CONSENSUS-BREAKING RE-ROLLOUT: RECOVERING MISSING ACTION CONTRASTS FOR LONG-HORIZON AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning for long-horizon language agents relies on limited rollout budgets, yet group-relative objectives obtain no learning signal from groups in which every rollout fails. Existing exploration strategies decide where to spend extra samples based on reward diversity or policy uncertainty. This overlooks a structural signal inside failed groups. We identify this signal as *failure-conditioned consensus*, where failed rollouts repeatedly select the same action at a shared decision point. Such agreement does not show that the action is correct. It marks a decision whose alternatives the current budget has never tested. To exploit this signal, we propose *Consensus-Breaking Re-rollout* (CBR), a budget-aware exploration framework that turns failed consensus into missing action contrasts. CBR operates through two complementary mechanisms. First, it locates consensus anchors within all-failure groups and re-rolls from these anchors while excluding the consensus action, so that alternatives are sampled within the support of the current policy. Second, it merges the recovered outcomes into the original group for two-level group-relative credit assignment. CBR requires neither process supervision nor a learned critic. In a fixed-policy diagnostic with two additional requests per all-failure group, CBR recovers a successful trajectory in 39.6% of groups, compared with 15.6% for ordinary retries and 27.3% for entropy-guided branching. Across ALFWorld and WebShop with 1.5B- and 7B-parameter agents, CBR improves its base learner, GiGPO, by 4.4–8.1 percentage points at a matched generation budget. It also remains competitive with recent credit-assignment methods. These results validate failed consensus as an effective signal for allocating exploration in long-horizon agent learning.

1 INTRODUCTION

Large language model (LLM) agents are increasingly trained with reinforcement learning (RL) in long-horizon environments that provide verifiable terminal outcomes (Sutton & Barto, 2018; Wang et al., 2025b; Luo et al., 2025). Group-relative methods estimate advantages by comparing multiple rollouts of the same task, removing the need for a learned critic (Shao et al., 2024; Guo et al., 2025). However, when every rollout in a group fails, all rewards become identical and the resulting advantages vanish. Such all-failure groups are common, accounting for 18.5% of ALFWorld and 27.8% of WebShop groups at the 1.5B scale. Since each additional rollout requires another multi-step interaction, this creates a fundamental allocation problem: under a limited budget, which failed decisions should receive further exploration?

A widely adopted strategy is to allocate additional sampling where useful learning signals are expected (Bellemare et al., 2016; Osband et al., 2016; Pathak et al., 2017; Burda et al., 2019). Dynamic sampling favors groups with diverse outcomes (Yu et al., 2025), Prefix Sampling reuses prefixes to adjust pass rates (Zhu et al., 2026), entropy-guided branching expands uncertain decisions (Dong et al., 2026), while credit-assignment and weighting methods refine how collected samples influence updates (Feng et al., 2025; Wang et al., 2025a). Despite these advances, exploration allocation is still primarily guided by reward diversity or policy uncertainty. This leaves an important blind spot: an all-failure group contains no reward diversity, and its rollouts may repeatedly select the same ac-

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.

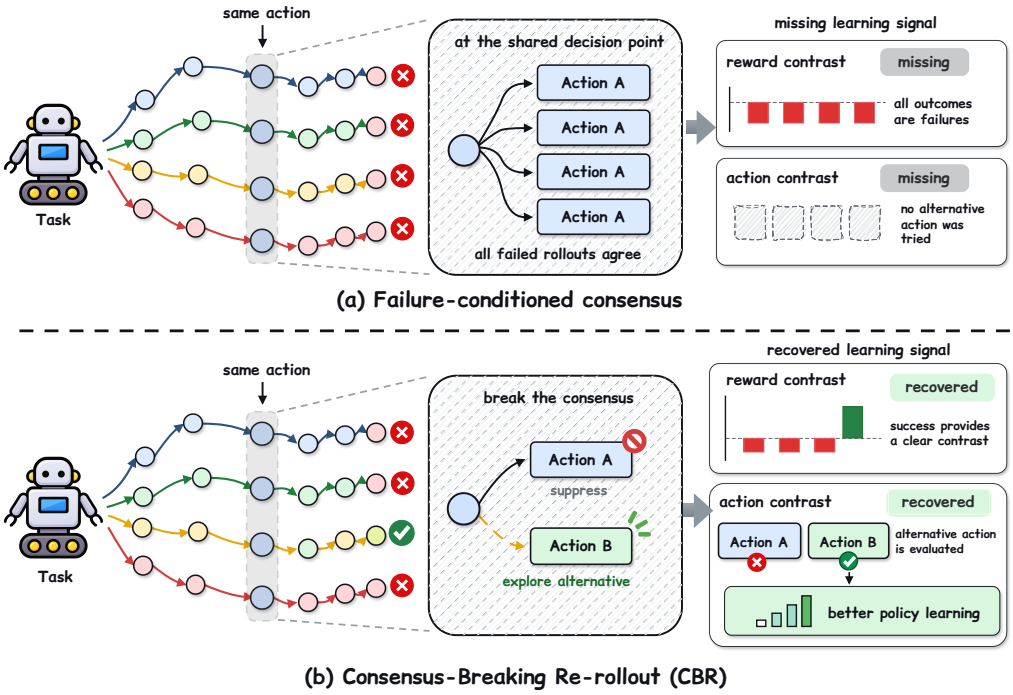


Figure 1: Conceptual illustration of Consensus-Breaking Re-rollout (CBR). (a) All-failure rollouts may repeatedly choose the same action at a shared decision point. This failure-conditioned consensus leaves both reward contrast and action contrast missing, producing little learning signal. (b) CBR breaks this consensus by suppressing the repeated action and sampling an alternative. A successful alternative restores the missing contrasts and provides more informative supervision for policy learning.

tion at a shared decision point, making the group appear settled exactly where alternative behaviors remain untested.

We refer to this phenomenon as *failure-conditioned consensus*: multiple failed rollouts visit the same decision point and select an identical normalized action. Such agreement merely reflects repeated sampling of the same behavior. It indicates neither that the selected action caused the failure nor that the policy has low uncertainty; rather, it reveals where the current rollout budget has failed to measure alternative choices. Empirically, 78.0%–87.3% of all-failure groups contain such consensus points, and policy-proposed alternatives at these points yield larger estimated success gains (0.180) than alternatives at failed non-consensus points (0.121).

These observations suggest that effective exploration under limited budgets requires *targeted contrast acquisition*: additional rollouts should focus on alternatives whose outcomes have not yet been observed. This raises two challenges. First, informative decision points must be identified from failed rollouts alone, without process rewards or learned critics (Lightman et al., 2023; Xi et al., 2025). Second, alternatives must remain within the current policy's support, since ordinary resampling often reproduces the consensus action while manually forcing actions introduces off-policy behavior.

To address these challenges, we propose **Consensus-Breaking Re-rollout (CBR)**, a budget-aware exploration framework that acquires missing action contrasts from failure-conditioned consensus. CBR first identifies *consensus anchors*, i.e., shared decision points in all-failure groups that are repeatedly visited with a single normalized action. It then returns to a selected anchor, samples alternative actions from the current policy while excluding the consensus action, and continues execution on-policy. The acquired outcomes are merged with the original rollouts and consumed by an existing two-level group-relative learner. CBR requires no process supervision, learned critic,

or modified optimization objective; it only changes where additional interaction budget is allocated (Figure 1).

