CROSS-ROLLOUT BELLMAN CLOSURE FOR LONG-HORIZON AGENTIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-based reinforcement learning such as GRPO has become a standard recipe for post-training LLM agents, replacing a learned critic with relative comparison among rollouts sampled for each task. In long-horizon settings, these rollouts revisit shared anchor states, offering cross-rollout evidence for step-level credit. Ideally, step-level credit should incorporate evidence beyond the realized suffixes observed at an anchor while aggregating alternative continuations according to their empirical frequencies. Existing estimators capture only one of these properties: visit-local averaging pools realized suffix returns at shared anchors and respects observed frequencies, but does not recursively propagate evidence across rollouts; whereas shortest-path estimators have global reach but allow a rarely observed route to dominate an anchor's value. Instead, we introduce a **C**ross-**R**ollout **B**ellman **C**losure (**CRBC**) method, which merges each rollout group into a finite empirical process with absorbing success and failure boundaries, and evaluates its behaviour-policy Bellman fixed point with one linear solve. This fixed point uses the same empirical action and transition frequencies to propagate evidence through shared anchors and aggregate alternative continuations. Backing up the resulting state values through observed transitions yields action values, whose gain over the corresponding state value provides step-level credit. A corresponding finite-depth family recovers visit-local return averaging at zero depth and converges to the exact closure as depth increases. The normalized closure credit is combined with the trajectory-level group advantage for policy optimization, without additional environment rollouts. Across ALFWorld, WebShop, and Sokoban benchmarks with multiple model scales, CRBC consistently improves final performance and learning efficiency. For example, CRBC outperforms the strongest baseline by 5.59 percentage points on ALFWorld with Qwen2.5-1.5B-Instruct, achieving a new state-of-the-art performance.

1 INTRODUCTION

Large Language Models (LLMs) (OpenAI, 2023; Yang et al., 2024; DeepSeek-AI, 2025) increasingly serve as interactive agents in embodied worlds, web interfaces, and tool-augmented workflows, where task completion requires planning across multiple interdependent decisions (Shridhar et al., 2021; Yao et al., 2022; Furuta et al., 2024; Schick et al., 2023). Reinforcement learning (RL) provides a post-training approach to improving these capabilities (Ziegler et al., 2020; Ouyang et al., 2022; Team, 2025). Group-based methods such as RLOO and GRPO replace PPO's learned critic (Schulman et al., 2017) with relative comparisons among same-task rollouts, avoiding the cost of training a separate value network (Ahmadian et al., 2024; Shao et al., 2024; Yu et al., 2025).

A single trajectory-level advantage, when broadcast to every step, cannot distinguish the contributions of individual actions (Wang et al., 2025; Jin et al., 2025; Chen et al., 2025). Yet rollouts sampled from the same task and initial condition often visit shared intermediate situations, or *anchors*, providing transition evidence beyond terminal outcomes. Step-level methods exploit this structure by comparing visits at shared anchors, with some further refining comparison groups by the consistency of preceding histories (Feng et al., 2025; Wang et al., 2026c; He et al., 2026). A further line pools transitions across rollouts into a merged graph and scores actions on it (Cheng et al., 2026; Wang et al., 2026d). Together, these approaches expose a reusable cross-rollout structure within each task group (Figure 1(a)).

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

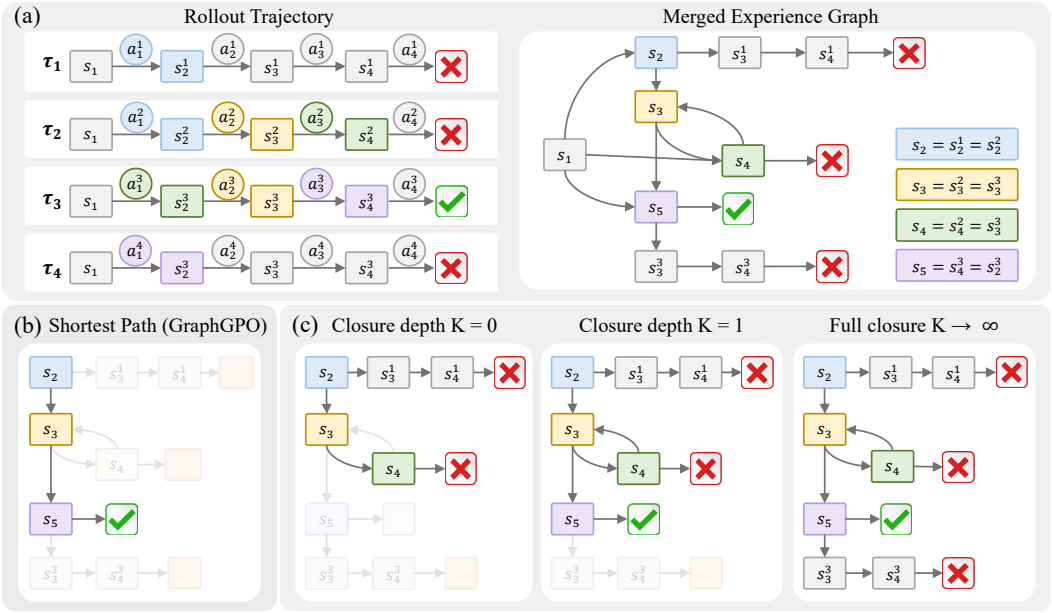


Figure 1: Reach and aggregation for step-level credit. For the rollout trajectory τ, node s denotes an *anchor*: an environment-relevant state representation that may be shared by multiple visits, and the edge represents an observed transition after executing semantic action a. (a) Shared anchors merge rollouts into a finite empirical process. (b) Shortest-path estimation aggregates by an extremum, ignoring transition frequencies; a rarely executed continuation can dominate anchor value. (c) The finite-depth family on this process, illustrated at s_2, uses behaviour-consistent expectations: $K = 0$ pools visit-local suffix returns; $K = 1$ adds one cross-rollout backup; and $K \to \infty$ evaluates all supported continuations at the behaviour-policy Bellman fixed point (CRBC), giving global reach.

Yet exposing this structure determines which observed transitions may be recombined, not how they should be evaluated. Two choices remain open: how far evidence may travel (*reach*), and how the evidence that arrives is combined (*aggregation*). Visit-local averaging (Feng et al., 2025) aggregates by an expectation over the observed behaviour, averaging realized suffix returns across visits to an anchor, but its reach is local: each occurrence is scored by its own continuation, so evidence pooled at shared successors is not recursively propagated back to the current anchor (Figure 1(c)). A shortest-path estimator (Cheng et al., 2026) has global reach over the merged graph, but aggregates by an extremum (Figure 1(b)): it values the shortest route to success without accounting for how frequently its transitions were observed, so a rarely observed continuation can dominate the estimate. Intuitively, an action's credit should account for both successful and failed continuations revealed by other rollouts at a shared successor. These limitations motivate combining global reach with behaviour-consistent aggregation on the group's empirical process, leading to our central question: *once a rollout group has been merged into an empirical process, how should step credit propagate through and aggregate its supported continuations?*

To address this issue, we introduce **C**ross-**R**ollout **B**ellman **C**losure (**CRBC**), which merges each rollout group into a finite empirical process over shared anchors and evaluates it at its behaviour-policy Bellman fixed point with one linear solve (Figure 2). Specifically, the Bellman closure repeatedly propagates successor values backward through the observed transitions until each anchor value is consistent with the values of its possible continuations. This closure is also the limit of Bellman backups initialized with visit-local suffix averages, allowing anchors to incorporate evidence from rollouts that never visited them. With success and failure as absorbing boundaries, this fixed point is the state value V, the exact discounted probability of reaching success under the group's own behaviour. Backing V up through the observed transitions gives the action value Q, and $Q - V$ measures how much an action improves on that behaviour at the same anchor. On this empirical process, feasible policy reweighting along the closure-credit direction does not decrease discounted success value (Proposition 2). We combine the normalized closure credit with the trajectory-level group advantage for critic-free policy optimization, without additional environment rollouts. Our main contributions are summarized as follows:

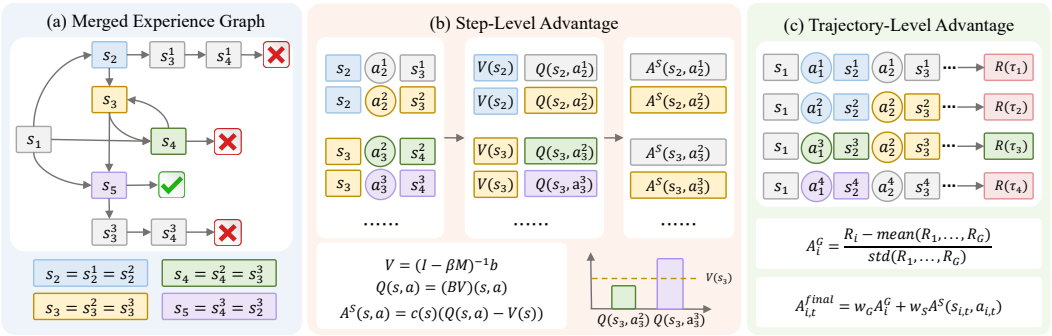


Figure 2: **Overview of CRBC.** (a) Shared anchors merge rollouts into an empirical process with absorbing success and failure boundaries. (b) A Bellman fixed-point solve yields the state value V, followed by action backups to obtain the action value Q and normalized $Q - V$ credit, which gives relative action credit. (c) Step-level credit and trajectory-level group-relative advantages are combined for policy optimization.

- We construct a finite empirical process from each rollout group's transition counts, allowing credit at an anchor to use downstream evidence from rollouts that never visited it. This evidence is weighted by observed action and transition frequencies.
- We introduce CRBC, which computes the group's behaviour-policy Bellman fixed point with one linear solve and scores each action by its gain over the state value, giving step credit with global reach and behaviour-consistent aggregation.
- We show that feasible policy reweighting along the closure-credit direction does not decrease discounted success value on the fixed empirical process (Proposition 2). Experiments on ALFWorld, WebShop, and Sokoban demonstrate gains over the evaluated baselines, including a 5.59-percentage-point improvement over the strongest baseline on ALFWorld with Qwen2.5-1.5B-Instruct.

