D_{i,t,j}\right\}\right], \tag{16}$$

$$\ell(r, A) = \min\{rA,\ \text{clip}(r, 1-\epsilon_{\text{ppo}}, 1+\epsilon_{\text{ppo}})A\},$$

where $N_i = \sum_t L_{i,t}$ and $D_{i,t,j}$ denotes the token-level KL penalty relative to a reference policy. The expectation is over tasks and rollout groups, while β and ϵ_{ppo} control KL regularization and PPO clipping, respectively.

When only a terminal reward is available, A_i^E is shared by every action in trajectory τ_i. GRPO therefore distinguishes trajectories by overall outcome but does not by itself distinguish the contributions of individual intermediate actions.

Hindsight credit assignment. Hindsight Credit Assignment (HCA) (Harutyunyan et al., 2019) uses future outcomes to retrospectively assess earlier actions. For a Markov state s, define the hindsight action posterior

$$h_k^\pi(a \mid s, s') = \Pr_\pi(A_0 = a \mid S_0 = s, S_k = s').$$

For $\pi(a \mid s) > 0$, HCA gives the action-value identity

$$Q^\pi(s, a) = r(s, a) + \mathbb{E}_{\tau \sim \pi(\cdot|s)}\left[\sum_{k \geq 1} \gamma^k \frac{h_k^\pi(a \mid s, S_k)}{\pi(a \mid s)} r_k\right], \tag{17}$$

where $r(s, a)$ is the expected immediate reward and r_k is a future reward. The expectation is over trajectories beginning from s with $A_0 \sim \pi(\cdot \mid s)$, rather than trajectories conditioned on $A_0 = a$.

For sparse terminal rewards, HCAPO (Tan et al., 2026) motivates the step-level quantity

$$Q^H_{i,t} = \frac{h^\pi(a_{i,t} \mid c_{i,t}, s_i^{\text{final}})}{\pi(a_{i,t} \mid c_{i,t})} \gamma^{T_i - t} R_i, \tag{18}$$

with $\pi = \pi_{\theta_{\text{old}}}$ and the hindsight distribution conditioned on the observed terminal outcome. In practical language-agent implementations, the exact hindsight posterior is unavailable and must be approximated using policy likelihoods under outcome-conditioned contexts.

The likelihood-based scores used by RELACE are therefore practical credit proxies rather than exact HCA posteriors; the HCA identity above does not imply that the resulting estimator is unbiased.

B FULL RELACE ALGORITHM

Algorithm 1 summarizes the complete RELACE training procedure. The algorithm first samples grouped trajectories using the frozen behavior policy, retrospectively scores each executed action under its original and outcome-conditioned contexts, constructs trajectory-normalized likelihood-ratio weights, and uses these weights to form state-conditioned local advantages. These local advantages are then combined with the trajectory-level GRPO advantage for policy optimization.

Algorithm 1 Retrospective Likelihood-based Action Credit Estimation (RELACE)

Require: Policy $\pi_{\theta_{\text{old}}}$, task batch $\mathcal{X}$, group size K, discount γ, micro-advantage weight ω, smoothing coefficient α, ratio bounds $\rho_{\min}, \rho_{\max}$, and numerical constant ϵ

for each task $x \in \mathcal{X}$ **do**
  Sample K trajectories $\{\tau_i\}_{i=1}^K \sim \pi_{\theta_{\text{old}}}$
  Record states, actions, final states, returns, and success labels
  for each trajectory τ_i **do**
    for $t = 1, \ldots, T_i$ **do**
      Construct outcome-conditioned context $c^{\text{H}}_{i,t} \leftarrow \text{Relabel}(c_{i,t}, s_i^{\text{final}})$
      Teacher-force $a_{i,t}$ under $c_{i,t}$ and $c^{\text{H}}_{i,t}$
      Compute $p_{i,t}$ and $p^{\text{H}}_{i,t}$ using Eqs. (2)–(3)
      $u_{i,t} \leftarrow p^{\text{H}}_{i,t} / (p_{i,t} + \epsilon)$
    end for
    $\bar{u}_i \leftarrow T_i^{-1} \sum_{t=1}^{T_i} u_{i,t}$
    for $t = 1, \ldots, T_i$ **do**
      $\rho_{i,t} \leftarrow \text{clip}(u_{i,t} / (\bar{u}_i + \epsilon), \rho_{\min}, \rho_{\max})$
      $Q^{\text{H}}_{i,t} \leftarrow \rho_{i,t} \gamma^{T_i - t} R_i$
    end for
    Optionally smooth $Q^{\text{H}}_{i,t}$ using Eq. (7)
  end for
  Group valid actions within each task using the state-matching rule in Section 3.3
  Compute $\mu_V(x, s)$ and $\sigma_V(x, s)$
  Compute state-relative retrospective likelihood-based advantages $A^{\text{H}}_{i,t}$
  Apply the do-no-harm mask to obtain $\widehat{A}^{\text{H}}_{i,t}$
  Compute episode-wise advantages A^{E}_i
  Combine advantages: $A^{\text{RELACE}}_{i,t} \leftarrow A^{\text{E}}_i + \omega \widehat{A}^{\text{H}}_{i,t}$
end for
Update θ by maximizing $\mathcal{J}_{\text{RELACE}}(\theta)$

C EXPERIMENTAL DETAILS AND HYPERPARAMETERS

C.1 TRAINING CONFIGURATION.