We evaluate CBR on ALFWorld and WebShop with Qwen2.5-1.5B and 7B agents under matched generation budgets (Shridhar et al., 2021; Yao et al., 2022). With two additional requests per all-failure group, CBR obtains at least one successful alternative in 39.6% of groups, compared with 15.6% for ordinary retries, 20.8% for higher-temperature retries, and 27.3% for entropy-guided branching. In training, CBR improves the shared two-level learner by 6.7 points on average (4.4–8.1 across settings), with the smallest gain in the near-saturated 7B ALFWorld setting.

Our contributions are summarized as follows:

- We identify *failure-conditioned consensus* as a missing exploration signal in long-horizon agent reinforcement learning, where repeated agreement among failed rollouts can hide untested alternative actions.
- We propose *Consensus-Breaking Re-rollout (CBR)*, a budget-aware framework that targets consensus anchors and acquires missing action contrasts through selective on-policy exploration.
- We provide diagnostic, intervention, and learning evidence on ALFWorld and WebShop, showing when consensus-based exploration improves group-relative policy optimization and characterizing its limitations.

2 Related Work

Group-relative learning and rollout allocation. Group-relative methods compare rollouts of the same input (Ahmadian et al., 2024). GRPO estimates relative advantages without a learned critic (Shao et al., 2024), while DAPO filters uniform-reward groups and samples further for outcome contrasts (Yu et al., 2025). Rollout Pass-Rate Control uses Prefix Sampling to increase reward diversity: successful prefixes help mostly failing groups, and failing prefixes challenge mostly successful groups (Zhu et al., 2026). These methods manage contrast at the group level. CBR asks where to spend additional samples within an all-failure group, using action consensus at shared decision points to target untested alternatives.

Step-level credit assignment. Process supervision and learned evaluators provide intermediate feedback beyond terminal rewards (Lightman et al., 2023; Xi et al., 2025). Critic-free methods reorganize observed returns: GiGPO compares repeated state visits (Feng et al., 2025), HGPO adds historical context through hierarchical grouping (He et al., 2026), and BiPACE uses behavioral state clusters and action-conditioned peer baselines (Wang et al., 2026). Turn-level and state-abstraction methods offer other credit granularities (Zong et al., 2026; Li et al., 2026). Under the terminal-reward estimator studied here, all-failure groups yield zero episode- and anchor-level reward advantages. If actions also agree at an anchor, alternative-action outcomes are absent. CBR acquires these outcomes for an existing two-level learner; it does not introduce a new credit estimator.

Uncertainty-guided sampling and optimization. ARPO branches at uncertain tool-use decisions (Dong et al., 2026). EMPG adjusts policy-gradient weights using entropy and outcomes (Wang et al., 2025a), and STAPO uses normalized entropy for selective trajectory optimization (Qi et al., 2026). Entropy thus guides sampling and updates. CBR tests failure-conditioned action agreement as an additional allocation signal. Finite-sample agreement implies neither low policy entropy nor an incorrect action, so entropy-based selectors may also choose consensus anchors. The comparison is which allocation rule recovers useful outcome contrasts under a constrained budget.

Returning and branching from intermediate states. Go-Explore returns to archived states before exploring (Ecoffet et al., 2021). Tree-based methods organize branched continuations (Hou et al., 2025; Zhou et al., 2024), while Prefix Sampling reconstructs intermediate states from trajectory prefixes (Zhu et al., 2026). CBR builds on returning and branching but selects shared anchors in all-failure groups by action consensus, excludes the observed action, and incorporates the resulting outcomes into group-relative learning. This ties intervention selection to a missing action comparison. We test whether failed agreement marks decisions worth revisiting.

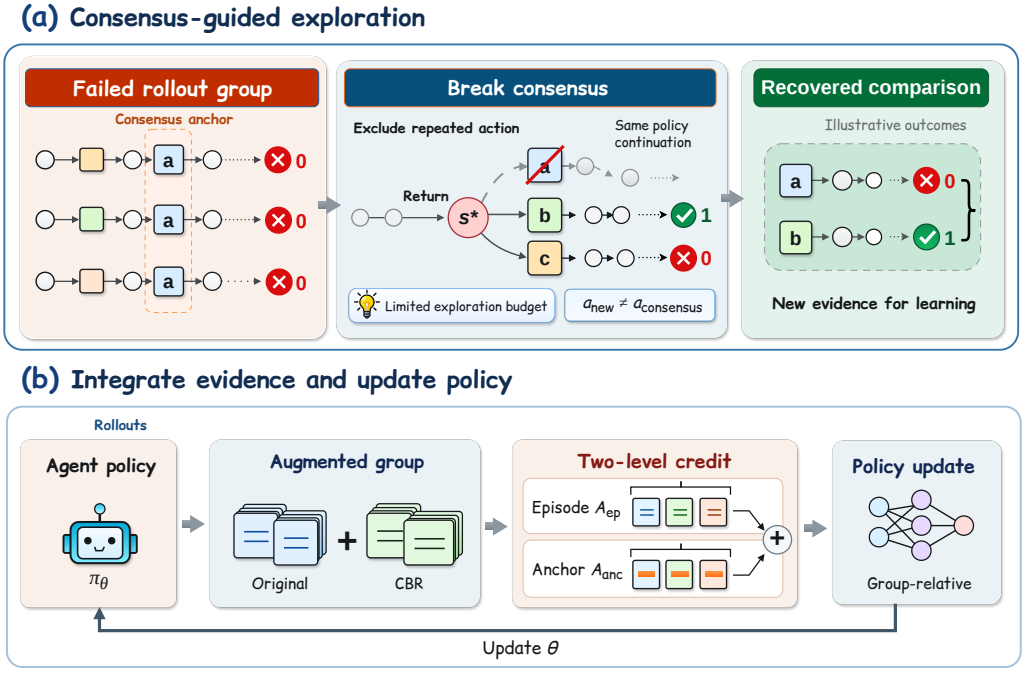


Figure 2: Overview of Consensus-Breaking Re-rollout (CBR). Unlike standard group-relative learning, which cannot learn from all-failure groups with repeated actions, CBR explicitly targets failure-conditioned consensus to recover missing action and reward contrasts. (a) It revisits a consensus anchor, suppresses the repeated action, and explores alternatives under a limited budget. (b) The resulting evidence is merged into the rollout group and used in two-level credit assignment for improved policy optimization.

3 METHOD

The key idea of *Consensus-Breaking Re-rollout (CBR)* is that failed rollout groups can still contain actionable exploration signals. When all sampled trajectories fail and repeatedly select the same action at a decision point, the group lacks an observed alternative rather than evidence that the consensus action is correct. CBR treats such failure-conditioned consensus as a signal for acquiring missing action contrasts. It consists of three stages: identifying consensus blind spots, generating alternative-action comparisons through targeted re-rollouts, and incorporating the acquired contrasts into existing group-relative optimization. CBR does not estimate action values or require process supervision; it only reallocates a limited rollout budget toward informative failures.