2 Related Work

Reinforcement learning for LLMs and agents. LLM post-training uses human feedback (Ziegler et al., 2020; Stiennon et al., 2022; Ouyang et al., 2022), preference optimization (Rafailov et al., 2024), and verifiable rewards (DeepSeek-AI, 2025; Team, 2025). Group-based methods replace PPO's critic (Schulman et al., 2017) with comparisons among same-prompt responses, including RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), Dr. GRPO, DAPO, and CPPO (Liu et al., 2025; Yu et al., 2025; Lin et al., 2025). Agent training progressed from DQN in text games (Mnih et al., 2015; Narasimhan et al., 2015) to PPO- and AWR-style updates (Peng et al., 2019) for embodied and device-control tasks (Zhai et al., 2024; Bai et al., 2024). ArCHer (Zhou et al., 2024) learns hierarchical values for tasks such as WebShop (Yao et al., 2022), while Agent Q (Putta et al., 2024) uses Monte Carlo tree search (Silver et al., 2017). LOOP (Chen et al., 2025) combines leave-one-out estimation with PPO-style updates on AppWorld (Trivedi et al., 2024), and RAGEN (Wang et al., 2025) studies multi-turn trajectory optimization. Sparse rewards in long-horizon tasks such as ALFWorld (Shridhar et al., 2021) motivate finer credit assignment.

Step-level credit assignment in agentic RL. Step-level credit can use process reward models (Lightman et al., 2024), LLM judges (Wang et al., 2026b), learned action-boundary critics (Wang et al., 2026a), or rollout-group statistics. GiGPO (Feng et al., 2025) compares repeated anchor visits, HGPO (He et al., 2026) refines groups by history consistency, and RTMC (Wang et al., 2026c) averages returns over rollout trees. GraphGPO (Cheng et al., 2026) scores merged graphs by shortest-path distance to success, while G2PO (Wang et al., 2026d) derives edge credit from one-step TD errors. Other methods calibrate credit under limited rollout evidence (Li et al., 2026), add progress signals for all-failing groups (Yang et al., 2026), derive credit from state potentials (Fan & Liu, 2026), or extract attribution from model reasoning (Tan et al., 2026). Unlike return averaging, shortest-path scoring, or one-step TD credit, CRBC explicitly solves for the exact behavior-policy

Bellman fixed point of the group's empirical process, using observed action and transition frequencies for cross-rollout propagation and aggregation.

3 PRELIMINARIES

Rollout groups and anchors. We consider an LLM agent with policy π_θ, parameterized by θ, solving a task g through G rollouts from the same initial condition. At step t of rollout i, it observes a language context $h_{i,t}$, generates a response $y_{i,t} \sim \pi_\theta(\cdot \mid h_{i,t})$, and executes the semantic action $a_{i,t} = \mathrm{Exec}(y_{i,t})$, after which the environment returns $h_{i,t+1}$. Rollout i ends after $T_i \le T_{\max}$ steps with a binary success indicator $R_i \in \{0, 1\}$ and no intermediate reward. An environment-aware map $\phi : \mathcal{H} \to \mathcal{S}$ from language contexts to anchors defines $s_{i,t} = \phi(h_{i,t})$, with anchors matched only within each task group (Feng et al., 2025; Cheng et al., 2026). We represent the group by

$$\mathcal{D}_g = \{\tau_i\}_{i=1}^{G}, \qquad \tau_i = \big((s_{i,t}, a_{i,t}, x_{i,t})_{t=1}^{T_i},\, R_i\big), \tag{1}$$

where $x_{i,t} = s_{i,t+1}$ for $t < T_i$, and the terminal successor x_{i,T_i} is the absorbing success boundary z^+ if $R_i = 1$, or the absorbing failure boundary z^- otherwise. We write $\mathcal{S}_g$ for the observed non-terminal anchors, $\mathcal{A}_g(s)$ for the actions observed at s, and $\mathcal{X}_g = \mathcal{S}_g \cup \{z^+, z^-\}$. Contexts and responses are retained for policy optimization. Time-limit truncations are treated as failures in this representation.

Group-based RL. Let $A_{i,t}$ denote the advantage assigned to environment step (i, t). The optimizer maximizes a clipped surrogate with a reference-policy penalty,

$$\mathcal{J}(\theta) = \mathbb{E}\Big[\min\big(\rho_{i,t}(\theta)A_{i,t},\ \mathrm{clip}(\rho_{i,t}(\theta), 1 \pm \epsilon)A_{i,t}\big)\Big] - \lambda_{\mathrm{KL}}\, \mathbb{D}_{\mathrm{KL}}[\pi_\theta \,\|\, \pi_{\mathrm{ref}}]\,. \tag{2}$$

Here $\mathbb{E}$ averages uniformly over sampled groups and, within each group, over environment steps. ϵ sets the clipping range, and λ_{KL} weights the KL penalty against the fixed reference policy π_{ref}. The importance ratio is $\rho_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid h_{i,t})/\pi_{\theta_{\mathrm{old}}}(y_{i,t} \mid h_{i,t})$, where $\pi_{\theta_{\mathrm{old}}}$ is the rollout policy. Group-relative methods form the advantage by comparing outcomes within the group rather than against a learned value function; GRPO (Shao et al., 2024) uses

$$A_i^{\mathrm{G}} = \frac{R_i - \mathrm{mean}(R_1, \ldots, R_G)}{\mathrm{std}(R_1, \ldots, R_G)}, \tag{3}$$

and RLOO (Ahmadian et al., 2024) a leave-one-out variant. Ours is agnostic to the choice, and we denote by A_i^{G} whatever the underlying optimizer outputs. Setting $A_{i,t} = A_i^{\mathrm{G}}$ for every step of τ_i gives one scalar per trajectory, broadcast uniformly, which cannot distinguish the actions inside it.

Step-level credit. Step-level methods exploit shared anchors to replace uniform trajectory-level credit with signals that vary within a trajectory. Let $\mathcal{I}_g(s) := \{(i, t) : s_{i,t} = s\}$ collect occurrences of anchor s, and let $G_{i,t} = \beta^{T_i - t + 1} R_i$ denote the realized discounted return, with discount $\beta \in (0, 1)$. The step-level advantage is

$$A_{i,t}^{step} = \frac{G_{i,t} - \mathrm{mean}\{G_{i',t'} : (i', t') \in \mathcal{I}_g(s_{i,t})\}}{\mathrm{std}\{G_{i',t'} : (i', t') \in \mathcal{I}_g(s_{i,t})\}}, \tag{4}$$

combined with A_i^{G} by a weighted sum (He et al., 2026; Feng et al., 2025). Here equation 4 scores each occurrence by its own realized suffix return, so evidence from other rollouts at shared successors never propagates back to s.

4 CROSS-ROLLOUT BELLMAN CLOSURE

To propagate and aggregate evidence from the observed continuations of rollouts that share anchors, this section constructs a finite empirical process from a rollout group (Section 4.1), computes its Bellman closure (Section 4.2), analyses credit alignment across propagation depths (Section 4.3), and integrates closure credit into policy optimization (Section 4.4), as summarized in Figure 2. Throughout the section and its proofs, P_g, μ_g, and all associated Bellman operators, values, and credits refer to the process induced by group g.

4.1 THE EMPIRICAL PROCESS

We first turn the rollouts in a group into a common empirical process over shared anchors. This representation preserves the observed action and transition frequencies needed to propagate and aggregate cross-rollout evidence.

Empirical model. Let $N_g(s,a,x)$ count occurrences of transition (s,a,x) in $\mathcal{D}_g$, with marginals $N_g(s,a) = \sum_{x \in \mathcal{X}_g} N_g(s,a,x)$ and $N_g(s) = \sum_{a \in \mathcal{A}_g(s)} N_g(s,a)$. The group induces an empirical transition kernel and an empirical behaviour policy over anchors,

$$P_g(x \mid s,a) := \frac{N_g(s,a,x)}{N_g(s,a)}, \qquad \mu_g(a \mid s) := \frac{N_g(s,a)}{N_g(s)}. \tag{5}$$

The unsmoothed frequencies in equation 5 define the group-induced process, with zero probability assigned to unobserved transitions. Our analysis concerns this empirical process rather than the unknown environment dynamics.

The visit-local estimator. Let $\mathcal{I}_g(s,a) := \{(i,t) \in \mathcal{I}_g(s) : a_{i,t} = a\}$ collect occurrences of (s,a) within group g, so that $|\mathcal{I}_g(s,a)| = N_g(s,a)$. Using the realized discounted returns $G_{i,t}$ defined in Section 3, we define the visit-local estimates

$$Q_g^{(0)}(s,a) := \frac{1}{|\mathcal{I}_g(s,a)|} \sum_{(i,t) \in \mathcal{I}_g(s,a)} G_{i,t}, \qquad V_g^{(0)}(s) := \sum_{a \in \mathcal{A}_g(s)} \mu_g(a \mid s)\, Q_g^{(0)}(s,a). \tag{6}$$

Weighting these action values by their empirical action frequencies makes $V_g^{(0)}(s)$ equal to the mean return over all visits to s. Each occurrence contributes only the discounted outcome of its own complete continuation to termination.

4.2 THE BELLMAN CLOSURE

Given the empirical process, we now evaluate how success evidence propagates through its observed transitions. We achieve this by defining Bellman backups and their fixed point, which yields behaviour-consistent state and action values.

Backup operators. Let $\mathcal{V}_g$ denote the value functions $V : \mathcal{X}_g \to [0,1]$ with $V(z^+) = 1$ and $V(z^-) = 0$. The action backup averages discounted successor values under the empirical transition kernel, and the state backup averages these action values under the empirical behaviour policy:

$$\begin{aligned} (\mathcal{B}_g V)(s,a) &:= \beta \sum_{x \in \mathcal{X}_g} P_g(x \mid s,a)\, V(x), && s \in \mathcal{S}_g,\ a \in \mathcal{A}_g(s), \\ (\mathcal{T}_g V)(s) &:= \sum_{a \in \mathcal{A}_g(s)} \mu_g(a \mid s)\, (\mathcal{B}_g V)(s,a), && s \in \mathcal{S}_g. \end{aligned} \tag{7}$$

The state backup keeps both boundary values fixed. Each application of $\mathcal{T}_g$ performs one round of cross-rollout recombination: evidence at successor anchors propagates back to s across one additional observed edge, including evidence from rollouts that did not visit s. By contrast, equation 6 pools complete realized suffix returns without recombining continuations across rollouts. Repeated backups extend this propagation along longer observed paths.

The behaviour-policy fixed point. We define the closure of this propagation as a value assignment unchanged by further Bellman backups. Each anchor's value then equals the empirical average of its discounted successor values, yielding the fixed-point condition $\mathcal{T}_g V_g = V_g$. We express this condition as a linear system by collecting the transitions among non-terminal anchors and the one-step contribution from the success boundary into

$$M_g(s,s') := \sum_{a \in \mathcal{A}_g(s)} \mu_g(a \mid s) P_g(s' \mid s,a), \qquad b_g(s) := \beta \sum_{a \in \mathcal{A}_g(s)} \mu_g(a \mid s) P_g(z^+ \mid s,a), \tag{8}$$

where $s, s' \in \mathcal{S}_g$. Restricting values to these non-terminal anchors, the state backup is affine, $\mathcal{T}_g V = \beta M_g V + b_g$. The matrix M_g is substochastic: its row sums are at most one because probability mass

may leave to the absorbing boundaries. The fixed-point condition therefore becomes $(I-\beta M_g)V_g = b_g$, so well-posedness reduces to the invertibility of $I - \beta M_g$.