For the 1.5B experiments, we use a training batch size of 16 with $K = 8$ environment rollouts per prompt. The actor learning rate is 10^{-6} and the KL regularization coefficient is 0.01. KL shaping of the environment reward is disabled. Rollouts are generated using vLLM with sampling temperature 1.0, while validation uses temperature 0.4 and a validation batch size of 128. We evaluate the policy every five training steps. Episodes are limited to 50 environment steps in ALFWorld and 15 steps in WebShop.

For RELACE, we set the temporal smoothing coefficient to $\alpha = 0.5$, the micro-advantage weight to $\omega = 1$, and clip the normalized retrospective likelihood ratio to $[\rho_{\min}, \rho_{\max}] = [0.8, 1.2]$. The likelihood-score temperature is 1. Temporal smoothing and the success-protecting mask are enabled. The discount factor is $\gamma = 0.6$ for RELACE and $\gamma = 0.95$ for the GiGPO baseline Feng et al. (2025). The PPO mini-batch size is 64 for all experiments except for for WebShop 7B, where the mini-batch size is 128.

State matching. The exact normalized equality in Section 3.3 defines idealized anchor groups. In practice, we construct state-conditioned groups within each task using string-similarity matching with threshold 0.95. Exact matching can produce sparse groups when observable representations differ textually, while approximate matching risks grouping distinct decision situations.

C.2 COMPUTATIONAL OVERHEAD ANALYSIS

Following the stage-wise analysis of HCAPO (Tan et al., 2026, Section 6.5), we profile RELACE on WebShop with Qwen2.5-1.5B-Instruct on one node with eight NVIDIA H100 80GB GPUs. The run uses 16 prompts and 8 rollouts per prompt, an actor PPO mini-batch size of 64, and a maximum of 15 environment steps. We average the first 200 timed training iterations (logged steps 1–200), retaining initial iterations. This dedicated timing run is separate from the validation histories in Figure 2.

Cost of retrospective scoring. Figure 4 shows that generation remains the dominant stage at $49.16\,\text{s}$ per iteration, followed by actor updates at $9.03\,\text{s}$. Hindsight scoring averages $6.85\,\text{s}$, accounting for 7.97% of the full logged iteration time of $86.01\,\text{s}$. Restricting the denominator to the five major stages—generation, actor update, hindsight scoring, old-policy scoring, and reference scoring—gives $68.90\,\text{s}$ and a hindsight share of 9.94%. The remaining $17.11\,\text{s}$ comprises other logged work, including advantage and reward processing, periodic evaluation, and checkpoint saving.

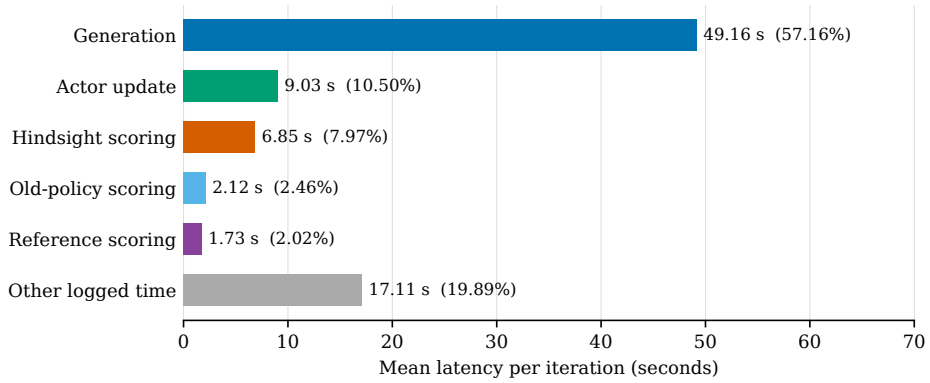


Figure 4: Mean training latency on WebShop over logged steps 1–200 of one timing run. Labels show seconds per iteration and the share of full logged iteration time. Hindsight scoring (orange) averages $6.85\,\text{s}$, or 7.97% of the full iteration. "Other logged time" includes periodic evaluation, checkpointing, and residual work.

The relatively small scoring share is consistent with the computation used by RELACE: teacher forcing evaluates tokens from existing trajectories without additional autoregressive action generation or environment interaction. In this profile, the hindsight pass takes approximately 14% of the generation time, making retrospective scoring a modest component of the measured training pipeline. These measurements quantify the recorded scoring stage; incremental end-to-end slowdown relative to GiGPO would require a matched run with hindsight scoring disabled. Timing aggregation details are provided in Appendix C.

D PROMPT TEMPLATES

For completeness and reproducibility, we provide the prompt templates used for policy interaction in ALFWorld and WebShop, together with the outcome-conditioned prompt used for retrospective likelihood evaluation. Variables enclosed in braces are instantiated from the current environment state and interaction history.