3.1 IDENTIFYING FAILURE-CONDITIONED CONSENSUS

For task x, a fixed policy π_{old} generates a rollout group $\mathcal{G}(x) = \{\tau_i\}_{i=1}^{N}$. Each trajectory contains histories $h_{i,t}$, actions $a_{i,t}$, and receives a terminal reward R_i. We define an all-failure group as

$$\text{Stuck}(x) = \mathbb{I}\left[\sum_{i=1}^{N} R_i = 0\right]. \tag{1}$$

To locate missing action comparisons, CBR identifies decision points repeatedly visited with identical actions. An anchor map σ matches either exact interaction histories or task-conditioned observation signatures. For an anchor state s, its visited instances are

$$\mathcal{V}_x(s) = \{(i,t) : \tau_i \in \mathcal{G}(x), \sigma(h_{i,t}) = s\}. \tag{2}$$

Let $\nu(\cdot)$ normalize actions and define the observed action set

$$\mathcal{A}_x(s) = \{\nu(a_{i,t}) : (i,t) \in \mathcal{V}_x(s)\}. \tag{3}$$

A consensus blind spot is defined as

$$\mathcal{C}(x) = \{s : \mathrm{Stuck}(x) = 1, |\mathcal{V}_x(s)| \geq n_{\min}, |\mathcal{A}_x(s)| = 1\}. \tag{4}$$

Each $s \in \mathcal{C}(x)$ represents a decision point where failure removes reward contrast and consensus removes action contrast. Importantly, neither condition implies that the consensus action caused failure or that the policy is uncertain at this state. Instead, the combination identifies where the current rollout budget has not tested alternatives.

CBR selects an intervention target as

$$s^\star \sim \mathrm{Uniform}(\mathcal{C}(x)). \tag{5}$$

Support counts visits rather than independent trajectories. Groups without eligible consensus blind spots receive no targeted intervention. Additional implementation details are provided in Appendix A.

3.2 Breaking Consensus through Alternative Rollouts

Given a selected blind spot $s^\star$, CBR allocates additional rollout requests to obtain an alternative action comparison. The policy first reaches the selected anchor through reliable prefix re-execution or signature matching. If the anchor is not reached, the rollout is treated as an ordinary additional sample.

At the matched history $h^\star$, standard sampling may repeatedly reproduce the consensus action and fail to reveal new evidence. Therefore, CBR samples from a constrained policy that excludes the previously observed action class:

$$q_{\mathrm{old}}(a \mid h^\star) = \frac{\pi_{\mathrm{old}}(a \mid h^\star)\mathbb{I}[\nu(a) \neq a_{\mathrm{con}}(s^\star)]}{1 - p_{\mathrm{con}}(h^\star)}, \tag{6}$$

where

$$p_{\mathrm{con}}(h^\star) = \Pr_{a \sim \pi_{\mathrm{old}}(\cdot|h^\star)}[\nu(a) = a_{\mathrm{con}}(s^\star)]. \tag{7}$$

This intervention does not assume that alternative actions are superior. Instead, it converts an unobserved comparison into an observed one while remaining within the support of the current policy.

The primary implementation uses rejection sampling with at most $L = 4$ proposals. Accepted actions follow q_{old} exactly, and if all proposals are rejected, the intervention is considered unsuccessful. After selecting the alternative action, subsequent generation follows π_{old} normally.

3.3 Learning from Acquired Action Contrasts

Completed CBR rollouts are appended to the original group:

$$\mathcal{G}^+(x) = \mathcal{G}(x) \cup \mathcal{G}_{\mathrm{CBR}}(x). \tag{8}$$

CBR does not introduce a new optimization objective. Instead, it improves existing group-relative learning by enriching the sampled group with missing outcome contrasts.

The episode-level and anchor-level advantages are computed as

$$A_i^{\mathrm{ep},+} = \frac{R_i - \mu_{\mathcal{G}^+}}{s_{\mathcal{G}^+} + \epsilon}, \tag{9}$$

$$A_{i,t}^{\mathrm{anc}} = \frac{G_{i,t} - \mu_{\sigma(h_{i,t})}}{s_{\sigma(h_{i,t})} + \epsilon}, \qquad G_{i,t} = \gamma^{T_i - t} R_i, \tag{10}$$

$$A_{i,t} = A_i^{\mathrm{ep},+} + \lambda A_{i,t}^{\mathrm{anc}}. \tag{11}$$

A successful additional rollout introduces trajectory-level reward variation, while matched anchor visits with different returns provide local relative supervision. If all additional rollouts fail, CBR does not create artificial rewards and the corresponding reward-derived advantages remain unchanged.

3.4 Budget Control and Applicability

CBR operates under a fixed interaction budget. For minibatch $\mathcal{X}$, the additional rollout pool is

$$P = \lfloor \rho N |\mathcal{X}| \rfloor . \tag{12}$$

Each eligible group receives at most b additional requests. All main rollouts, CBR rollouts, and rejected proposals are counted within the same generation budget; prefix replay is counted as environment cost.

CBR relies on three conditions: comparable anchor matching, reachable alternative actions, and informative terminal rewards. When matching is unreliable, intervention cannot reach the target anchor, or the policy lacks alternative support, the acquired contrast may be limited. CBR therefore introduces no process supervision, learned critic, or inference-time mechanism; it only changes where additional exploration budget is allocated.

4 Experiments

We test whether failure-conditioned consensus identifies useful exploration targets, whether targeted re-rollouts recover outcome contrasts, and whether these contrasts improve learning under a fixed generation budget.

4.1 Experimental Setup

Tasks and models. We evaluate Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct on ALFWorld and WebShop (Shridhar et al., 2021; Yao et al., 2022). Interaction horizons are 50 and 15 steps, respectively; success uses the environment's terminal verifier. Train/dev/test/diagnostic tasks are disjoint, with 800 diagnostic tasks reserved per environment. Methods share initialization, prompts, action parsing, evaluation, and common optimization settings. We evaluate the fixed-budget endpoint over five training seeds.

Sampling and budgets. We use $N = 8$, temperature 1.0, uniform consensus-anchor selection, $b = 2$, and $\rho = 0.10$. All training methods receive 12.0B generated tokens (matching tolerance $\pm 5\%$), including additional rollouts and rejected proposals. Environment steps, including prefix re-execution, and GPU-hours are reported separately. The 1.5B/7B runs use 16/32 H800 GPUs. Implementation and sampling corrections appear in Appendices A and B.

Comparisons and statistics. GiGPO retains the same two-level credit estimator and allocates additional sampling uniformly. Dynamic sampling and entropy branching implement DAPO- and ARPO-style allocation; Prefix Sampling uses its hard-side adaptation. Diagnostic intervals use 10,000 task-cluster BCa bootstrap resamples, with sampling rules paired on the same groups and policy snapshots (Efron, 1987; Agarwal et al., 2021). Training results report mean $\pm$ sample standard deviation across seeds; contrast intervals use paired hierarchical BCa resampling over tasks and seeds (Appendix C).

4.2 Failed Consensus Contains Recoverable Alternatives

Consensus is common among the remaining failures. At 1.5B, 18.5% of ALFWorld groups and 27.8% of WebShop groups fail completely; 82.4% and 78.0% of these groups contain consensus anchors. At 7B, complete failure falls to 8.9% and 19.6%, while consensus occurs in 87.3% and 82.8% of remaining failures. Repeated agreement remains prevalent as model size increases.

Alternatives reveal the value hidden by agreement. At a fixed checkpoint, let $Q^\pi(h, a)$ denote success probability after action a at history h, followed by π. We screen policy-proposed alternatives, select $\widehat{a}$, and use independent continuations to evaluate the selected-candidate gain:

$$G_{\text{selected}}(h) = Q^\pi(h, \widehat{a}) - Q^\pi(h, a_{\text{ref}}), \tag{13}$$

where a_{ref} is the observed reference action, from the consensus class at consensus anchors. Candidate screening is used only for this offline diagnostic; training selects anchors uniformly.