Proposition 1 (Well-posedness). *For any group and any $\beta \in (0,1)$, $\mathcal{T}_g$ is a β-contraction on $\mathcal{V}_g$ in $\|\cdot\|_\infty$ and admits a unique fixed point $V_g \in \mathcal{V}_g$. Bellman iteration converges to V_g from every initializer in $\mathcal{V}_g$, and $I - \beta M_g$ is invertible.*

The closure is therefore independent of the initializer and has the closed form

$$V_g = (I - \beta M_g)^{-1} b_g, \qquad Q_g := \mathcal{B}_g V_g. \tag{9}$$

In practice, we obtain V_g by solving $(I - \beta M_g)V_g = b_g$ directly, without forming the inverse or iterating $\mathcal{T}_g$ (Appendix B.1). With no intermediate rewards and boundary values $V(z^+) = 1$ and $V(z^-) = 0$, $V_g(s)$ gives the *exact* β-discounted probability of reaching success from s under μ_g on the empirical process induced by the current group.

4.3 Finite depth and the reference direction

We next relate visit-local estimation to the full closure through a finite-depth family. We then characterize when feasible policy reweighting using these credits does not decrease discounted success value on the fixed empirical process.

The finite-depth family. We initialize Bellman iteration at the visit-local estimator in equation 6, with the same fixed boundary values. For integers $K \geq 1$,

$$V_g^{(K)} := \mathcal{T}_g^K V_g^{(0)}, \qquad Q_g^{(K)} := \mathcal{B}_g V_g^{(K-1)}, \qquad A_g^{(K)} := Q_g^{(K)} - V_g^{(K)}, \tag{10}$$

with $A_g^{(0)} := Q_g^{(0)} - V_g^{(0)}$. As $K \to \infty$, the value estimates converge to V_g and Q_g in equation 9, and the limiting credit is denoted by $A_g^{(\infty)} := Q_g - V_g$. Here K counts rounds of recombination rather than steps of lookahead, since equation 6 already carries complete realized suffixes at $K = 0$. All members remain centred under μ_g at each anchor, and varying K changes reach while holding the empirical process and aggregation fixed. The $K = 0$ member uses visit-local suffix averaging (Feng et al., 2025), whereas shortest-path estimators (Cheng et al., 2026) lie outside this family because they aggregate by an extremum rather than an expectation.

On the fixed empirical process, let V_g^ν denote the β-discounted success value under a policy ν on the observed action support, and let M_g^ν be obtained from M_g in equation 8 by replacing μ_g with ν. In particular, $V_g^{\mu_g} = V_g$.

Proposition 2 (The closure credit as a reference direction). *Let $\tilde{A}$ be centred under μ_g at each anchor, let $w : \mathcal{S}_g \to [0,\infty)$, and define $\xi(s,a) := \mu_g(a \mid s) w(s) \tilde{A}(s,a)$. For any $\alpha \geq 0$ such that $\nu_\alpha := \mu_g + \alpha\xi$ is a valid policy,*

$$V_g^{\nu_\alpha} - V_g = \alpha\big(I - \beta M_g^{\nu_\alpha}\big)^{-1}\Big(w \odot \big\langle \tilde{A}, A_g^{(\infty)} \big\rangle_{\mu_g}\Big), \tag{11}$$

where $\langle f, h \rangle_{\mu_g}(s) := \sum_a \mu_g(a \mid s) f(s,a) h(s,a)$ and $\odot$ denotes componentwise multiplication. For $\tilde{A} = A_g^{(\infty)}$, the inner product is the anchor-wise variance $\sigma_g(s)^2 := \sum_a \mu_g(a \mid s)[A_g^{(\infty)}(s,a)]^2 \geq 0$, giving $V_g^{\nu_\alpha} \succeq V_g$ componentwise.

Finite-depth credits inherit this guarantee when their deviation from the closure is sufficiently small.

Corollary 3 (Finite-depth alignment). *For $K \geq 1$, let $\bar{\delta}_g^{(0)} := \mathcal{T}_g V_g^{(0)} - V_g^{(0)}$ be the empirical Bellman residual on non-terminal anchors. Then*

$$\big\|A_g^{(\infty)} - A_g^{(K)}\big\|_\infty \leq \frac{2\beta^K}{1-\beta}\big\|\bar{\delta}_g^{(0)}\big\|_\infty. \tag{12}$$

Under the assumptions of Proposition 2, taking $\tilde{A} = A_g^{(K)}$ gives $V_g^{\nu_\alpha} \succeq V_g$ whenever the right-hand side of equation 12 does not exceed $\sigma_g(s)$ at every anchor with $w(s) > 0$ and $\sigma_g(s) > 0$; anchors with $\sigma_g(s) = 0$ contribute a zero inner product in equation 11.

Together, Proposition 2 and Corollary 3 establish closure credit as a reference direction on the fixed empirical process: feasible policy reweighting along this direction does not decrease discounted success value, and sufficiently accurate finite-depth credits inherit the same guarantee. Proofs are provided in Appendix B, with further analysis of Bellman residuals and finite-depth errors in Appendix B.3.

Table 1: Performance comparison on ALFWorld and WebShop. For ALFWorld, we report the average success rate (%) for each subtask and the overall success rate. For WebShop, we report the average task score and average success rate (%). Most results are averaged over three random seeds. The best results are highlighted in **bold**.

		ALFWorld							WebShop	
Type	**Method**	Pick	Clean	Cool	Look	Heat	Pick2	All	Score	Succ.
Closed-Source Models										
Prompting	GPT-4o	75.30	60.80	31.20	56.70	21.60	49.80	48.00	31.80	23.70
Prompting	Gemini-2.5-Pro	92.80	63.30	62.10	69.00	26.60	58.70	60.30	42.50	35.90
Qwen2.5-1.5B-Instruct										
Prompting	Qwen2.5	5.90	5.50	3.30	9.70	4.20	0.00	4.10	23.10	5.20
Prompting	ReAct	17.40	20.50	15.70	6.20	7.70	2.00	12.80	40.10	11.30
Prompting	Reflexion	35.30	22.20	21.70	13.60	19.40	3.70	21.80	55.80	21.90
RL Training	PPO	$64.80_{(3.50)}$	$40.50_{(6.90)}$	$57.10_{(4.90)}$	$60.60_{(6.60)}$	$46.40_{(4.00)}$	$47.40_{(1.90)}$	$54.30_{(3.10)}$	$73.80_{(3.00)}$	$51.50_{(2.90)}$
RL Training	RLOO	$88.30_{(3.00)}$	$52.80_{(8.60)}$	$71.00_{(5.90)}$	$62.80_{(8.70)}$	$66.40_{(5.50)}$	$56.90_{(4.70)}$	$69.70_{(2.50)}$	$73.90_{(5.60)}$	$52.10_{(6.70)}$
RL Training	GRPO	$85.27_{(1.40)}$	$65.79_{(2.63)}$	$88.08_{(8.08)}$	$58.33_{(0.00)}$	$78.57_{(7.14)}$	$75.00_{(5.00)}$	$77.73_{(1.95)}$	$83.33_{(3.52)}$	$71.09_{(0.78)}$
RL Training	GiGPO	$96.67_{(4.71)}$	$84.21_{(4.30)}$	$88.21_{(3.02)}$	$72.22_{(3.93)}$	$95.24_{(3.89)}$	$88.33_{(2.36)}$	$89.32_{(1.95)}$	$88.93_{(1.55)}$	$75.78_{(2.21)}$
RL Training	GraphGPO	$95.63_{(3.09)}$	$84.21_{(4.30)}$	$88.31_{(0.22)}$	$88.89_{(3.93)}$	$92.06_{(5.94)}$	$88.33_{(2.36)}$	$90.10_{(1.84)}$	$89.60_{(1.46)}$	$79.69_{(1.28)}$
RL Training	HGPO	$94.30_{(2.90)}$	$69.10_{(9.20)}$	$\mathbf{95.90}_{(2.80)}$	$\mathbf{97.40}_{(3.60)}$	$92.00_{(2.30)}$	$82.30_{(3.20)}$	$90.50_{(0.40)}$	$85.50_{(0.50)}$	$70.50_{(1.70)}$
RL Training	**CRBC (Ours)**	$\mathbf{100.00}_{(0.00)}$	$\mathbf{92.98}_{(6.56)}$	$92.15_{(3.33)}$	$91.67_{(0.00)}$	$\mathbf{96.83}_{(4.49)}$	$\mathbf{100.00}_{(0.00)}$	$\mathbf{96.09}_{(0.64)}$	$\mathbf{90.11}_{(1.24)}$	$\mathbf{81.51}_{(0.74)}$
Qwen2.5-7B-Instruct										
Prompting	Qwen2.5	33.40	21.60	19.30	6.90	2.80	3.20	14.80	26.40	7.80
Prompting	ReAct	48.50	35.40	34.30	13.20	18.20	17.60	31.20	46.20	19.50
Prompting	Reflexion	62.00	41.60	44.90	30.90	36.30	23.80	42.70	58.10	28.80
RL Training	PPO	$92.30_{(4.00)}$	$64.00_{(8.40)}$	$92.50_{(2.40)}$	$89.50_{(7.00)}$	$80.30_{(2.00)}$	$68.80_{(8.30)}$	$80.40_{(2.70)}$	$81.40_{(3.10)}$	$68.70_{(5.10)}$
RL Training	RLOO	$87.60_{(4.30)}$	$78.20_{(8.30)}$	$87.30_{(5.80)}$	$81.30_{(7.60)}$	$71.90_{(5.20)}$	$48.90_{(8.40)}$	$75.50_{(4.60)}$	$80.30_{(3.20)}$	$65.70_{(4.00)}$
RL Training	GRPO	$88.98_{(5.30)}$	$91.98_{(4.43)}$	$77.89_{(4.58)}$	$78.57_{(0.00)}$	$\mathbf{90.74}_{(5.24)}$	$71.43_{(3.89)}$	$83.33_{(2.05)}$	$84.31_{(1.27)}$	$75.00_{(2.78)}$
RL Training	GiGPO	$\mathbf{100.00}_{(0.00)}$	$92.11_{(7.89)}$	$80.77_{(7.69)}$	$95.83_{(4.17)}$	$90.48_{(0.00)}$	$90.00_{(5.00)}$	$91.41_{(2.34)}$	$86.60_{(1.17)}$	$76.56_{(1.56)}$
RL Training	GraphGPO	$97.85_{(3.04)}$	$89.47_{(7.44)}$	$88.31_{(0.22)}$	$94.44_{(3.93)}$	$82.54_{(8.09)}$	$86.67_{(6.24)}$	$90.10_{(3.51)}$	$87.24_{(1.57)}$	$78.13_{(0.00)}$
RL Training	**CRBC (Ours)**	$\mathbf{100.00}_{(0.00)}$	$\mathbf{100.00}_{(0.00)}$	$\mathbf{94.23}_{(5.77)}$	$\mathbf{100.00}_{(0.00)}$	$90.48_{(0.00)}$	$\mathbf{97.50}_{(2.50)}$	$\mathbf{96.88}_{(1.56)}$	$\mathbf{90.43}_{(1.25)}$	$\mathbf{81.51}_{(3.51)}$

4.4 Step credit and policy optimization

Finally, we convert closure values into normalized step-level credit and combine it with the trajectory-level group advantage for critic-free policy optimization.

