RMPO: REPLAY-MEASURED TURN-LEVEL CREDIT FOR LLM AGENT TRAINING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-turn agent training often relies on sparse episode rewards, making it hard to determine which actions contributed to the final outcome. Standard solutions either infer credit from trajectory statistics or rely on compute-heavy rollout sampling. In this paper, we show that credit can be measured more directly: by restoring the execution state in the environment, applying a localized change to the action, and evaluating the difference in outcome. We formulate this procedure as a common replay interface and propose *Replay-Measured Policy Optimization* (RMPO), a critic-free method that adds normalized replay measurements to the episode-level advantage. Depending on what the environment supports, RMPO implements three pragmatic credit operators: leave-one-out action deletion, prefix-progress scoring, and same-state action replacement. We evaluate RMPO in ALFWorld, WebShop, and Search-QA with Qwen3-4B and Qwen3-8B. It consistently outperforms GRPO and GiGPO. In ALFWorld, deletion measurements correlate strongly with symbolic-planner progress ($\rho = 0.842$), and RMPO exceeds four-continuation sampling at equal GPU-hours (90.5% versus 85.5%). The method requires a resettable replay environment. Code is available at `https://anonymous.4open.science/r/rmpo-CD48`.

1 INTRODUCTION

Consider an agent asked to put a heated apple on a table. Picking up the apple is necessary for completing the task, while opening an unrelated cabinet is not. Yet both actions may appear in a trajectory that receives the same episode reward. When the same episode-level advantage is assigned to every turn, the learning signal cannot distinguish actions that contributed to success from actions that merely occurred along the way.

Group-based reinforcement learning is attractive for agent training because it provides relative supervision without learning a separate critic (Shao et al., 2024; Ahmadian et al., 2024). However, an episode-level advantage does not resolve which decisions within a trajectory were useful. Recent methods address this problem by estimating turn-level credit from observed rollouts (Feng et al., 2025; Zhu et al., 2026; Li et al., 2026; Zheng et al., 2026), while VinePPO samples additional policy continuations from intermediate states (Kazemnejad et al., 2025). These approaches infer the contribution of an earlier action from rollout statistics or additional policy samples.

When the environment can be restored and replayed, we can obtain a more direct signal about an individual decision. For example, if removing "take apple" causes the recorded plan to fail while removing "open cabinet" leaves the outcome unchanged, replay reveals that the two actions played different roles in that trajectory. Figure 1 illustrates this idea in ALFWorld.

Building on this idea, we propose *Replay-Measured Policy Optimization* (RMPO). RMPO uses environment replay to measure how the evaluated outcome changes around a recorded turn, and uses this measurement to refine the episode-level advantage into turn-level credit. Depending on the environment, the replay comparison can remove an action, measure progress across prefixes, or compare the recorded action with executable alternatives. The same normalization and aggregation rule is used across all three cases.

We evaluate RMPO on ALFWorld, WebShop, and Search-QA using Qwen3-4B and Qwen3-8B. Across both model scales, RMPO consistently outperforms GRPO and GiGPO; at 8B, it reaches

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.