In Table 1, failed consensus anchors have the largest mean gain (0.180), exceeding failed non-consensus anchors (0.121) and non-stuck consensus anchors (0.044). The share with estimated gains above 0.03 is also highest (31.6%). Failure and agreement together identify richer exploration opportunities.

Table 1: **Failed consensus concentrates recoverable alternatives.** Offline selected-candidate gains in success probability, with 95% task-cluster bootstrap CIs. The last column is the percentage of anchors with estimated gain above 0.03.

Group	Anchor	Count	Mean gain [95% CI]	$\widehat{G} > 0.03$
Stuck	Consensus	1,624	0.180 [0.153, 0.208]	31.6%
Stuck	Non-consensus	1,102	0.121 [0.096, 0.148]	22.1%
Non-stuck	Consensus	1,600	0.044 [0.028, 0.061]	9.8%
Non-stuck	Non-consensus	1,600	0.057 [0.040, 0.075]	11.6%

Frequent actions can admit valuable alternatives. Mean gain rises from 0.106 to 0.171 and 0.263 across increasing thirds of consensus-action probability. An action can therefore be selected frequently while a better continuation remains available. Full stratification appears in Appendix D.

4.3 Targeted Re-rollouts Recover More Outcome Contrasts

We compare five sampling rules on the same 800 stuck groups at fixed policy snapshots. Branching rules share the alternative-action sampling procedure and change the target selector. Rescue measures the probability of at least one successful additional rollout within b requests:

$$\text{Rescue}_b(\mathcal{R}) = \Pr_{\mathcal{R}}\left[\exists j \le b : R_j^{\text{extra}} = 1 \mid \text{Stuck}(x) = 1\right]. \tag{14}$$

A rescue supplies the missing success–failure comparison. Failed arrival and exhausted proposals remain in the evaluation denominator.

Table 2: **Consensus targeting recovers more groups across request allowances.** Rescue rates (%) on the same 800-group diagnostic cohort. The final column is CBR minus each rule at $b = 2$, in percentage points, with 95% task-bootstrap CIs. Costs appear in Appendix D.2.

Sampling rule	$b = 1$	$b = 2$	$b = 4$	CBR advantage at $b = 2$
Ordinary retry	8.4	15.6	26.1	24.0 [19.1, 28.7]
Higher-temperature retry	11.8	20.8	32.8	18.8 [13.8, 23.2]
Random anchor	14.1	24.5	38.8	15.1 [9.9, 19.7]
Entropy anchor	16.0	27.3	42.4	12.3 [7.0, 16.9]
Consensus anchor	**24.8**	**39.6**	**56.9**	—

In Table 2, two consensus-targeted requests rescue 39.6% of groups, versus 15.6% for ordinary retry and 20.8% for higher-temperature retry. Locating the missing comparison adds 18.8 points

Table 3: **CBR improves held-out success at matched generation budgets.** Success (%, mean ± standard deviation over five seeds) at 12.0B tokens. Best means are bold. GiGPO is the shared base learner.

	Qwen2.5-1.5B		Qwen2.5-7B	
Method	ALFWorld	WebShop	ALFWorld	WebShop
GRPO (Shao et al., 2024)	72.8 ± 3.6	56.8 ± 3.8	77.6 ± 5.2	66.1 ± 3.7
Uniform extra sampling	75.4 ± 2.8	59.1 ± 3.1	80.5 ± 3.8	68.4 ± 3.0
Dynamic sampling (Yu et al., 2025)	79.8 ± 2.5	61.7 ± 2.8	84.3 ± 3.2	71.2 ± 2.6
Entropy branching (Dong et al., 2026)	87.6 ± 1.9	66.5 ± 2.4	93.5 ± 2.0	75.4 ± 2.2
Prefix Sampling (hard-side) (Zhu et al., 2026)	88.8 ± 1.8	66.9 ± 2.5	93.2 ± 1.5	75.1 ± 2.1
EMPG (Wang et al., 2025a)	73.7 ± 2.7	60.8 ± 1.3	78.5 ± 1.7	69.3 ± 0.5
HGPO (He et al., 2026)	90.7 ± 1.4	67.8 ± 2.4	96.0 ± 1.0	79.3 ± 1.9
STAPO (Qi et al., 2026)	92.2 ± 1.2	69.0 ± 2.5	96.9 ± 1.6	77.6 ± 3.9
BiPACE-Q (Wang et al., 2026)	$\mathbf{93.5} \pm 1.2$	70.1 ± 2.0	$\mathbf{97.1} \pm 0.9$	79.8 ± 1.6
GiGPO (Feng et al., 2025)	86.7 ± 1.7	65.0 ± 3.2	90.8 ± 1.3	72.8 ± 3.2
CBR	93.4 ± 1.4	$\mathbf{72.6} \pm 1.8$	95.2 ± 1.1	$\mathbf{80.9} \pm 1.6$

beyond increasing sampling temperature. Consensus targeting also exceeds entropy branching by 12.3 points (95% CI: $[7.0, 16.9]$), demonstrating useful allocation information in failed agreement.

The advantage holds at every request allowance. One consensus-targeted request rescues 24.8% of groups, approaching four ordinary retries (26.1%); two approach four entropy-guided branches (42.4%). Targeting is especially useful when few additional attempts are affordable. Arrival statistics, alternative selectors, and the diagnostic oracle appear in Appendix D.2.

4.4 Recovered Comparisons Improve Agent Learning

Gains hold across environments and model sizes. Table 3 shows gains over GiGPO of 6.7, 7.6, 4.4, and 8.1 points across the four settings. The equal-weight mean of these within-setting differences is 6.7 points (95% CI: $[5.2, 8.0]$). With the two-level credit estimator retained, this comparison measures the benefit of changing the acquired evidence.

CBR consistently improves the shared GiGPO learner and remains competitive with recent state- and action-aware credit-assignment methods. Its largest gains occur on WebShop, whereas performance is comparable to strong source-reported baselines in the near-saturated ALFWorld settings.

Improvements extend beyond the revisited training tasks. On ALFWorld/1.5B, held-out success rises from GiGPO's 86.7% to 93.4%. We report training and held-out results separately and avoid interpreting a smaller train–test gap as direct evidence of generalization.

Recovered successes persist as training progresses. From the quarter-budget to final checkpoint, CBR's stuck-group rate falls from 20.1% to 5.8%, versus GiGPO's final 10.9% (Table 4). Among rescued tasks, success at the next visit rises from 31.6% to 68.4%, supporting lasting policy improvement. Fewer remaining stuck groups also reduce the number of groups requiring targeted exploration.

4.5 Target Selection and Credit Assignment Play Distinct Roles

Table 5 pairs held-out success with training-time rescue on ALFWorld/1.5B at 12.0B tokens. Training-time rescue is measured separately from the fixed-policy cohort in Table 2.

Both failure conditioning and consensus targeting matter. Ordinary and higher-temperature retries lower success by 5.6 and 4.4 points; random and entropy anchors lower it by 5.0 and 3.8 points. Breaking consensus in all groups costs 2.6 points. Conditioning on failure focuses the fixed exploration budget on groups that still lack a successful comparison.

Table 4: **Fewer groups remain stuck, and recovered success becomes more persistent.** ALFWorld/1.5B at fractions of the 12.0B-token budget. Persistence is success at the next visit among previously rescued tasks. All entries except budget fraction are percentages. Full intervention statistics appear in Appendix D.2.