Step-level advantage. For CRBC, we gate closure credit on anchors with at least two distinct observed actions and normalise it by the anchor-wise standard deviation (Figure 2(b)):

$$A_g^{\mathrm{S}}(s,a) := \frac{\mathbf{1}\{|\mathcal{A}_g(s)| \geq 2\}}{\max\{\sigma_g(s),\, \sigma_{\min}\}}\big(Q_g(s,a) - V_g(s)\big). \tag{13}$$

Here $\sigma_g(s)$ is defined in Proposition 2, and $\sigma_{\min} > 0$ guards nearly flat rows. Denote the non-negative prefactor by $c_g(s)$; the same proposition then applies with $w = c_g$, preserving alignment. All occurrences of the same (anchor, semantic action) pair share this credit.

Policy optimization. CRBC retains the trajectory-level advantage A_i^{G} (Figure 2(c)) so that steps without an identifiable anchor comparison remain covered. The advantage substituted into equation 2 at step (i,t) is

$$A_{i,t}^{\mathrm{final}} = w_{\mathrm{G}} A_i^{\mathrm{G}} + w_{\mathrm{S}} A_g^{\mathrm{S}}(s_{i,t}, a_{i,t}), \tag{14}$$

with weights $w_{\mathrm{G}}, w_{\mathrm{S}} \geq 0$, giving the CRBC objective

$$\mathcal{J}_{\mathrm{CRBC}}(\theta) = \mathbb{E}\Bigg[\frac{1}{\sum_i T_i}\sum_{i,t}\min\big(\rho_{i,t}A_{i,t}^{\mathrm{final}},\ \mathrm{clip}(\rho_{i,t}, 1\pm\epsilon)A_{i,t}^{\mathrm{final}}\big)\Bigg] - \lambda_{\mathrm{KL}}\mathbb{D}_{\mathrm{KL}}\big[\pi_\theta \| \pi_{\mathrm{ref}}\big], \tag{15}$$

where $\sum_i T_i$ is the total number of environment steps in the group, and the remaining notation ($\rho_{i,t}$, ϵ, λ_{KL}) is that of equation 2. Only the advantage differs from equation 2, so CRBC introduces no parametric critic and no additional rollouts. The advantage and the empirical process behind it receive no gradient.

5 Experiments

Our experiments answer three questions: (1) does cross-rollout Bellman closure improve LLM-agent training over trajectory-level and step-level baselines; (2) how does it behave over the course

Table 2: Closure-depth ablation on ALFWorld with Qwen2.5-1.5B-Instruct. Only K varies; all other settings are fixed. Best values are in **bold**.

Depth K	Step-credit	Score	Success (%)
0	$A^{(0)}$	$7.00_{(0.27)}$	$92.97_{(0.00)}$
1	$A^{(1)}$	$7.05_{(0.53)}$	$92.97_{(1.56)}$
2	$A^{(2)}$	$7.16_{(1.20)}$	$93.75_{(3.12)}$
3	$A^{(3)}$	$7.21_{(0.13)}$	$93.75_{(0.78)}$
∞	$A^{(\infty)}$	$\mathbf{8.01}_{(0.27)}$	$\mathbf{96.09}_{(0.64)}$

Table 3: Test performance on 6×6 Sokoban with Qwen2.5-VL-3B-Instruct. Results are averaged over three seeds; best values are in **bold**.

Type	Method	Score	Success (%)
Prompting	Qwen2.5-VL*	—	11.70
RL Training	GRPO†	$1.81_{(2.50)}$	$46.48_{(3.64)}$
RL Training	GiGPO	$4.00_{(1.12)}$	$76.56_{(5.47)}$
RL Training	GraphGPO	$4.13_{(0.34)}$	$77.60_{(2.88)}$
RL Training	**CRBC (Ours)**	$\mathbf{5.06}_{(0.54)}$	$\mathbf{82.81}_{(2.30)}$

of training; and (3) how do its two design choices (combining step- and trajectory-level credit, and taking the closure depth to $K \to \infty$) affect performance.

5.1 Experiment Setup

Benchmarks and baselines. We evaluate CRBC on text-based **ALFWorld** (Shridhar et al., 2021) and **WebShop** (Yao et al., 2022) for household interaction and web shopping, respectively, and on visual 6×6 **Sokoban** (Schrader, 2018). Baselines include closed-source models GPT-4o (OpenAI, 2023) and Gemini-2.5-Pro (Team, 2023); prompting baselines Qwen2.5 (Qwen et al., 2025), ReAct (Yao et al., 2023), and Reflexion (Shinn et al., 2023); and RL methods PPO (Schulman et al., 2017), RLOO (Ahmadian et al., 2024), GRPO (Shao et al., 2024), GiGPO (Feng et al., 2025), GraphGPO (Cheng et al., 2026), and HGPO (He et al., 2026). For Sokoban, group-based RL baselines share the same VLM backbone. Benchmark details and baseline descriptions, including implementation provenance, appear in Appendices C.2 and C.1, respectively.

Implementation details. We use Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Qwen et al., 2025) for ALFWorld and WebShop, and Qwen2.5-VL-3B-Instruct (Bai et al., 2025) for Sokoban. Within each benchmark and model scale, reproduced RL methods share the same data, rollout, and optimization protocol, differing only in the advantage estimator and its method-specific settings. Each update uses $G = 8$ rollouts per group, with 16 groups for ALFWorld/WebShop and 32 for Sokoban, and episodes capped at 50, 15, and 15 environment steps, respectively. Rollout and validation temperatures are 1.0 and 0.4, the actor learning rate is 10^{-6}, and the KL coefficient is 0.01; we train for 150 updates and validate every five. All local runs use one node of eight 80 GB NVIDIA A100 GPUs. Unless stated otherwise, CRBC uses $\beta = 0.98$, full closure ($K \to \infty$), and $(w_{\mathrm{G}}, w_{\mathrm{S}}) = (1, 5)$ without auxiliary predictors. Full configurations appear in Appendix C.3.

5.2 Experimental Results

Performance on agentic benchmarks. Table 1 reports the comparison between CRBC and the baselines on ALFWorld and WebShop, and Table 3 reports the results on Sokoban. Across both model sizes and all three environments, CRBC consistently outperforms every baseline in overall success. On ALFWorld with Qwen2.5-1.5B-Instruct, CRBC reaches an overall success rate of 96.09%, improving over GRPO by 18.36 points and over the strongest baseline (HGPO, 90.50%) by 5.59 points; it improves over GRPO on every subtask, ranks first on four of the six, and solves Pick and Pick2 perfectly. On WebShop, CRBC attains the best task score (90.11) and success rate (81.51%). The advantage persists at the 7B scale, where CRBC achieves 96.88% on ALFWorld and 81.51% success on WebShop, surpassing all baselines on both overall metrics. On the vision-language Sokoban benchmark (Table 3), CRBC reaches 82.81% success, exceeding GraphGPO by 5.21 points, GiGPO by 6.25 points, and GRPO by a large margin. These results show that recovering high-fidelity step-level credit through cross-rollout Bellman closure translates into consistent end-task gains across embodied, web, and visual multi-turn settings.

Training dynamics. Figure 3 shows mean trajectory reward for CRBC, GraphGPO, GiGPO, and GRPO, with validation success rates in Figure 6 (appendix). Across the three environments, step-level methods improve faster and attain higher rewards than trajectory-level GRPO. CRBC rises fastest during early and middle training and achieves the highest final reward; on ALFWorld with Qwen2.5-1.5B-Instruct, it matches GraphGPO's final reward in roughly half the updates. Simi-

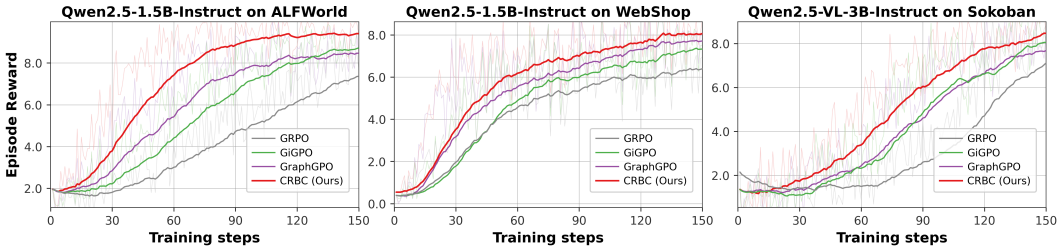


Figure 3: Training reward curves for CRBC (Ours, red), GraphGPO (blue), GiGPO (green), and GRPO (gray) on ALFWorld, WebShop, and Sokoban. Light and dark curves denote per-update and EMA-smoothed rewards ($\alpha = 0.95$), respectively. Qwen2.5-1.5B-Instruct is used for ALFWorld/WebShop and Qwen2.5-VL-3B-Instruct for Sokoban; panel scales are independent.

lar validation trends show improved held-out success alongside the reward gains, supporting the learning-efficiency benefit of cross-rollout Bellman closure.

5.3 Ablation Study

Closure depth. Table 2 varies only the closure depth K, keeping the aggregation rule, anchor map ϕ, and other settings fixed. Figure 7 shows that reward, validation success, and validation score generally improve faster as K increases, with $K = 3$ and full closure showing the strongest early trends. Final success rates are 92.97% for $K = 0, 1$, 93.75% for $K = 2, 3$, and 96.09% for full closure, a 3.12-point gain over the visit-local estimator; validation score also rises from 7.00 at $K = 0$ to 8.01 under full closure. Figure 5 provides a structural explanation using rollouts from an early-training checkpoint: the proportion of eligible visits with access to all eight rollouts increases from 12.9% at $K = 0$ to 61.5% under full closure, while the paired heatmap shows expanded coverage for 63.0% of eligible visits. These results support the benefit of deeper Bellman propagation and illustrate how closure makes additional cross-rollout evidence accessible beyond the current anchor.