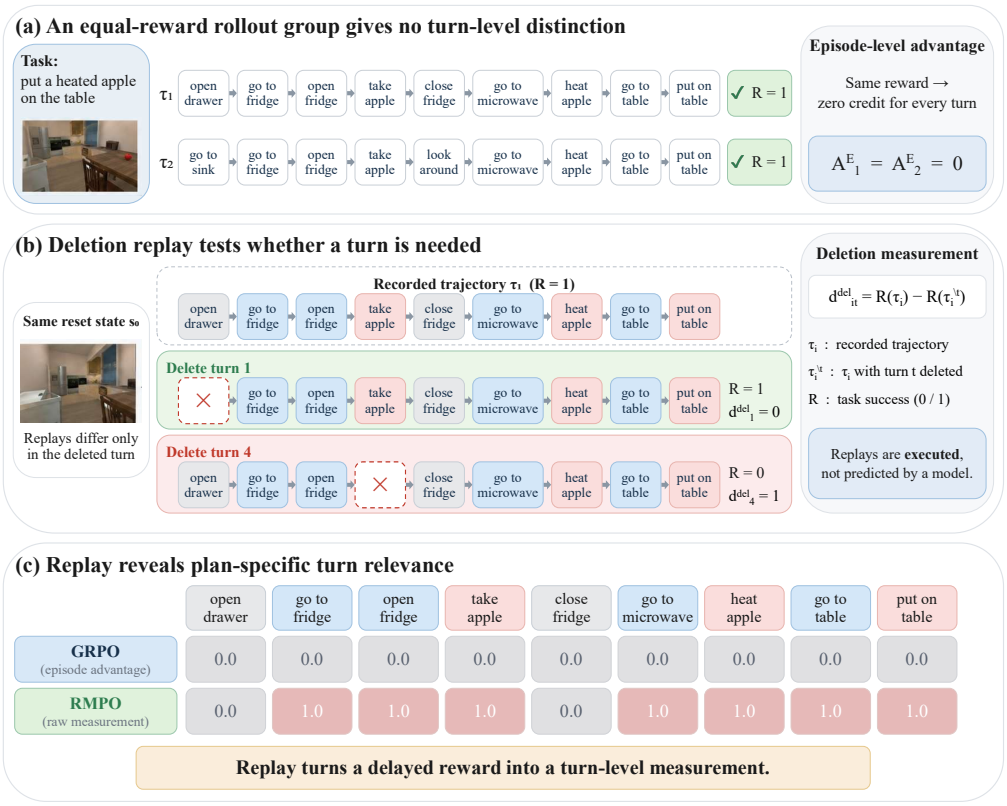


Figure 1: **Replay provides turn-level evidence from executed alternatives.** Two ALFWorld trajectories may receive the same episode reward even though their actions contribute differently to the outcome. By removing one action and replaying the recorded suffix under the same environment conditions, we can test whether the recorded plan still succeeds.

90.5% success on ALFWorld, a 91.5 WebShop score, and 47.9 macro EM on Search-QA. RMPO also achieves the highest mean score against the stronger turn-level credit baselines across ALFWorld, WebShop, and Search-QA. Additional analyses examine whether the replay measurements identify meaningful turns and whether the gains remain under matched computational budgets.

Our main contributions are:

- We introduce RMPO, a critic-free approach that obtains turn-level training signals by replaying controlled alternatives in the environment.
- We characterize what each replay measurement estimates: under a fixed evaluator, progress and replacement act as action scores relative to a history-dependent reference with a fixed-completion policy-improvement interpretation, whereas success-conditioned deletion is a hindsight contrast whose group normalization also induces a trajectory-level length effect (Appendix A).
- We demonstrate consistent improvements across three environments and two model scales, supported by ablations, a symbolic-planner audit, and matched-budget comparisons.

2 Related Work

Credit assignment in reinforcement learning. Temporal credit assignment under delayed rewards is a long-standing problem in reinforcement learning. Difference rewards measure individual contributions by comparing the observed return with a baseline in which that contribution is removed (Wolpert & Tumer, 2001; 1999), and COMA uses related counterfactual baselines for multi-agent policy gradients (Foerster et al., 2018). RMPO's action deletion is an executable, trajectory-level instance of this difference-reward or leave-one-out idea; the deletion primitive itself is not new. Other

approaches assign credit using future outcomes (Harutyunyan et al., 2019), future-conditioned critics (Mesnard et al., 2021), or return decomposition (Arjona-Medina et al., 2019). Likewise, a raw prefix-score difference resembles reward shaping with an environment-provided potential (Ng et al., 1999). RMPO uses these familiar forms as outputs of one replay interface, then studies how they can be constructed and combined for LLM-agent training. Counterfactually-Guided Policy Search reasons about alternative experience through a learned structural causal model (Buesing et al., 2019). RMPO focuses on settings where the changed interaction can be executed and scored by the training environment rather than predicted by a learned model.

Turn-level credit for LLM agents. Recent work has developed finer credit signals for multi-turn LLM agents. GiGPO, GAGPO, ECPO, and PGPO derive local supervision from observed rollouts or grouped states (Feng et al., 2025; Zhu et al., 2026; Li et al., 2026; Zheng et al., 2026). Graph-based approaches such as MileGPO use relationships among sampled trajectories to estimate progress (Qian et al., 2026). Other methods branch from intermediate states: VinePPO samples Monte Carlo policy continuations (Kazemnejad et al., 2025), and TCPO compares alternatives under a fixed interaction history (Liao et al., 2026). The Search-QA replacement construction is particularly close to GiGPO's anchor-state grouping: sibling queries are available only when rollouts share the same pre-action evidence state. RMPO treats that group as a candidate set for an executable comparison and scores the evidence returned by each query. More generally, its contribution is a unified environment-facing interface for such measurements, not a claim that every candidate construction is distinct from prior grouping methods.