	Stuck groups ↓		Test success ↑		CBR persistence ↑
Budget	GiGPO	CBR	GiGPO	CBR	Next visit
25%	23.6	20.1	49.7	60.8	31.6
50%	18.8	12.7	68.9	80.9	48.9
75%	14.2	7.9	79.8	89.7	60.8
100%	10.9	5.8	86.7	93.4	68.4

Table 5: **Targeting acquires useful outcomes; anchor credit improves their use.** ALFWorld/1.5B ablations at 12.0B generated tokens. Rescue and held-out success are percentages. The final column gives the success difference from full CBR in percentage points and its 95% CI. Sample-count reweighting equalizes the loss mass contributed by additional samples.

Variant	Rescue	Test success	Difference from CBR [95% CI]
Full CBR	43.1	93.4	—
Ordinary retry	12.5	87.8	-5.6 $[-7.4, -3.9]$
Higher-temperature retry	17.6	89.0	-4.4 $[-6.1, -2.8]$
Random anchor	13.2	88.4	-5.0 $[-6.8, -3.3]$
Entropy anchor	16.9	89.6	-3.8 $[-5.5, -2.2]$
All-group breaking	29.5	90.8	-2.6 $[-4.2, -1.1]$
No anchor credit	42.8	91.0	-2.4 $[-3.9, -1.0]$
Sample-count reweighting	42.7	92.4	-1.0 $[-2.1, 0.1]$

Local credit helps absorb the acquired comparisons. Removing anchor credit preserves rescue (42.8% versus 43.1%) but lowers test success by 2.4 points: exposing the local comparison to the optimizer improves its use. Equalizing additional samples' loss mass retains 92.4% success, 5.7 points above GiGPO. Extended ablations appear in Appendix D.2.

Resource use. On ALFWorld/1.5B, CBR reaches 93.4% success using 262 GPU-hours, compared with GiGPO's 86.7% at 246 GPU-hours. Relative to GiGPO, the 6.7-point improvement costs 6.5% more GPU time at the same generation budget. Environment steps increase from 2.88M to 2.96M (2.8%). Detailed interaction, training-token, latency, and memory measurements appear in Appendix E; additional environments and perturbation analyses appear in Appendices E and F.

5 Conclusion

We identify consensus blind spots in long-horizon agent learning: all-failure rollout groups can exhibit repeated action agreement while providing neither reward variation nor alternative-action evidence at shared decision points. CBR uses this missing comparison to direct exploration and supplies the resulting outcomes to existing two-level group-relative learning. Across ALFWorld and WebShop, the diagnostics, targeting comparisons, and training results support failure-conditioned consensus as a signal for recovering useful contrasts and improving held-out performance.

The benefit depends on reachable and comparable anchors, informative terminal rewards, and useful alternatives within policy support. The evidence supports targeted contrast acquisition without establishing exact causal credit for individual actions. Extending anchor matching to stochastic and partially observed environments is a natural next step. More broadly, these findings suggest that repeated agreement within failed rollouts can guide where new evidence should be collected: an untested alternative may remain valuable even when observed actions show no disagreement.

AI USE STATEMENT

Generative AI tools were used only to edit the manuscript for grammar, clarity, readability, and academic style. They were not used to formulate the research questions or hypotheses, design the method or experiments, implement the proposed method, generate or process experimental data, formulate mathematical claims, interpret the results, or produce the reported empirical findings. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of this work, including all text, claims, results, and artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We provide the information needed to reproduce the proposed method and evaluate the reported results throughout the paper and appendix. The algorithm, intervention mechanism, credit-assignment procedure, and budget-control rules are described in Sections 3.1–3.4. The experimental setup is reported in Section 4.1. Implementation and training details, the complete evaluation and statistical protocols, extended results, resource accounting, and robustness analyses are provided in Appendices A, C, D, E, and F, respectively. Appendix F.3 further specifies the task splits, random seeds, checkpoints, environment versions, prompts, normalization rules, training configurations, and aggregation procedures used for the reported experiments.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/f514cec81cb148559cf475e7426eed5e-Abstract.html.

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Marc G. Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Rémi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems*, volume 29, pp. 1471–1479, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/afda332245e2af431fb7b672a68b659d-Abstract.html.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=H1lJJnR5Ym.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, Guorui Zhou, Yutao Zhu, Ji-Rong Wen, and Zhicheng Dou. Agentic reinforced policy optimization. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=TX4k7BF6aO.

Adrien Ecoffet, Joost Huizinga, Joel Lehman, Kenneth O. Stanley, and Jeff Clune. First return, then explore. *Nature*, 590:580–586, 2021. doi: 10.1038/s41586-020-03157-9. URL https://doi.org/10.1038/s41586-020-03157-9.

Bradley Efron. Better bootstrap confidence intervals. *Journal of the American Statistical Association*, 82(397):171–185, 1987. doi: 10.1080/01621459.1987.10478410. URL https://doi.org/10.1080/01621459.1987.10478410.

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*, 2025. URL https://openreview.net/forum?id=QXEhBMNrCW.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

Shuo He, Lang Feng, Qi Wei, Xin Cheng, Lei Feng, and Bo An. Hierarchy-of-groups policy optimization for long-horizon agentic tasks. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=T8Dev99qnz.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. TreeRL: LLM reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.604. URL https://aclanthology.org/2025.acl-long.604/.

Jiazheng Li, Yawei Wang, Qiaojing Yan, Yijun Tian, Zhichao Xu, Huan Song, Panpan Xu, and Lin Lee Cheong. SALT: Step-level advantage assignment for long-horizon agents via trajectory graph. In *Findings of the Association for Computational Linguistics: EACL 2026*, pp. 4709–4725. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.findings-eacl.247. URL https://aclanthology.org/2026.findings-eacl.247/.

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.

Xufang Luo, Yuge Zhang, Zhiyuan He, Zilong Wang, Siyun Zhao, Dongsheng Li, Luna K. Qiu, and Yuqing Yang. Agent lightning: Train any AI agents with reinforcement learning. *arXiv preprint arXiv:2508.03680*, 2025. URL https://arxiv.org/abs/2508.03680.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances in Neural Information Processing Systems*, volume 29, pp. 4026–4034, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8d8818c8e140c64c743113f563cf750f-Abstract.html.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 2778–2787. PMLR, 2017. URL https://proceedings.mlr.press/v70/pathak17a.html.

Qiuyi Qi, Tian Liang, Mutian Bao, Jinjian Zhang, Dongnan Liu, Wei Zhou, Linjian Mo, Ming Kong, Jie Liu, Feng Zhang, and Qiang Zhu. STAPO: Selective trajectory-aware policy optimization for LLM agent training. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28371–28392. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1308. URL https://aclanthology.org/2026.acl-long.1308/.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 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. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

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://openreview.net/forum?id=0IOX0YcCdTn.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 2 edition, 2018. ISBN 9780262039246. URL https://mitpress.mit.edu/9780262039246/reinforcement-learning/.

Hanyang Wang, Weijieying Ren, Yuxiang Zhang, Ding Cao, Zhizhao Zeng, Ke Zeng, and Tianxiang Zhao. BiPACE: Bisimulation-guided policy optimization with action counterfactual estimation for LLM agents. *arXiv preprint arXiv:2606.25556*, 2026. URL https://arxiv.org/abs/2606.25556.

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.

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.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentPRM: Process reward models for LLM agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025. URL https://arxiv.org/abs/2511.08325.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. WebShop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1508. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/82ad13ec01f9fe44c01cb91814fd7b8c-Abstract-Conference.html.

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. *arXiv preprint arXiv:2503.14476*, 2025. URL https://arxiv.org/abs/2503.14476.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62138–62160. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhou24r.html.