Computational overhead. Figure 4 presents the per-update runtime breakdown of CRBC on ALFWorld with Qwen2.5-1.5B-Instruct, averaged over 120 non-validation updates from a profiling run. An update takes $187.46\,\mathrm{s}$ on average, dominated by rollout ($154.99\,\mathrm{s}$) and the policy update ($20.35\,\mathrm{s}$). The trajectory-level group advantage A^{G} and the instrumented CRBC step-credit core A^{S} take only $0.044\,\mathrm{s}$ and $0.057\,\mathrm{s}$, respectively, corresponding to 0.02% and 0.03% of the total update time. Including tensor preparation and advantage assembly, the complete advantage block takes $0.330\,\mathrm{s}$, or 0.18% of the update. Since CRBC reuses the same rollout and actor-update pipeline without introducing additional model rollouts or a critic network, its credit construction remains negligible relative to the main training stages.

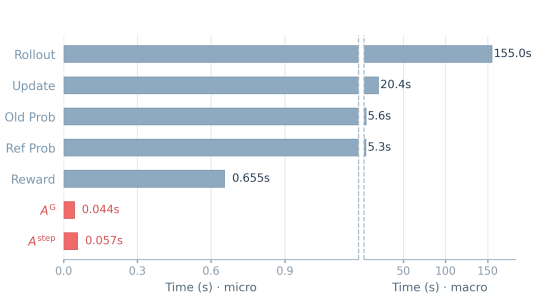


Figure 4: Per-update runtime breakdown of CRBC on ALFWorld with Qwen2.5-1.5B-Instruct. Blue bars denote shared stages, while red bars denote trajectory-level and step-level credit construction.

6 Conclusion

We presented CRBC, which evaluates each rollout group's empirical process at its behavior-policy Bellman fixed point to derive critic-free $Q - V$ credit with global cross-rollout propagation and behavior-consistent aggregation. Feasible policy reweighting along this credit does not decrease discounted success value on the fixed process. Combined with trajectory-level advantages, the normalized credit improves performance and learning efficiency on ALFWorld, WebShop, and Sokoban without additional rollouts, supported by depth and credit ablations.

AI USE STATEMENT

We used AI tools to assist with language polishing and improve the readability of the manuscript. The authors reviewed all revisions and take full responsibility for the final content.

ETHICS STATEMENT

CRBC is studied in controlled text-based and visual environments using ALFWorld, WebShop, and Sokoban. Actions are restricted to these benchmark interfaces and do not initiate actual purchases or affect physical systems. Our experimental protocol uses existing benchmark tasks without conducting human-subject studies or gathering additional personal information.

REPRODUCIBILITY STATEMENT

The empirical model, Bellman closure, and policy objective are specified in Section 4, with the training workflow given in Algorithm 1 and supporting derivations in Appendix B. Section 5 presents the evaluation protocol, while Appendix C documents environment settings, anchor construction, method-specific hyperparameters, and hardware configurations. The prompts used for agent interaction appear in Appendix E. We report means and standard deviations across independent runs, with the main CRBC results averaged over three random seeds.

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

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning, 2024. URL https://arxiv.org/abs/2406.11896.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive llm agents, 2025. URL https://arxiv.org/abs/2502.01600.

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, 2026. URL https://arxiv.org/abs/2605.26684.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025. doi: 10.48550/ARXIV.2501.12948. URL https://doi.org/10.48550/arXiv.2501.12948.

Mingxuan Fan and Peiyang Liu. Progress- and reliability-oriented group policy optimization for agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2607.04242.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training, 2025. URL https://arxiv.org/abs/2505.10978.

Hiroki Furuta, Kuang-Huei Lee, Ofir Nachum, Yutaka Matsuo, Aleksandra Faust, Shixiang Shane Gu, and Izzeddin Gur. Multimodal web navigation with instruction-finetuned foundation models, 2024. URL https://arxiv.org/abs/2305.11854.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks, 2026. URL https://arxiv.org/abs/2602.22817.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.09516.

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, 2026. URL https://arxiv.org/abs/2606.05885.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models, 2025. URL https://arxiv.org/abs/2503.22342.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective, 2025. URL https://arxiv.org/abs/2503.20783.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1–11, 2015.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning, 2019. URL https://arxiv.org/abs/1910.00177.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. Agent q: Advanced reasoning and learning for autonomous ai agents, 2024. URL https://arxiv.org/abs/2408.07199.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

Max-Philipp B. Schrader. gym-sokoban. https://github.com/mpSchrader/gym-sokoban, 2018.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. 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, 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, 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, 2021. URL https://arxiv.org/abs/2010.03768.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback, 2022. URL https://arxiv.org/abs/2009.01325.

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, 2026. URL https://arxiv.org/abs/2603.08754.

Gemini Team. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805, 2023. doi: 10.48550/ARXIV.2312.11805. URL https://doi.org/10.48550/arXiv.2312.11805.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with llms. *CoRR*, abs/2501.12599, 2025. doi: 10.48550/ARXIV.2501.12599. URL https://doi.org/10.48550/arXiv.2501.12599.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents, 2024. URL https://arxiv.org/abs/2407.18901.

Daoyu Wang, Qingchuan Li, Mingyue Cheng, Jie Ouyang, Shuo Yu, Chunli Liu, Shijin Wang, Qi Liu, and Enhong Chen. Capo: Critic-guided action-aligned policy optimization for advancing llm agent capabilities, 2026a. URL https://arxiv.org/abs/2604.18401.

Junzhe Wang, Zhiheng Xi, Yajie Yang, Hao Luo, Shihan Dou, Tao Gui, and Qi Zhang. Enhancing llm-based search agents via contribution weighted group relative policy optimization, 2026b. URL https://arxiv.org/abs/2604.14267.

Tao Wang, Suhang Zheng, and Xiaoxiao Xu. Rtmc: Step-level credit assignment via rollout trees, 2026c. URL https://arxiv.org/abs/2604.11037.

Yunan Wang, Minghui Song, Zihan Zhang, Shaohan Huang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. Group-graph policy optimization for long-horizon agentic reinforcement learning, 2026d. URL https://arxiv.org/abs/2606.22995.

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, 2025. URL https://arxiv.org/abs/2504.20073.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. Qwen2 technical report. *CoRR*, abs/2407.10671, 2024. doi: 10.48550/ARXIV.2407.10671. URL https://doi.org/10.48550/arXiv.2407.10671.

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, 2026. URL https://arxiv.org/abs/2607.22724.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 20744–20757. Curran Associates, Inc., 2022. doi: 10.52202/068431-1508. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/82ad13ec01f9fe44c01cb91814fd7b8c-Paper-Conference.pdf.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023. URL https://arxiv.org/abs/2210.03629.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Yuexiang Zhai, Hao Bai, Zipeng Lin, Jiayi Pan, Shengbang Tong, Yifei Zhou, Alane Suhr, Saining Xie, Yann LeCun, Yi Ma, and Sergey Levine. Fine-tuning large vision-language models as decision-making agents via reinforcement learning, 2024. URL https://arxiv.org/abs/2405.10292.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl, 2024. URL https://arxiv.org/abs/2402.19446.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL https://arxiv.org/abs/1909.08593.

A ALGORITHM

Algorithm 1 CRBC with full Bellman closure

Require: Initial policy π_θ, fixed reference policy π_{ref}, task distribution p, anchor map ϕ, discount $\beta \in (0,1)$, advantage weights $w_{\mathrm{G}}, w_{\mathrm{S}}$, clipping parameter ϵ, KL penalty λ_{KL}, group size G, normalization floor $\sigma_{\min} > 0$

for each training iteration **do**
  Set $\theta_{\mathrm{old}} \leftarrow \theta$ and sample a batch of tasks from p
  for each task g in the batch **do**
    // Group rollout
    Initialise G environments from the same task condition
    Collect complete rollouts using $y_{i,t} \sim \pi_{\theta_{\mathrm{old}}}(\cdot \mid h_{i,t})$ and $a_{i,t} = \mathrm{Exec}(y_{i,t})$; retain contexts, responses, and terminal outcomes R_i
    // Empirical process and Bellman closure
    Map $s_{i,t} = \phi(h_{i,t})$ and construct $\mathcal{D}_g$ with successors $x_{i,t}$ and absorbing boundaries as in Section 3
    Build (P_g, μ_g) from transition counts using equation 5
    Form M_g, b_g using equation 8 and solve $(I - \beta M_g)V_g = b_g$
    Set $V_g(z^+) = 1$ and $V_g(z^-) = 0$, then compute $Q_g = \mathcal{B}_g V_g$
    // Credit construction and combination
    Compute trajectory-level group-relative advantages A_i^{G} using the underlying optimizer
    Compute $A_g^{(\infty)}(s,a) = Q_g(s,a) - V_g(s)$ and its standard deviation $\sigma_g(s)$ under $\mu_g(\cdot \mid s)$ at each anchor
    Compute the gated and normalized step-level credit A_g^{S} using equation 13
    Set $A_{i,t}^{\mathrm{final}} = w_{\mathrm{G}} A_i^{\mathrm{G}} + w_{\mathrm{S}} A_g^{\mathrm{S}}(s_{i,t}, a_{i,t})$ for every step in the group
  end for
  // Policy update
  Hold advantages and graph statistics fixed, and update θ using the objective $\mathcal{J}_{\mathrm{CRBC}}(\theta)$ in equation 15
end for

Algorithm 1 summarizes the full-closure training procedure. At each iteration, CRBC constructs an empirical process for each rollout group, solves its behaviour-policy Bellman fixed point, and combines the resulting normalized step credit with trajectory-level group-relative advantages for policy optimization.

B PROOFS

All value functions, including $V_g^{(0)}$, have fixed boundary values $V(z^+) = 1$ and $V(z^-) = 0$. Matrix expressions act on their restrictions to $\mathcal{S}_g$, where $\mathcal{T}_g V = \beta M_g V + b_g$ and $\|M_g\|_\infty \le 1$. For a state vector u and a policy ν on the observed action support, define

$$(P_g^S u)(s,a) := \sum_{s' \in \mathcal{S}_g} P_g(s' \mid s,a)\, u(s'), \qquad (Lu)(s,a) := u(s),$$
$$M_g^\nu(s,s') := \sum_{a \in \mathcal{A}_g(s)} \nu(a \mid s)\, P_g(s' \mid s,a). \tag{16}$$

These operators are non-expansive in the sup norm. Let $\mathcal{T}_g^\nu$ be the state backup with μ_g replaced by ν, and V_g^ν its fixed point. Since boundary differences vanish, for $U, W \in \mathcal{V}_g$,

$$(\mathcal{B}_g U)(s,a) - (\mathcal{B}_g W)(s,a) = \beta\big(P_g^S(U - W)\big)(s,a). \tag{17}$$

B.1 PROOF OF PROPOSITION 1

Invariance. For $V \in \mathcal{V}_g$ and $s \in \mathcal{S}_g$,

$$0 \le (\mathcal{T}_g V)(s) = \beta \sum_a \mu_g(a \mid s) \sum_x P_g(x \mid s,a) V(x) \le \beta < 1.$$

Together with the fixed boundary values, this shows that $\mathcal{T}_g$ maps $\mathcal{V}_g$ into itself.