Agent optimization and search. Our work is also related to studies of multi-turn training dynamics (Wang et al., 2025) and improvements to group-based policy optimization (Yu et al., 2025; Liu et al., 2025). For retrieval-augmented reasoning, Search-R1 trains agents to interleave search and reasoning, while StepSearch introduces step-wise supervision for search (Jin et al., 2025; Zheng et al., 2025). We use search-augmented QA as a setting where retrieval actions can be replayed and compared, while the final answer does not have a comparable replay alternative.

3 Problem Setup

Multi-turn agent training. We consider a multi-turn agent with policy π_θ. At turn t, the policy selects an action $a_t \sim \pi_\theta(\cdot \mid h_t)$ from the interaction history h_t. An interaction produces a trajectory $\tau = (s_0, a_1, s_1, \ldots, a_T, s_T)$, where s_t denotes the environment state after turn t. The trajectory receives an episode reward $R(\tau)$.

For each task instance, the policy samples a group of G trajectories $\{\tau_i\}_{i=1}^{G}$. Group Relative Policy Optimization (GRPO) computes an episode-level advantage for trajectory i via group normalization:

$$A_i^E = \frac{R(\tau_i) - \bar{R}}{\sigma_R + \epsilon_R},$$

where $\bar{R}$ and σ_R are the mean and population standard deviation of rewards within the group, and $\epsilon_R > 0$ is a small constant for numerical stability (Shao et al., 2024). The same advantage A_i^E is applied to all generated action tokens in trajectory τ_i. Thus, A_i^E captures how the trajectory performs relative to others in the group, but does not indicate how credit should be assigned to individual actions. RMPO retains this episode-level signal and supplements it with turn-level information obtained through replay.

Environment replay. RMPO assumes that, during training, a recorded interaction can be restored and re-executed under the same task conditions. Replay constructs matched comparisons around individual turns and is used only for training-signal construction; inference requires no restoration or re-execution.

4 Replay-Measured Policy Optimization

Section 3 shows that the episode-level advantage A_i^E captures how a trajectory performs relative to others in the group, but assigns the same signal to all turns in that trajectory. This is sufficient to tell whether a trajectory is relatively good or bad, but not which decisions were more useful within it.

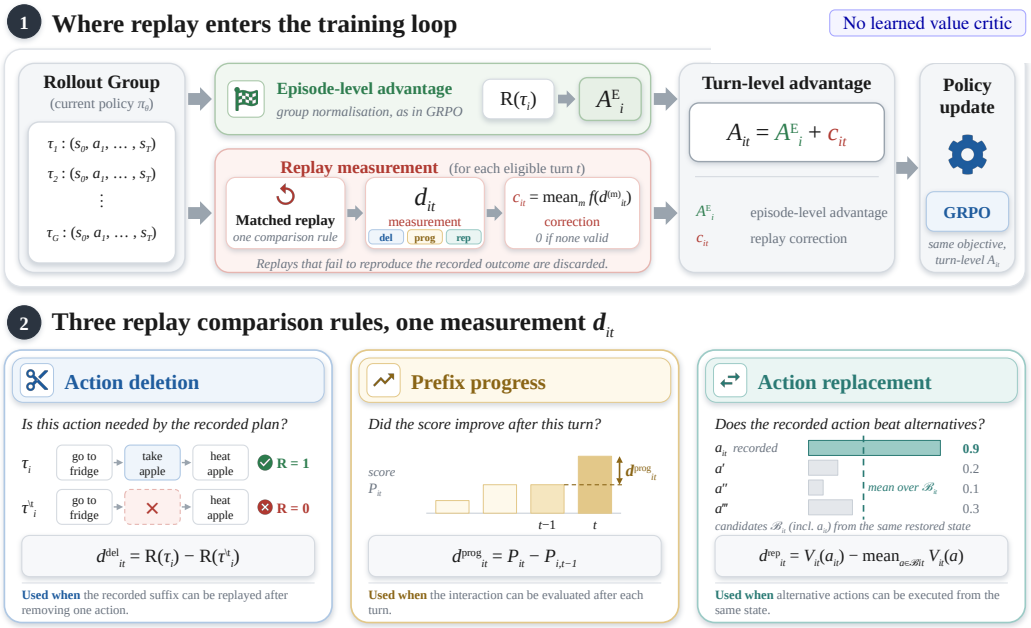


Figure 2: **Overview of RMPO.** Replay measurements refine the episode-level advantage into turn-level advantages for policy optimization.