Tianshu Zhu, Wenyu Zhang, Xiaoying Zuo, Lun Tian, Haotian Zhao, Yucheng Zeng, Jingnan Gu, Daxiang Dong, Jianmin Wu, Dawei Yin, and Dou Shen. Rollout pass-rate control: Steering binary-reward RL toward its most informative regime. *arXiv preprint arXiv:2605.05112*, 2026. URL https://arxiv.org/abs/2605.05112.

Zefang Zong, Dingwei Chen, Yang Li, Qi Yi, Bo Zhou, Chengming Li, Bo Qian, Peng Chen, and Jie Jiang. $\mathrm{AT}^2\mathrm{PO}$: Agentic turn-based policy optimization via tree search. *arXiv preprint arXiv:2601.04767*, 2026. URL https://arxiv.org/abs/2601.04767.

A Implementation and Training Procedure

A.1 Anchors and Alternative Rollouts

Exact-prefix anchors match complete interaction histories. Signature anchors match a task identifier and normalized observation; actions are normalized through the environment parser or canonical tool names and arguments. Detection uses the original group's histories, actions, and terminal rewards. An eligible anchor has at least $n_{\min}$ visits and one observed normalized action class in an all-failure group. Support counts visits, including repeated visits within a trajectory.

CBR selects one eligible anchor uniformly. In replay mode (R), it resets the task with the original seed and executes the cached action prefix. Replay reliability is assessed by comparing observations and terminal outcomes across repeated executions. In forward-matching mode (F), it samples an ordinary rollout until the selected signature is encountered. A completed rollout that misses the anchor remains an ordinary additional sample. Signature matching compares observations; its sensitivity to hidden-state differences is evaluated in Appendix F.

At the reached history, CBR samples up to four proposals and accepts the first whose normalized action differs from the consensus class. Temperature remains 1.0. After acceptance, generation follows the rollout policy for the remaining episode horizon. Exhausting the proposal cap counts as an unsuccessful intervention. Rejected proposals incur generation cost; prefix replay incurs environment cost.

A.2 Budgeted Training

For a minibatch $\mathcal{X}$, the request pool is $P = \lfloor \rho N |\mathcal{X}| \rfloor$. Eligible groups receive at most b requests each until this pool is exhausted; unused requests do not carry into the next minibatch. The generation budget additionally caps the total tokens consumed by main sampling, additional sampling, and rejected proposals.

Each update follows five steps:

1. Sample N main rollouts per task and collect rewards, histories, normalized actions, and action log-probabilities.
2. Identify all-failure groups and construct their eligible anchors.
3. Select anchors uniformly and generate additional rollouts within the request pool, including arrival and intervention outcomes.
4. Append completed rollouts to their original groups and recompute episode- and anchor-level advantages on the enlarged groups.
5. Apply local reweighting at intervened actions and update the policy with the base learner's clipping, KL regularization, and training mask.

Offline diagnostic continuations are excluded from these updates.

Table 6: Primary configuration. The pool ratio ρ controls requests; actual generated-token consumption is accounted for separately.

Parameter	Value
Main group size N / per-group request cap b	8 / 2
Request-pool ratio ρ	0.10
Stuck criterion	All main rollouts fail
Minimum anchor support $n_{\min}$	$\max(2, \lceil N/2 \rceil)$
Anchor selector / proposal cap	Uniform / 4
Anchor-credit weight λ / discount γ	1 / 0.95
Sampling temperature / temperature fallback	1.0 / disabled
Generated-token budget / matching tolerance	12.0B / $\pm 5\%$
Main training seeds	5
Hardware, 1.5B / 7B	16 / 32 H800 GPUs

B Missing Comparisons and Local Reweighting

B.1 From Failed Agreement to Learning Signals

Let $\mathcal{G}^+$ denote an enlarged rollout group and $\mathcal{V}_x^+(s)$ its visits to anchor s. The two-level estimator is

$$A_i^{\mathrm{ep},+} = \frac{R_i - \mu_{\mathcal{G}^+}}{s_{\mathcal{G}^+} + \epsilon}, \tag{15}$$

$$A_{i,t}^{\mathrm{anc}} = \frac{G_{i,t} - \mu_{\sigma(h_{i,t})}}{s_{\sigma(h_{i,t})} + \epsilon}, \qquad G_{i,t} = \gamma^{T_i - t} R_i, \tag{16}$$

$$A_{i,t} = A_i^{\mathrm{ep},+} + \lambda A_{i,t}^{\mathrm{anc}}. \tag{17}$$

Means and standard deviations are computed over the corresponding group or anchor visits. Before augmentation, an all-failure group has $R_i = G_{i,t} = 0$, so both reward-derived advantages vanish. Consensus adds a separate property: the observed action set at an anchor has cardinality one. CBR targets the conjunction of missing reward variation and missing action alternatives.

A successful additional rollout creates episode-level reward variation. Matched visits with different returns also create anchor-level credit. When all additional outcomes remain zero, both reward-derived advantages remain zero; KL regularization can still contribute to the update. Finite-sample consensus identifies an untested comparison, rather than determining policy entropy or the cause of failure.

B.2 Finite Rejection Sampling

At fixed history h, define the accepted action set $E = \{a : \nu(a) \neq a_{\mathrm{con}}\}$ and $p = p_{\mathrm{con}}(h) = \Pr_{\pi_{\mathrm{old}}}(a \notin E \mid h)$. With independent proposals and cap L,

$$\Pr(\mathrm{accept}) = 1 - p^L. \tag{18}$$

For $B \subseteq E$ and $p < 1$,

$$\Pr(a_{\mathrm{accepted}} \in B, \mathrm{accept}) = \sum_{\ell=1}^{L} p^{\ell-1} \pi_{\mathrm{old}}(B \mid h), \tag{19}$$

$$\Pr(a_{\mathrm{accepted}} \in B \mid \mathrm{accept}) = \frac{\pi_{\mathrm{old}}(B \mid h)}{1 - p}. \tag{20}$$

Thus, the cap controls coverage while accepted actions follow the policy conditioned on excluding the consensus class. When additional validity filters are applied, they become part of the conditioning set. Exhaustion denotes failure within the finite proposal allowance.

B.3 Local Reweighting and the Training Objective

For an accepted action, $q_{\mathrm{old}}(a \mid h) = \pi_{\mathrm{old}}(a \mid h)/(1-p)$. Consequently,

$$\mathbb{E}_{a \sim q_{\mathrm{old}}}[(1-p) f(a)] = \mathbb{E}_{a \sim \pi_{\mathrm{old}}}[\mathbb{I}[a \in E] f(a)] \tag{21}$$

for integrable f at this history. The primary implementation uses

$$\widetilde{A}_{i,t} = \begin{cases} (1 - \widehat{p}_{\mathrm{con}}(h_{i,t})) A_{i,t}, & \text{intervened action,} \\ A_{i,t}, & \text{otherwise.} \end{cases} \tag{22}$$

The estimate uses the sequence probability of the observed consensus surface form evaluated at the intervention history. Since several surface forms can share one normalized class, this estimate lower-bounds the class probability and produces weaker attenuation than the exact class-probability multiplier. Advantages and probability estimates are held fixed during optimization.