Contraction and conclusion. For $U, V \in \mathcal{V}_g$, substochasticity gives

$$\|\mathcal{T}_g U - \mathcal{T}_g V\|_\infty = \beta \|M_g(U - V)\|_\infty \le \beta \|U - V\|_\infty.$$

The space $\mathcal{V}_g$ is closed in the finite-dimensional sup-norm space and hence complete. The Banach fixed point theorem therefore gives a unique fixed point, and Bellman iteration converges to it from every initializer with error at most β^K times the initial error. Moreover, $V_g(s) \in [0, \beta]$ for $s \in \mathcal{S}_g$. Since $\|\beta M_g\|_\infty \le \beta < 1$, the inverse $(I - \beta M_g)^{-1} = \sum_{k \ge 0} (\beta M_g)^k$ exists, is entrywise non-negative, and has norm at most $(1 - \beta)^{-1}$. The fixed-point equation then yields equation 9. □

B.2 Proof of Proposition 2

Feasible direction. Centring gives $\sum_a \xi(s,a) = w(s) \sum_a \mu_g(a \mid s) \tilde{A}(s,a) = 0$, so ν_α sums to one at each anchor. Non-negativity holds for $0 \le \alpha \le \alpha_\xi$, where $\alpha_\xi := \min\{\mu_g(a \mid s)/|\xi(s,a)| : \xi(s,a) < 0\}$, with $\alpha_\xi = \infty$ when no negative entries exist.

Value difference. Using the fixed-point equation and centring of $\tilde{A}$,

$$\begin{aligned} \big(\mathcal{T}_g^{\nu_\alpha} V_g - V_g\big)(s) &= \sum_a \big[\nu_\alpha(a \mid s) - \mu_g(a \mid s)\big] Q_g(s,a) \\ &= \alpha\, w(s) \langle \tilde{A}, A_g^{(\infty)} \rangle_{\mu_g}(s). \end{aligned} \tag{18}$$

Subtracting the two Bellman equations gives $\big(I - \beta M_g^{\nu_\alpha}\big)\big(V_g^{\nu_\alpha} - V_g\big) = \mathcal{T}_g^{\nu_\alpha} V_g - V_g$, establishing equation 11. The resolvent uses the perturbed policy ν_α and is entrywise non-negative by its Neumann-series representation.

Self-alignment. For $\tilde{A} = A_g^{(\infty)}$,

$$\big\langle A_g^{(\infty)}, A_g^{(\infty)} \big\rangle_{\mu_g}(s) = \mathrm{Var}_{a \sim \mu_g(\cdot|s)}\big[Q_g(s,a)\big] = \sigma_g(s)^2 \ge 0.$$

The forcing and resolvent are therefore non-negative, yielding $V_g^{\nu_\alpha} \succeq V_g$. Improvement is strict at s when $\alpha > 0$ and an anchor with $w > 0$ and $\sigma_g > 0$ is reachable from s under ν_α, including a path of length zero. □

B.3 Residual decomposition and finite-depth transport

Residual definitions. Define the action-level residual and its state-wise average:

$$\begin{aligned} \delta_g^{(0)}(s,a) &:= (\mathcal{B}_g V_g^{(0)})(s,a) - Q_g^{(0)}(s,a) = Q_g^{(1)}(s,a) - Q_g^{(0)}(s,a), \\ \bar{\delta}_g^{(0)}(s) &:= \sum_a \mu_g(a \mid s)\, \delta_g^{(0)}(s,a) = (\mathcal{T}_g V_g^{(0)})(s) - V_g^{(0)}(s). \end{aligned} \tag{19}$$

Action-level residuals can cancel in this average, so $\bar{\delta}_g^{(0)} \equiv 0$ need not imply $\delta_g^{(0)} \equiv 0$.

Provenance. For an observed transition (s, a, x), define the suffix evidence arriving from (s, a) by

$$V_{g \leftarrow (s,a)}^{(0)}(x) := \frac{1}{N_g(s,a,x)} \sum_{\substack{(i,t):\, s_{i,t} = s,\ a_{i,t} = a, \\ x_{i,t} = x}} \beta^{T_i - t} R_i. \tag{20}$$

Set this quantity to zero for unobserved successors, whose transition probabilities are zero. Factoring $\beta^{T_i - t + 1} R_i = \beta(\beta^{T_i - t} R_i)$ and grouping occurrences by successor gives

$$Q_g^{(0)}(s,a) = \beta \sum_{x \in \mathcal{X}_g} P_g(x \mid s,a)\, V_{g \leftarrow (s,a)}^{(0)}(x).$$

Subtracting this from the action backup yields

$$\delta_g^{(0)}(s,a) = \beta \sum_{x \in \mathcal{X}_g} P_g(x \mid s,a)\Big[V_g^{(0)}(x) - V_{g\leftarrow(s,a)}^{(0)}(x)\Big]. \tag{21}$$

Boundary contributions vanish. For $x \in \mathcal{S}_g$, $V_g^{(0)}(x) = N_g(x)^{-1}\sum_{(i,t)\in\mathcal{I}_g(x)} G_{i,t}$ averages all occurrences at x, whereas the incoming average includes only those reached from (s,a). Thus the residual vanishes if every observed non-terminal successor has no initial occurrences and receives transitions only from (s,a), or if the weighted discrepancies cancel. Acyclicity alone does not suffice, since different state-action pairs may share a successor.

Transport. Since $(I-\beta M_g)(V_g - V_g^{(0)}) = \bar{\delta}_g^{(0)}$, set $d_g := (I-\beta M_g)^{-1}\bar{\delta}_g^{(0)} = V_g - V_g^{(0)}$. The affine Bellman recurrence gives, by induction,

$$V_g - V_g^{(K)} = (\beta M_g)^K d_g, \qquad K \geq 0. \tag{22}$$

For $K \geq 1$, equation 17 then gives

$$\begin{aligned} Q_g - Q_g^{(K)} &= \beta P_g^S (\beta M_g)^{K-1} d_g, \\ A_g^{(\infty)} - A_g^{(K)} &= \Big[\beta P_g^S (\beta M_g)^{K-1} - L(\beta M_g)^K\Big] d_g. \end{aligned} \tag{23}$$

At $K = 0$, adding and subtracting $\mathcal{B}_g V_g^{(0)}$ instead yields

$$\begin{aligned} Q_g - Q_g^{(0)} &= \delta_g^{(0)} + \beta P_g^S d_g, \\ A_g^{(\infty)} - A_g^{(0)} &= \delta_g^{(0)} + \big(\beta P_g^S - L\big) d_g. \end{aligned} \tag{24}$$

Consequently, the $K = 0$ deviation cannot be controlled solely by $\|\bar{\delta}_g^{(0)}\|_\infty$: when $\bar{\delta}_g^{(0)} \equiv 0$, we have $d_g = 0$, but $A_g^{(\infty)} - A_g^{(0)} = \delta_g^{(0)}$ may remain nonzero. Considering the sequences from $K = 0$ onward, the state-value sequence is constant if and only if $\bar{\delta}_g^{(0)} \equiv 0$, whereas the paired sequence $(V_g^{(K)}, Q_g^{(K)})$ is constant if and only if $\delta_g^{(0)} \equiv 0$.

B.4 Proof of Corollary 3

Depth bound. For $K \geq 1$, non-expansiveness of the operators in equation 23 and the bound on $(I-\beta M_g)^{-1}$ give

$$\big\|A_g^{(\infty)} - A_g^{(K)}\big\|_\infty \leq 2\beta^K \|d_g\|_\infty \leq \frac{2\beta^K}{1-\beta}\big\|\bar{\delta}_g^{(0)}\big\|_\infty =: E_K,$$

establishing equation 12. The restriction $K \geq 1$ follows from the additional action-level residual in equation 24.

Alignment and value improvement. For a credit $\tilde{A}$ centred under μ_g, write $e := A_g^{(\infty)} - \tilde{A}$ and use the weighted norm $\|e(s,\cdot)\|_{\mu_g} := \big(\sum_a \mu_g(a \mid s) e(s,a)^2\big)^{1/2}$. Cauchy–Schwarz gives, anchor-wise,

$$\big\langle \tilde{A}, A_g^{(\infty)} \big\rangle_{\mu_g} = \sigma_g^2 - \big\langle e, A_g^{(\infty)} \big\rangle_{\mu_g} \geq \sigma_g\big(\sigma_g - \|e\|_{\mu_g}\big). \tag{25}$$

Taking $\tilde{A} = A_g^{(K)}$, which is centred under μ_g, gives $\|e(s,\cdot)\|_{\mu_g} \leq \|e\|_\infty \leq E_K$. Hence the inner product is non-negative whenever $E_K \leq \sigma_g(s)$. If $\sigma_g(s) = 0$, then $A_g^{(\infty)}(s,\cdot) = 0$ on the observed support, so the inner product is zero regardless of E_K. At anchors with $w(s) = 0$, the forcing term in equation 11 is also zero. Thus, if $E_K \leq \sigma_g(s)$ at every anchor with $w(s) > 0$ and $\sigma_g(s) > 0$, the forcing is componentwise non-negative. For any feasible reweighting from Proposition 2, the non-negative resolvent in equation 11 then yields $V_g^{\nu_\alpha} \succeq V_g$.

Existence of a finite depth. Let $\mathcal{S}_{g,+}(w) := \{s \in \mathcal{S}_g : w(s) > 0, \sigma_g(s) > 0\}$. If this set is non-empty, its finiteness gives $\min_{s\in\mathcal{S}_{g,+}(w)} \sigma_g(s) > 0$. Since $E_K \to 0$, the sufficient condition holds for all sufficiently large finite K. If the set is empty, all forcing terms are zero and the condition holds vacuously. Failure to satisfy this sufficient condition does not imply misalignment. The decay of E_K improves the certified lower bound $\sigma_g(\sigma_g - E_K)$, but does not imply monotone alignment, preserved action rankings, or a particular training-curve shape. □

Table 4: Environment-level training settings used in our reproduced experiments. Settings are shared across methods within each model scale and environment.