RMPO adds this missing turn-level information through environment replay. The key idea is simple: when the training environment allows a turn to be re-executed or evaluated through a matched comparison, the resulting score difference provides additional evidence about that turn. For each eligible turn, RMPO constructs such a replay comparison and records the resulting measurement d_{it}. It then converts this measurement into a replay correction c_{it} and combines it with the episode-level advantage to obtain the turn-level advantage A_{it}. Figure 2 summarizes the procedure.

4.1 Replay Comparison Rules

A replay comparison executes the recorded interaction and a matched comparison that differs from it only around turn t, scores both with the same fixed procedure, and records the difference as the replay measurement d_{it}. We use three comparison rules. They share this interface and differ only in the matched comparison: the same execution without the action (deletion), the same trajectory one turn earlier (progress), or the same pre-action state with a different action (replacement).

Scores. All scores come from the environment or from a fixed evaluator that is not trained with the policy; none requires additional policy samples. Let x_{it} denote the interaction of τ_i up to and including turn t, with x_{i0} the initial task state. The episode reward $R(\tau)$ scores a complete execution. A *stopping score* $S(x)$ scores an unfinished interaction by the outcome of ending it at x under a fixed rule, for example answering from the evidence gathered so far or purchasing the best item selected so far. When the effect of an action becomes scorable only after further steps, a fixed deterministic completion C performs those steps before S is applied; otherwise C is the identity.

Action deletion. Let $\tau_i^{\setminus t}$ denote trajectory τ_i with action a_{it} removed and the remaining actions replayed in their original order, with invalid actions handled by the environment's usual rules. The replay measurement is

$$d^{\text{del}}_{it} = R(\tau_i) - R(\tau_i^{\setminus t}). \tag{1}$$

A positive value means that removing a_{it} lowers the episode reward. Deletion therefore measures the action's hindsight necessity for the recorded execution (Figure 1).

Prefix progress. Let $P_{it} = S(x_{it})$ be the stopping score after turn t. The replay measurement is

$$d^{\text{prog}}_{it} = P_{it} - P_{i,t-1}, \tag{2}$$

the change in what could be achieved by stopping, caused by turn t.

Action replacement. Let $x_{i,t-1} \oplus a$ denote the interaction obtained by restoring $x_{i,t-1}$ and executing action a, and let $\mathcal{B}_{it}$ be a candidate set that contains the recorded action a_{it}. Each candidate is scored by $V_{it}(a) = S\big(C(x_{i,t-1} \oplus a)\big)$, and the replay measurement is

$$d_{it}^{\text{rep}} = V_{it}(a_{it}) - \frac{1}{|\mathcal{B}_{it}|} \sum_{a \in \mathcal{B}_{it}} V_{it}(a). \tag{3}$$

A positive value means that the recorded action scores above the candidate average. Because progress and replacement share S, the recorded action receives $V_{it}(a_{it}) = S(C(x_{it}))$, which equals P_{it} whenever C is the identity. The two rules still answer different questions: progress asks what the turn changed along the recorded trajectory, whereas replacement asks how the chosen action compares with alternatives from the same state.

Example. Consider an instruction to buy a black T-shirt with women's fit in size XX-large, where stopping means buying the best product–option combination selected so far (detailed in Figure 8). Before any product is opened, $P_{it} = 0$. Opening a matching product raises P_{it} to 0.73, because the product matches but its options are not yet selected; each matching option then adds 0.09 until $P_{it} = 1.00$. Turns that change nothing, such as reading the product description, receive $d_{it}^{\text{prog}} = 0$. For replacement at the product-opening turn, the results page is restored, and each of the five listed products is opened and completed by C, which selects its options by a fixed rule. The recorded product scores $V_{it}(a_{it}) = 1.00$ against a candidate mean of 0.56, so $d_{it}^{\text{rep}} = +0.44$. Here C is needed because, without it, every candidate would be scored before its options are chosen; $V_{it}(a_{it}) = 1.00$ while $P_{it} = 0.73$. For an action whose effect can be scored at once, such as a retrieval query whose passages can be read immediately, C is the identity and the two scores coincide.

Choosing a replay comparison. Deletion requires that the recorded suffix remain executable after one action is removed; it is not used when the removal invalidates the suffix. Progress requires a stopping score, and replacement additionally requires executing alternatives from the restored pre-action state. Section 5.1 lists the scores, completions, and candidate sets that each environment provides, with further details in Appendix B.2.