For action token $y_{i,t,k}$, let $r_{i,t,k}(\theta)$ be its probability under π_θ divided by that under π_{old}, conditioned on $h_{i,t}$ and preceding action tokens. The policy loss is

$$\mathcal{L}_{\mathrm{policy}} = -\mathbb{E}_{(i,t,k) \in \mathcal{T}_{\mathrm{loss}}} \Big[\min \big\{ r_{i,t,k} \widetilde{A}_{i,t}, \; \mathrm{clip}(r_{i,t,k}, 1-\varepsilon, 1+\varepsilon) \widetilde{A}_{i,t} \big\} \Big], \tag{23}$$

(Schulman et al., 2017) where $\mathcal{T}_{\text{loss}}$ is the base learner's token mask. The behavior-log-probability ablation instead supplies the conditioned behavior distribution q_{old} to the clipped objective. At action level, its ratio is $\pi_\theta(a \mid h)/q_{\text{old}}(a \mid h)$; moving a factor inside clipping differs from multiplying the advantage.

Equation equation 21 concerns the non-consensus contribution at one fixed history. Anchor selection and altered future histories remain part of the intervention distribution, so the complete clipped update is a locally reweighted training procedure. Similarly, centered episode advantages cancel at a shared prefix only with identical contexts and gradient weights across the complete group. The anchor-credit ablation measures its contribution under the actual training objective.

C Evaluation and Statistical Protocols

Populations and checkpoints. Training, development, test, and diagnostic tasks are disjoint. E1 measures natural rollout groups and anchor opportunities; E2 compares sampling rules on 800 stuck groups; E3 evaluates trained policies on held-out tasks. Group and anchor counts are nested observations, with resampling clustered by task identity. Main learning results use the 12.0B-token endpoint and five seeds. The diagnostic pool contains 800 tasks per environment; its natural-group prevalence and E2's selected stuck-group cohort have different denominators.

Independent candidate evaluation. E1 proposes up to $K = 4$ alternatives, screens each with $M_1 = 8$ continuations, and evaluates the selected alternative and reference action with independent $M_2 = 32$ confirmation continuations each. The estimand is G_{selected} in Eq. equation 13; the practical gain threshold is 0.03. Arrival failures contribute zero recoverable gain. Candidate screening outcomes are separated from confirmation outcomes and training updates.

Uncertainty and comparisons. E1 uses 10,000 task-cluster BCa bootstrap resamples. E2 preserves pairing across rules and uses paired task-cluster BCa intervals. E3 uses paired hierarchical BCa resampling over tasks and seeds. The primary learning contrast is CBR minus GiGPO. Timeouts and invalid actions count as failed requests. Each reproduced configuration is aggregated over five seeds.

Rescue, persistence, and cost. Rescue is measured per originally stuck group, including unsuccessful requests. Persistence is success at the next visit among previously rescued tasks. Fixed-policy E2 rescue and training-time rescue are analyzed separately. For n visits with k successful outcomes, $Y = k(n-k)/n^2$ measures anchor-level binary reward dispersion. It uses an anchor denominator, distinct from group-level rescue. Generated tokens, environment steps, and tokens admitted to the loss form separate resource accounts. GPU-hours equal GPU count times wall time, before rounding. Diagnostic continuation costs are separate from training costs.

D Extended Diagnostic and Learning Results

D.1 Where Alternative-Action Value Occurs

Table 7 expands the phenomenon analysis. Higher consensus probability is associated with larger selected gains, while earlier anchors and longer remaining horizons offer more recovery opportunity. Both anchor representations expose useful alternatives.

Table 7: Selected-candidate gains within failed consensus anchors. Counts are anchors; gains and task-cluster 95% CIs are in success-probability units.

Variable	Bin	Count	Gain	95% CI
Consensus probability	Low	542	.106	[.079,.134]
	Middle	541	.171	[.141,.202]
	High	541	.263	[.222,.304]
Relative depth	Early	552	.205	[.173,.238]
	Middle	540	.174	[.144,.204]
	Late	532	.157	[.127,.188]
Remaining horizon	Short	536	.126	[.099,.153]
	Middle	548	.175	[.145,.205]
	Long	540	.241	[.203,.278]
Anchor type	Exact prefix	984	.190	[.165,.215]
	Signature	640	.164	[.133,.195]

Potential value and acquisition difficulty vary differently. In the controlled mixed-effects analysis, consensus probability has a negative association with online rescue (standardized logit coefficient -0.31, 95% CI $[-0.45, -0.18]$), controlling for task difficulty, depth, remaining horizon, anchor type, and environment. High-probability consensus can hide valuable alternatives while making them harder to sample within a finite proposal cap, as Eq. equation 18 predicts.

D.2 Selectors and Realized Sampling Costs

Table 8 extends the main sampling comparison with earliest-anchor, cursor, and diagnostic-oracle selection. At $b = 2$, earliest and cursor selection exceed uniform selection by 1.7 points (95% CI $[-2.0, 5.6]$) and 1.2 points ($[-2.4, 5.0]$). Uniform selection remains the primary configuration. The oracle ranks anchors using offline outcome estimates and achieves 52.8% rescue; its additional information is excluded from CBR training.

Table 8: Diagnostic rescue (%) across request allowances. All rules use the same 800 stuck groups. The oracle uses additional offline outcome information.

Rule	$b = 1$	$b = 2$	$b = 4$
Retry	8.4	15.6	26.1
Higher-temperature retry	11.8	20.8	32.8
Random anchor	14.1	24.5	38.8
Entropy anchor	16.0	27.3	42.4
Consensus-uniform	24.8	39.6	56.9
Consensus-earliest	25.9	41.3	58.0
Consensus-cursor	25.3	40.8	59.1
Diagnostic oracle	34.3	52.8	70.0

Table 9: Realized costs of branching rules at $b = 2$ on 800 stuck groups. Each rule requests 1,600 rollouts. Completed rollouts can be fewer than requests. Tokens per rescued group equal total generated tokens divided by the number of rescued groups.

Rule	Completed	Gen. tokens	Env. steps	Tokens/rescue
Random anchor	1,568	8.16M	.231M	41.6k
Entropy anchor	1,560	8.20M	.229M	37.6k
Consensus-uniform	1,576	8.31M	.237M	26.2k

The three rules in Table 9 have similar generation costs, while consensus targeting uses fewer tokens per rescued group. At $b = 2$, their reach rates are 91.2%, 90.5%, and 94.3%, respectively. These diagnostic summaries pool R/F modes; environment-specific arrival results are given below.

D.3 Selection, Dose, and Optimization Ablations

Table 10 complements the main component ablations. Increasing the joint request dose to $(b, \rho) = (4, 0.20)$ adds 0.8 success point and 20 GPU-hours. Removing local reweighting costs 2.8 points. Cursor selection, persistent memory, and EMPG composition have smaller gains with intervals spanning zero, supporting the use of the simpler uniform configuration.

Table 10: Extended ALFWorld/1.5B ablations. Success is in percent; differences from primary CBR are in percentage points with 95% CIs. All runs use 12.0B generated tokens unless indicated.

Variant	Success	GPU-hours	Difference [95% CI]
Primary CBR	93.4	262	—
Earliest anchor	92.1	260	-1.3 $[-2.5, -0.3]$
Cursor selector	93.7	267	$+0.3$ $[-0.6, 1.2]$
Persistent memory	93.6	265	$+0.2$ $[-0.7, 1.0]$
$b = 1, \rho = .05$	89.8	251	-3.6 $[-5.0, -2.3]$
$b = 4, \rho = .20$	94.2	282	$+0.8$ $[-0.1, 1.8]$
Uniform extra, 12.0B	75.4	242	-18.0 $[-19.9, -16.2]$
Uniform extra, 14.4B	78.6	286	-14.8 $[-16.7, -13.1]$
EMPG only	73.7	244	-19.7 $[-21.6, -17.9]$
CBR + EMPG	94.0	270	$+0.6$ $[-0.4, 1.6]$
Behavior-log-probability variant	92.9	268	-0.5 $[-1.5, 0.4]$
No local multiplier	90.6	257	-2.8 $[-4.2, -1.6]$

D.4 Intervention Dynamics and Trajectory Length

Table 11 supplements the main learning dynamics with intervention statistics. Arrival and breaking remain stable as the fraction of stuck groups declines; rescue among the remaining stuck groups increases from 37.8% to 43.1%.