Setting	ALFWorld	WebShop	Sokoban
Maximum environment steps	50	15	15
Maximum prompt tokens	2048	5120	1024
Maximum response tokens	512	512	512
Training groups per update	16	16	32
Rollouts per group	8	8	8
Validation trajectories	128	128	128
Base model	Qwen2.5-1.5B/7B	Qwen2.5-1.5B/7B (Qwen et al., 2025)	Qwen2.5-VL-3B (Bai et al., 2025)

C Experiment Details

C.1 Comparing Methods

We compare CRBC with the following baselines.

- **GPT-4o**: a closed-source, general-purpose LLM used as a reference for multi-turn agent capability (OpenAI, 2023).
- **Gemini-2.5-Pro**: a second closed-source LLM reference point with strong general-purpose reasoning ability (Team, 2023).
- **ReAct**: an in-context prompting agent that interleaves reasoning and acting (Yao et al., 2023).
- **Reflexion**: a prompting agent that uses verbal reflection and iterative self-improvement without parameter updates (Shinn et al., 2023).
- **PPO**: the standard clipped policy-gradient method with a learned value critic and GAE (Schulman et al., 2017).
- **RLOO**: a critic-free, trajectory-level leave-one-out baseline (Ahmadian et al., 2024).
- **GRPO**: a critic-free group objective that normalizes one outcome advantage over the rollouts of a task (Shao et al., 2024).
- **GiGPO**: a step-level method that compares occurrences at repeated anchor states (Feng et al., 2025).
- **GraphGPO**: a graph-based method that scores transitions by a shortest-path/extremal readout (Cheng et al., 2026).
- **HGPO**: a hierarchical grouping method that refines step comparisons using historical-context consistency (He et al., 2026).
- **CRBC (ours)**: the predictor-free, full-closure setting described in Appendix C.3.

C.2 Environment Details

ALFWorld. ALFWorld (Shridhar et al., 2021) is a text-based embodied environment in which an agent must complete a household task through a long-horizon sequence of admissible commands. We use the six task families reported by the environment evaluator: Pick, Clean, Cool, Look, Heat, and Pick2. The training wrapper is `AlfredTWEnv`; it augments the textual observation with the current location, holding/item status, item-transition history, and the sorted admissible-command list. Success and failure are terminal outcomes, and a trajectory is truncated after at most 50 environment steps. Each validation pass evaluates 128 in-distribution trajectories selected by the environment seed.

WebShop. WebShop (Yao et al., 2022) is a text-based interactive shopping environment in which the agent searches, navigates, and selects an item that matches a natural-language request. We use the repository's text-rich, small-catalogue configuration (`observation_mode=text_rich` and `use_small=True`, using the repository's

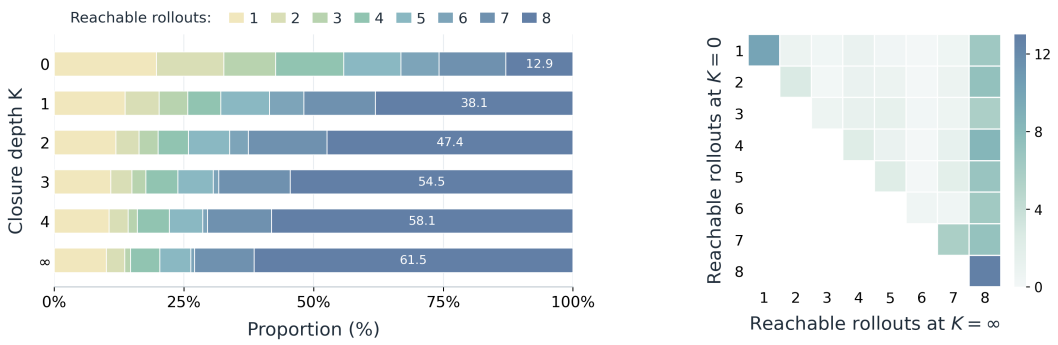


Figure 5: Cross-rollout evidence coverage measured on rollouts from an early-training checkpoint on ALFWorld. Left: distribution of the number of reachable rollouts at each closure depth K. Right: paired coverage at $K = 0$ and full closure, with color indicating the percentage of all eligible visits.

`items_shuffle_1000.json`/`items_ins_v2_1000.json` files) with two retained interaction records. The `WebAgentTextEnv` wrapper limits a trajectory to 15 environment steps; the validation goals use indices 0–499 and training starts at index 500 in the launcher. We report both the raw task score and the binary task success rate.

Sokoban. Sokoban (Schrader, 2018) tests transfer to a visual interactive setting. We use the 6×6 one-box configuration with `Qwen2.5-VL-3B-Instruct` (Bai et al., 2025); the checkpoint is pinned to the revision used by the releazed launcher. The environment returns an RGB observation and the agent chooses among the four box-pushing directions. Trajectories are limited to 15 environment steps. We report the final raw task score ("`text_score`") and the percentage of solved boards ("`success_rate`").

Anchor construction. Anchors are matched by exact equality of their environment-specific representations within each task group. ALFWorld uses the current textual observation augmented with location, held-item status, item-transition records, and sorted admissible commands; WebShop uses the formatted current-page observation; Sokoban uses the current RGB observation. The recent interaction records appended to policy prompts are excluded from the matching key.

C.3 Details of Training

Shared protocol. For each update, text experiments sample 16 task groups with $G = 8$ rollouts per group, giving 128 training rollouts. Sokoban uses 32 groups with the same group size. All settings use 128 validation trajectories, validate before training and every five updates, and train for 150 updates. The rollout temperature is 1.0 and the validation temperature is 0.4 with sampling enabled. The actor learning rate is 10^{-6}, the reference-policy KL-loss coefficient is 0.01 (the low-variance KL form), and gradient checkpointing is enabled. For ALFWorld and WebShop, a successful trajectory receives reward 10 and failure receives 0. An invalid-action penalty of 0.1 is applied before advantage computation. PPO alone uses a learned critic with learning rate 10^{-5}; group-based methods do not train an auxiliary critic. Environment and data-loader seeds are set to the same value for each run, and means and standard deviations are computed only over completed independent seeds.

Resource and batching settings. The batching choices are adjusted only to fit the model and environment; within each comparison they are identical across methods. For ALFWorld, the 1.5B and 7B runs use tensor parallel sizes 1 and 4, actor micro-batches 32 and 4 per GPU, and PPO mini-batches 256 and 64, respectively. For WebShop, the corresponding tensor parallel sizes are 2 and 4, actor micro-batches are 8 and 4, and the PPO mini-batch is 64 for both scales. Sokoban uses tensor parallel size 2, actor micro-batch 8, and PPO mini-batch 64. The rollout and reference log-probability micro-batches are 32 (ALFWorld 1.5B), 16 (ALFWorld 7B and WebShop), and 16 (Sokoban). All reproduced experiments are conducted on a single node equipped with eight NVIDIA A100 GPUs (80 GB each); tensor parallelism and micro-batches are adjusted within this fixed budget as specified above.

Table 5: Credit and weight ablations on ALFWorld with Qwen2.5-1.5B-Instruct. We report final validation success rates (%) for six subtasks and the overall success rate (All) at step 150, averaged over three random seeds. Best values are in **bold**.

w_{G}	w_{S}	Credit assignment	Pick	Clean	Cool	Look	Heat	Pick2	All
0	5	Step-level only	98.89$_{(1.57)}$	**98.25**$_{(2.48)}$	93.54$_{(1.74)}$	**94.44**$_{(3.93)}$	88.89$_{(2.24)}$	96.67$_{(2.36)}$	95.31$_{(1.69)}$
1	0	Trajectory-level only	84.62$_{(1.46)}$	64.91$_{(2.48)}$	86.92$_{(6.79)}$	58.33$_{(0.00)}$	77.78$_{(5.94)}$	73.33$_{(4.71)}$	76.56$_{(2.30)}$
1	1	Step + trajectory	98.89$_{(1.57)}$	87.72$_{(2.48)}$	94.82$_{(1.78)}$	**94.44**$_{(3.93)}$	88.89$_{(2.24)}$	93.33$_{(2.36)}$	93.75$_{(1.69)}$
1	2.5	Step + trajectory	98.89$_{(1.57)}$	92.98$_{(4.96)}$	**97.38**$_{(1.85)}$	86.11$_{(3.93)}$	92.06$_{(2.24)}$	93.33$_{(2.36)}$	94.53$_{(1.91)}$
1	5	**Step + trajectory (Ours)**	**100.00**$_{(0.00)}$	92.98$_{(6.56)}$	92.15$_{(3.33)}$	91.67$_{(0.00)}$	**96.83**$_{(4.49)}$	**100.00**$_{(0.00)}$	**96.09**$_{(0.64)}$
1	10	Step + trajectory	98.89$_{(1.57)}$	92.98$_{(2.48)}$	94.87$_{(3.63)}$	91.67$_{(0.00)}$	88.89$_{(2.24)}$	98.33$_{(2.36)}$	94.79$_{(1.95)}$

Table 6: Bellman-discount ablation on ALFWorld with Qwen2.5-1.5B-Instruct. Only β varies; all other CRBC settings are fixed. Results are averaged over three random seeds; best values are in **bold**.

Bellman discount β	Pick	Clean	Cool	Look	Heat	Pick2	All
0.95	98.89$_{(1.57)}$	**98.25**$_{(2.48)}$	**94.82**$_{(1.78)}$	91.67$_{(0.00)}$	88.89$_{(2.24)}$	98.33$_{(2.36)}$	95.57$_{(0.37)}$
0.98	**100.00**$_{(0.00)}$	92.98$_{(6.56)}$	92.15$_{(3.33)}$	91.67$_{(0.00)}$	**96.83**$_{(4.49)}$	**100.00**$_{(0.00)}$	**96.09**$_{(0.64)}$
0.99	**100.00**$_{(0.00)}$	86.84$_{(2.63)}$	94.23$_{(5.77)}$	**95.83**$_{(4.17)}$	92.86$_{(2.38)}$	**100.00**$_{(0.00)}$	95.31$_{(1.56)}$
1.00	**100.00**$_{(0.00)}$	89.47$_{(10.53)}$	92.23$_{(3.77)}$	83.33$_{(8.33)}$	95.24$_{(4.76)}$	92.50$_{(2.50)}$	93.36$_{(0.39)}$

Method-specific settings. GraphGPO uses its default shortest-path readout with graph discounts 0.10, 0.20, and 0.80 on ALFWorld, WebShop, and Sokoban, respectively, and step/trajectory weights $1/1$. GiGPO uses a return discount of 0.95 and step/trajectory weights $1/1$ (the final Sokoban launcher uses its `mean_norm` variant). GRPO uses the standard group-normalized trajectory advantage, while PPO and RLOO use GAE and a trajectory-level leave-one-out baseline, respectively. Unless otherwise stated, CRBC uses the full-closure configuration with $K = \infty$, $\beta = 0.98$, $w_{\mathrm{G}} = 1$, $w_{\mathrm{S}} = 5$, and a normalization floor $\sigma_{\min} = 0.1$. The empirical process is formed from the observed rollout transitions, and the resulting $Q - V$ signal is normalized per anchor before being combined with the trajectory-level advantage. All other training settings follow the common protocol above.