4.2 From Replay Measurements to Turn-Level Credit

The measurements above are score differences obtained from replay. RMPO processes each available measurement before using it for credit assignment. For comparison rule m, we write

$$c_{it}^{(m)} = f(d_{it}^{(m)}). \tag{4}$$

Because different rules produce scores on different scales, we normalize each rule separately:

$$f(d_{it}^{(m)}) = \frac{d_{it}^{(m)} - \mu_{\mathcal{G}}^{(m)}}{\sigma_{\mathcal{G}}^{(m)} + \epsilon}, \tag{5}$$

where $\mu_{\mathcal{G}}^{(m)}$ and $\sigma_{\mathcal{G}}^{(m)}$ are the mean and population standard deviation over all valid measurements produced by rule m in the current rollout group $\mathcal{G}$, and $\epsilon > 0$ is a small constant for numerical stability. Computing both statistics at the group level preserves informative measurements in trajectories that contain only one eligible turn; such a measurement is compared with the other valid measurements in the group rather than being centered against itself.

If multiple replay comparisons are available for the same turn, we average their normalized measurements. Let $\mathcal{M}_{it}$ denote the set of valid comparisons for turn t. We define

$$c_{it} = \frac{1}{|\mathcal{M}_{it}|} \sum_{m \in \mathcal{M}_{it}} f(d_{it}^{(m)}), \tag{6}$$

with $c_{it} = 0$ when no valid comparison is available. Averaging rather than summing keeps turns with more valid rules from receiving systematically larger corrections; Table 2 ablates the per-rule normalization.

The final turn-level advantage is

$$\boxed{A_{it} = A_i^E + c_{it}.} \tag{7}$$

The two terms capture complementary information: A_i^E carries the trajectory-level learning signal, while c_{it} adds evidence obtained from replay about the current turn. This turn-level advantage is then used in the same GRPO policy objective. Algorithm 1 summarizes one RMPO update.

Interpretation. Replay measurements need not equal the on-policy action advantage. Because progress and replacement score actions with a fixed S and C, they can be viewed as action scores relative to a common reference. Deletion is different because it reuses the realized suffix and therefore measures hindsight necessity for the recorded execution. Appendix A formalizes these cases.

Algorithm 1 Replay-Measured Policy Optimization (one update)

Require: Policy π_θ, group size G, replay rules $\mathcal{M}$, normalization constants ϵ_R, ϵ
Ensure: Updated policy parameters θ

Sample a group $\mathcal{G} = \{\tau_i\}_{i=1}^G$ from π_θ and obtain rewards $R_i = R(\tau_i)$
Compute $A_i^E = (R_i - \bar{R})/(\sigma_R + \epsilon_R)$ for every $\tau_i \in \mathcal{G}$
for each replay rule $m \in \mathcal{M}$ **do**
    $D^{(m)} \leftarrow \varnothing$
    for each turn (i, t) eligible for rule m **do**
        if the recorded execution and matched comparison can be replayed and validated **then**
            $d_{it}^{(m)} \leftarrow \mathrm{Replay}_m(\tau_i, t)$; $D^{(m)} \leftarrow D^{(m)} \cup \{d_{it}^{(m)}\}$
        end if
    end for
    if $D^{(m)} \neq \varnothing$ **then**
        Compute $\mu_{\mathcal{G}}^{(m)}$ and $\sigma_{\mathcal{G}}^{(m)}$ over $D^{(m)}$
        for each valid $d_{it}^{(m)} \in D^{(m)}$ **do**
            $c_{it}^{(m)} \leftarrow (d_{it}^{(m)} - \mu_{\mathcal{G}}^{(m)})/(\sigma_{\mathcal{G}}^{(m)} + \epsilon)$
        end for
    end if
end for
for each recorded turn (i, t) **do**
    $\mathcal{M}_{it} \leftarrow \{m \in \mathcal{M} : d_{it}^{(m)}$ is valid$\}$
    $c_{it} \leftarrow 0$ if $\mathcal{M}_{it} = \varnothing$; otherwise $c_{it} \leftarrow |\mathcal{M}_{it}|^{-1} \sum_{m \in \mathcal{M}_{it}} c_{it}^{(m)}$
    $A_{it} \leftarrow A_i^E + c_{it}$
end for
Update θ with the GRPO policy objective using A_{it}

4.3 Replay Availability and Validity

A replay measurement is used only when the corresponding comparison can be restored and executed reliably under matched conditions. We first re-execute the recorded interaction and verify that it reproduces the recorded outcome; failed checks are discarded. If no valid comparison is available, we set $c_{it} = 0$, so the turn retains its episode-level advantage. Empirical coverage is reported in Section 5.4, with execution details in Appendix B.3.