Table 11: CBR intervention dynamics on ALFWorld/1.5B. Budget is the fraction of 12.0B tokens. Rates are percentages; anchors is the mean consensus-anchor count per group. Persistence uses the next visit.

Budget	Anchors	Reach	Break	Rescue	Unbreakable	Persistence
25%	4.1	94.0	92.5	37.8	7.5	31.6
50%	3.5	94.2	93.0	40.5	6.8	48.9
75%	2.8	94.5	93.4	42.0	5.7	60.8
100%	2.3	94.6	93.6	43.1	4.9	68.4

The separate trajectory-length analysis gives CBR-minus-GiGPO success differences of 1.2 points (95% CI $[-0.2, 2.8]$) for horizons ≤ 15, 5.1 ($[3.0, 7.4]$) for 16–30 steps, and 10.3 ($[7.5, 13.4]$) for more than 30 steps. These strata contain 350, 350, and 300 tasks. Larger gains accompany longer trajectories, where more decisions remain available for targeted exploration.

E Resource Use and Additional Environments

E.1 Separate Resource Accounts

Table 12 compares generation, interaction, and optimization costs on identical hardware. CBR uses the same 9.4B training tokens as GiGPO and adds 0.08M environment steps and 16 GPU-hours, while improving held-out success by 6.7 points.

Table 12: ALFWorld/1.5B resource use at 12.0B generated tokens on 16 H800 GPUs. Memory is peak GB per GPU; latency is batch p50/p95 in seconds. GPU-hours and rounded wall time are listed separately.

Method	Env. steps	Train tok.	GPU-h	Wall-h	GB	Latency
Entropy branching	2.94M	9.3B	258	16.1	73	23.4/49.8
EMPG	2.85M	9.3B	244	15.3	72	22.5/47.9
GiGPO	2.88M	9.4B	246	15.4	73	23.0/48.8
HGPO	2.88M	9.5B	252	15.8	74	23.8/50.9
STAPO	2.90M	9.5B	260	16.3	74	24.0/51.4
Prefix Sampling	2.96M	9.3B	255	15.9	73	23.6/50.1
BiPACE-Q	2.88M	9.6B	268	16.8	76	24.6/53.0
CBR	2.96M	9.4B	262	16.4	74	24.3/52.1

E.2 Arrival Modes across Environments

Table 13 separates replay-based main experiments from forward-matching extensions. The latter obtain gains of 6.6 and 6.1 points on TextCraft and Search-QA, respectively. Forward matching extends CBR to settings where a cached prefix cannot be reliably replayed.

Table 13: Environment-specific arrival and learning results. R/F denote replay/forward matching. Success and reach are percentages; differences are CBR minus GiGPO in percentage points with 95% CIs. The two F-mode extensions use three seeds on A100 hardware.

Environment	Model	Mode	Reach	GiGPO	CBR	Difference [95% CI]
ALFWorld	1.5B	R	94.6	86.7	93.4	6.7 [4.5, 8.6]
WebShop	1.5B	R	91.8	65.0	72.6	7.6 [4.8, 10.7]
ALFWorld	7B	R	96.2	90.8	95.2	4.4 [2.5, 6.0]
WebShop	7B	R	93.5	72.8	80.9	8.1 [5.3, 10.9]
TextCraft	1.5B	F	87.1	54.6	61.2	6.6 [3.8, 9.7]
Search-QA	Qwen3-4B	F	84.0	31.4	37.5	6.1 [3.4, 8.7]

The 1.5B and 7B entries use Qwen2.5-Instruct models. TextCraft is the ALFWorld-to-TextCraft out-of-distribution evaluation; the search setting uses the held-out BrowseComp-Plus/Search-QA query set. Hardware timings from these A100 extensions are kept separate from the H800 comparisons.

F Boundary Conditions and Failure Analysis

F.1 Reliability of the Acquired Comparisons

Table 14 perturbs anchor comparability, replay reliability, and reward quality separately. At 20% severity, CBR retains gains of 2.9, 2.7, and 4.1 points. At 30%, the gains fall below one point and their intervals span zero. The degradation locates the practical boundary in the reliability of the acquired comparisons; the tested grid does not specify an exact failure threshold.

Table 14: ALFWorld/1.5B sensitivity at the fixed generation budget. Severity denotes controlled aliasing, transition noise, or reward-label flips. Rescue and test success are percentages. Differences from GiGPO are in percentage points with 95% CIs.

Perturbation	Severity	Rescue	CBR	GiGPO	Difference [95% CI]
Unperturbed	0%	43.1	93.4	86.7	6.7 [4.5, 8.6]
Signature aliasing	10%	38.5	91.8	86.6	5.2 [3.5, 6.8]
	20%	30.4	89.3	86.4	2.9 [1.1, 4.6]
	30%	22.0	86.9	86.3	0.6 [−1.2, 2.3]
Replay stochasticity	10%	37.6	91.5	86.5	5.0 [3.2, 6.6]
	20%	27.8	88.9	86.2	2.7 [0.9, 4.4]
	30%	18.6	86.5	86.0	0.5 [−1.4, 2.2]
Reward noise	10%	39.8	91.7	85.9	5.8 [4.0, 7.4]
	20%	32.5	88.9	84.8	4.1 [2.2, 5.9]
	30%	22.9	84.4	83.7	0.7 [−1.2, 2.5]

F.2 Where Interventions Fail

Table 15 distinguishes failure to obtain an action comparison from failure to obtain a successful continuation or retain subsequent success. The most common category is an accepted alternative without rescue, directing attention to anchor relevance and the quality of the continuation policy.

Table 15: Multi-label failure analysis over 1,000 stuck groups and 800 targeted groups. Categories can overlap; percentages use the full stuck-group denominator.

Stage	Category	Count	All stuck (%)
Detection	No consensus anchor	170	17.0
Arrival	Anchor not reached	47	4.7
Comparability	Signature collision / invalid comparison	24	2.4
Proposal	Rejection exhausted	64	6.4
Outcome	Alternative sampled without rescue	404	40.4
Learning	Rescue without next-visit success	71	7.1

CBR is most useful when anchors identify comparable decisions, policy support contains reachable alternatives, and terminal outcomes provide informative feedback. Signature aliasing weakens comparison; concentrated policy support limits proposal acceptance; later errors can erase the benefit of an alternative action. The evaluated scope is text-based agents with discrete or normalizable actions and verifiable terminal outcomes.

F.3 Reproducibility Details

For each aggregate, we use fixed task splits, random seeds, checkpoints, environment versions, prompts, parser and normalization rules, training configuration, and aggregation procedures. We distinguish rollout requests from completed samples and report reached anchors, rejected proposals, accepted actions, and environment-verifier outcomes separately. For ALFWorld, we additionally report the split name and per-seed integer success count. Resource accounting separates main sampling, additional sampling, prefix replay, and offline diagnostics under the definitions used in the main experiments.