D Additional Experiments

D.1 Credit assignment and weight sensitivity.

Table 5 evaluates the trajectory-level and step-level weights w_{G} and w_{S}. Trajectory-level credit alone $(1, 0)$ achieves 76.56% overall success, whereas step-level closure credit alone $(0, 5)$ reaches 95.31%. With $w_{\mathrm{G}} = 1$, increasing w_{S} from 1 to 2.5 and 5 raises success from 93.75% to 94.53% and 96.09%; a further increase to 10 reduces it to 94.79%. These results suggest that closure credit provides the main gains, complemented by appropriately weighted trajectory-level credit. The default $(1, 5)$ achieves the highest overall success, exceeding step-level credit alone by 0.78 points.

D.2 Bellman Discount

Table 6 evaluates the Bellman discount β on ALFWorld with Qwen2.5-1.5B-Instruct, keeping all other CRBC settings fixed. All results are averaged over three random seeds, with the default $\beta = 0.98$ using the main CRBC result. Across the tested discounted settings $\beta \in \{0.95, 0.98, 0.99\}$, overall success remains between 95.31% and 96.09%, with $\beta = 0.98$ achieving the highest observed mean. Removing discounting ($\beta = 1.00$) reduces overall success to 93.36%, a decrease of 2.73 percentage points relative to the default.

Subtask preferences vary: $\beta = 0.95$ performs best on Clean and Cool, $\beta = 0.99$ on Look, and $\beta = 0.98$ on Heat. All configurations use full closure, so β controls how strongly distant evidence is attenuated along the merged graph. These results suggest that mild discounting is beneficial in

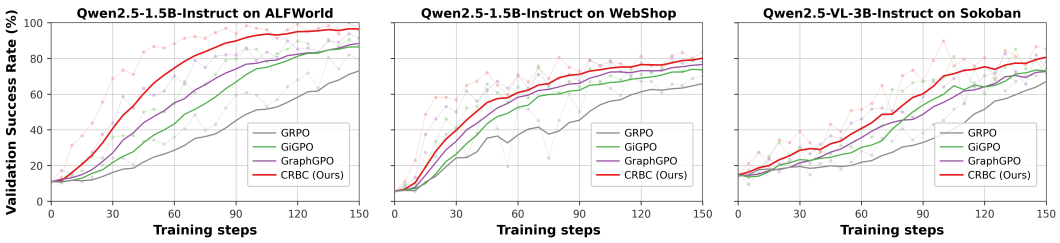


Figure 6: Validation success-rate curves for CRBC (Ours, red), GraphGPO (purple), GiGPO (green), and GRPO (gray) on ALFWorld, WebShop, and Sokoban. Light curves show validation measurements recorded every five training updates, while dark curves show EMA-smoothed trends ($\alpha = 0.95$ per update). Qwen2.5-1.5B-Instruct is used for ALFWorld and WebShop, and Qwen2.5-VL-3B-Instruct for Sokoban.

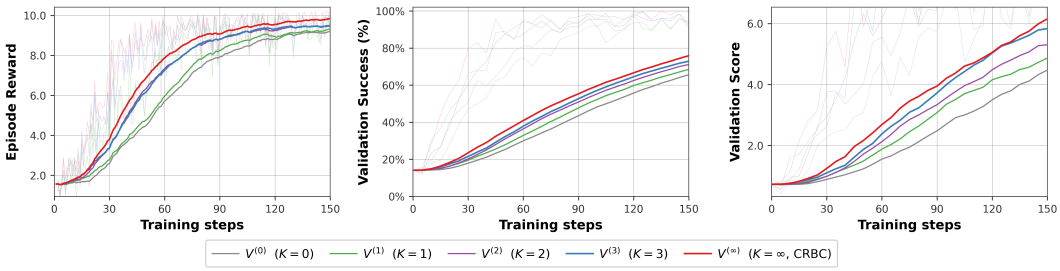


Figure 7: Training dynamics for the closure-depth ablation on ALFWorld with Qwen2.5-1.5B-Instruct. Only the closure depth K varies; all other settings are fixed. From left to right, we show episode reward, validation success rate, and validation score. Light and dark curves denote raw and EMA-smoothed trajectories ($\alpha = 0.95$); deeper closure gives stronger late-training performance.

this setting and support $\beta = 0.98$ as a practical default, with limited sensitivity among the tested discounted values.

D.3 Closure Depth and Evidence Coverage

Figure 5 examines cross-rollout evidence coverage using rollouts from an early-training checkpoint on ALFWorld. We consider visits to anchors with at least two distinct observed actions. For each eligible visit, coverage counts the distinct rollouts whose evidence is structurally accessible at depth K within its eight-rollout group, without weighting their contributions. The left panel shows the coverage distribution: increasing K allows evidence from shared successors to propagate back through the merged graph, shifting visits toward broader coverage. The proportion of eligible visits with access to all eight rollouts increases from 12.9% at $K = 0$ to 61.5% under full closure.

The right panel pairs coverage before and after closure for the same visits, with rows representing $K = 0$, columns representing full closure, and color intensity indicating the percentage of all eligible visits. Diagonal cells indicate unchanged coverage; cells with larger column values indicate expansion. Overall, coverage increases for 63.0% of eligible visits, including those whose expanded coverage remains below eight rollouts. This structural view complements the final performance comparison in Table 2 and the corresponding training curves in Figure 7, illustrating how closure makes additional cross-rollout evidence available.

D.4 Validation Success

Figure 6 reports validation success rates on ALFWorld, WebShop, and Sokoban, measured every five training updates. CRBC generally reaches high success rates earlier and achieves stronger final performance than the compared baselines. These trends complement the training reward curves in Figure 3, showing that the observed learning-efficiency gains are also reflected in task success on the validation sets.

Table 7: Rollout group size comparison on ALFWorld with Qwen2.5-1.5B-Instruct at step 150. We report validation success rates (%) for six subtasks and overall (All), together with task score. Best reported values within each group size are in **bold**.

Group size G	Method	Pick	Clean	Cool	Look	Heat	Pick2	All	Score
4	GRPO	60.00	36.84	46.15	50.00	38.10	50.00	47.66	2.27
4	GiGPO	90.00	63.16	**88.46**	66.67	80.95	75.00	79.69	5.34
4	GraphGPO	86.67	68.42	84.62	58.33	76.19	75.00	77.34	4.38
4	**CRBC (Ours)**	**100.00**	**94.74**	**88.46**	**91.67**	**100.00**	**80.00**	**92.97**	**7.12**
16	GRPO	93.33	73.68	80.77	58.33	85.71	80.00	81.25	5.08
16	GiGPO	**100.00**	89.47	88.46	83.33	**100.00**	90.00	92.97	7.01
16	GraphGPO	**100.00**	78.95	88.46	66.67	90.48	85.00	87.50	5.30
16	**CRBC (Ours)**	**100.00**	**94.74**	**96.15**	**100.00**	90.48	**100.00**	**96.88**	**8.34**

D.5 Rollout Group Size

Table 7 compares group-based RL methods with rollout group sizes $G \in \{4, 16\}$ on ALFWorld using Qwen2.5-1.5B-Instruct. At $G = 4$, CRBC achieves 92.97% success, compared with 47.66% for GRPO, 79.69% for GiGPO, and 77.34% for GraphGPO. At $G = 16$, CRBC reaches 96.88%, versus 92.97% for GiGPO and 87.50% for GraphGPO. CRBC therefore improves over the strongest baseline by 13.28 and 3.91 percentage points, respectively, while also obtaining the highest task scores of 7.12 and 8.34. These comparisons show that CRBC retains an advantage at both tested group sizes.

E Prompt Templates

Figure 8 shows the prompt templates for ALFWorld, WebShop, and Sokoban. For the text environments, prompts retain at most the two most recent observation–action records. For Sokoban, the prompt contains the current RGB image, task rules, and four movement directions, without textual interaction history. In all three environments, the agent is instructed to place its reasoning inside `<think>...</think>` and its selected action inside `<action>...</action>`. Within each environment, all reproduced RL methods share the same prompt templates and memory protocol.

ALFWorld — embodied action prompt

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

(a) ALFWorld prompt

WebShop — shopping action prompt

You are an expert autonomous agent operating in the WebShop e-commerce environment. Your task is to: `{task_description}`. Prior to this step, you have already taken `{step_count}` step(s). Below are the most recent `{history_length}` observations and the corresponding actions you took: `{action_history}`. You are now at step `{current_step}` and your current observation is: `{current_observation}`. Your admissible actions for the current situation are: [`{available_actions}`]. Now it is your turn to take one action for the current step. First reason step by step about the current situation, then carefully choose which admissible action best advances the shopping goal. This reasoning process **MUST** be enclosed within `<think>...</think>`. Once you have finished your reasoning, choose an admissible action for the current step and present it within `<action>...</action>`.

(b) WebShop prompt

Sokoban — visual action prompt

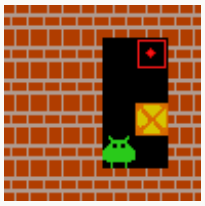
You are an expert agent operating in the Sokoban environment. Your goal is to push all the boxes onto the target spots. Once all boxes are on the targets, you win!

Rules

You can only push boxes. You can't pull them, so plan ahead to avoid getting stuck. You can't walk through or push boxes into walls. To avoid traps, do not push boxes into corners or against walls where they can't be moved again.

Visual Elements in the Image

Character: A small, green alien-like figure with two antennae and black eyes. It represents you.
Box: A yellow crate marked with an orange "X" across its front. It is the box you need to push.
Target: A black tile outlined in red, with a small red diamond shape in the center. It marks the destination where a box should be pushed.



Example RGB input

Current Step

Your current observation is shown in the image: `<image>`
Your admissible actions are ["up", "down", "left", "right"].

Now it's your turn to make a move (choose ONE action only for the current step). You should first reason step-by-step about the current situation—observe the positions of boxes and targets, plan a path to push a box toward a target, and avoid traps like corners or walls. 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.

(c) Sokoban prompt

Figure 8: Prompt templates for ALFWorld, WebShop, and Sokoban. Coloured placeholders denote runtime inputs. Text environments retain up to two observation–action records; Sokoban uses the current RGB observation without textual interaction history. Panel (c) includes an example RGB input.