D.1 ALFWORLD AGENT PROMPT

Prompt Template for ALFWorld

You are an expert autonomous 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:

{available_actions}

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

D.2 WEBSHOP AGENT PROMPT

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 of the current situation are:

{available_actions}

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

D.3 OUTCOME-CONDITIONED HINDSIGHT PROMPT

RELACE performs retrospective scoring after the trajectory terminates. For each executed action $a_{i,t}$, we retain the original decision context $c_{i,t}$ and augment it with the observed terminal outcome

s_i^{final}. The executed action itself is not regenerated; instead, it is scored using teacher forcing under both contexts.

Hindsight Prompt Template for ALFWorld

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

Hindsight Prompt Template for WebShop

You are an expert autonomous agent operating in the WebShop e-commerce environment.
Your task is to: Navigate the website to reach the following exact target page state:
`[Target Page]:` {`final_obs`}
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:

{`available_actions`}

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

The previously executed action

$$a_{i,t} = (y_{i,t,1}, \ldots, y_{i,t,L_{i,t}})$$

is then teacher-forced under this hindsight context to compute $p_{i,t}^H$. The same action is teacher-forced under the original context to compute $p_{i,t}$.

E QUALITATIVE CASE STUDIES

We provide qualitative examples illustrating how RELACE assigns action-dependent retrospective credit. In each example, we report the original-context score $p_{i,t}$, outcome-conditioned score $p_{i,t}^H$, and the resulting trajectory-normalized relevance weight $\rho_{i,t}$. These examples are intended to illustrate the behavior of the estimator, rather than establish causal attribution.

E.1 ALFWORLD CASE STUDY

We visualize a successful ALFWorld trajectory to illustrate the multi-turn decision process in embodied environments. The agent receives only a terminal success signal, while intermediate actions must be assigned credit based on their contribution to completing the task.

ALFWorld Rollout: Step 1

Task:
`cool some potato and put it in diningtable`

Observation:
The agent is in the middle of a room. Available objects include a diningtable, fridge, cabinets, drawers, microwave, sinkbasin, and other kitchen objects.

Action:
`go to diningtable 1`

ALFWorld Rollout: Step 2

Observation:
The agent arrives at the dining table.
Objects on the table include:
`apple 3, bread 1, butterknife 3, egg 2, fork 1, knife 1, knife 2, mug 1, plate 1, potato 4, potato 2, tomato 3`

Action:
`take potato 2 from diningtable 1`

ALFWorld Rollout: Step 3

Observation:
The agent is holding:
`potato 2`
Available actions include moving to the fridge.

Action:
`go to fridge 1`

ALFWorld Rollout: Step 4

Observation:
The agent arrives at the fridge.
`The fridge 1 is closed.`

Action:
`cool potato 2 with fridge 1`

ALFWorld Rollout: Step 5

Observation:
The agent successfully cools the potato:
`You cool the potato 2 using the fridge 1.`

Action:
`go to diningtable 1`

ALFWorld Rollout: Step 6

Observation:
The agent returns to the dining table with the cooled potato.

Action:
`move potato 2 to diningtable 1`

Outcome:
`success`

E.2 WEBSHOP CASE STUDY

We show a successful WebShop trajectory to illustrate the multi-turn interaction structure used by our agent. The trajectory receives a single terminal success reward, while RELACE assigns credit at the action level.

WebShop Rollout: Step 1

Task:
Find me machine wash men's tuxedo shirts with polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: heather grey, fit type: youth, size: medium, and price lower than 30.00 dollars.

Observation:
```
Search
```

Action:
```
search[machine wash men's tuxedo shirts | polyester heathers |
heathers cotton | cotton heather | needle sleeve | classic fit |
color: heather grey | fit type: youth | size: medium | price <
30.00]
```

WebShop Rollout: Step 2

Observation:
```
Page 1 (Total results: 50)
B07JVVDJ6L -- Canadian Tuxedo T Shirt - Novelty Denim Tux-T-Shirt
$19.99
```

Action:
```
click[B07JVVDJ6L]
```

WebShop Rollout: Step 3

Observation:
Product page:
```
fit type: men, women, youth
color: royal blue, baby blue, heather grey, heather blue
small, medium, large, x-large, xx-large, 3x-large, 2t, 3t, 4t,
x-small
```

Action:
```
click[youth]
```

WebShop Rollout: Step 4

Observation:
Product page after selecting fit type:
```
fit type: youth
```
Available colors:
```
royal blue, baby blue, heather grey, heather blue
```

Action:
```
click[heather grey]
```

WebShop Rollout: Step 5

Observation:
Product page after selecting color:
```
color: heather grey
```
Available sizes:
```
small, medium, large, x-large, xx-large
```

Action:
```
click[medium]
```

WebShop Rollout: Step 6

Observation:
Final product configuration:

```
Product:  Canadian Tuxedo T Shirt - Novelty Denim Tux-T-Shirt
color:  heather grey
fit type:  youth
size:  medium
price:  $19.99
```

Action:

```
click[buy now]
```

Outcome:

```
Successful purchase
```