5 Experiments

We evaluate whether replay-measured turn-level credit improves multi-turn policy optimization across different interaction settings. Our experiments examine four questions: whether RMPO improves task performance over episode-level and observed-rollout credit methods, whether the gains persist across model scales and stronger baselines, which replay measurements and aggregation choices are necessary, and whether the resulting measurements align with independent notions of turn progress at reasonable cost.

5.1 Experimental Setup

Environments and metrics. We evaluate on three multi-turn settings with different interaction structures. ALFWorld evaluates household planning on 134 `valid_unseen` tasks (Shridhar et al., 2021). We use a 50-turn cap, temperature 0.4, and three stochastic evaluation passes per training seed.

WebShop evaluates search, navigation, and purchase decisions on the official 500-task test set with a 15-turn cap (Yao et al., 2022). Each seed is evaluated twice, and the two passes are averaged before aggregation. We report both benchmark score and success rate.

Search-QA combines Natural Questions, TriviaQA, PopQA, HotpotQA, 2WikiMultiHopQA, MuSiQue, and Bamboogle with a frozen retrieval index and a four-turn cap (Jin et al., 2025). The final-answer reward is exact match (EM), and we report macro EM across the seven datasets.

RMPO instantiations. Each environment supplies the scores and candidate sets of Section 4.1; all settings use the same normalization and averaging rule (Section 4.2), with no environment-specific correction weights or replay schedules. ALFWorld uses action deletion, replaying the recorded suffix after removing one action and scoring it with the binary task reward. WebShop uses prefix progress and action replacement. Its stopping score $S(x)$ is the score of purchasing the best product–option combination selected in x, or zero if no purchase is possible. Replacement compares the first five executable products on the restored results page, together with the recorded product if it falls outside them; C opens each product and selects its options by matching the instruction's requested attributes, without model calls. Search-QA uses prefix progress and replacement among retrieval queries. Its stopping score $S(x)$ is the token-level F1 of a fixed reader's answer given the evidence retrieved in x, so it differs from the EM episode reward, and C is the identity. Candidate queries are the distinct queries issued by already collected sibling rollouts from the same pre-action evidence state; at the first retrieval turn, the original question is added as a fixed candidate. If deduplication leaves only the recorded query, the replacement measurement is zero. Appendix B.2 gives further construction details.

Models and training protocol. We train Qwen3-4B and Qwen3-8B policies with LoRA (Yang et al., 2025; Hu et al., 2021). Within each environment, GRPO, GiGPO, and RMPO use the same rollout budget, optimizer, adaptation setup, and checkpoint-selection protocol. All non-base results aggregate six independently trained seeds and report mean $\pm$ sample standard deviation. Multiple evaluation passes are averaged within each seed and are not treated as independent runs. We focus the analysis on Qwen3-8B and use Qwen3-4B to test whether the same pattern holds at a smaller scale. Appendix B provides the full training protocol.

Baselines. GRPO assigns the episode-level group advantage to every turn. GiGPO estimates turn-level credit from observed grouped rollouts (Feng et al., 2025). For both model scales, we additionally compare with GAGPO, ECPO, and PGPO, recent critic-free methods that derive finer credit from observed rollouts (Zhu et al., 2026; Li et al., 2026; Zheng et al., 2026). All baseline results are reproduced under our controlled training protocol rather than taken from the original papers.

5.2 Main Results

Table 1 reports the main controlled comparison. Under the unified configuration, RMPO outperforms both GRPO and GiGPO in every environment and at both model scales. Relative to GiGPO at 8B, RMPO improves ALFWorld success by 5.3 points, WebShop score by 4.4 points and success rate by 5.3 points, and Search-QA macro EM by 2.3 points. Against the additional baselines at 4B, RMPO exceeds the strongest result by 1.0 point on ALFWorld success, 1.5 on both WebShop metrics, and 1.2 on Search-QA macro EM. At 8B, the corresponding margins are 2.2, 1.8, 2.3, and 1.2 points. The base policy is included where available as a single-seed reference and is not used in uncertainty comparisons between trained methods.

Table 1: Controlled main results. Entries are training-seed mean ± sample standard deviation (%, $n = 6$), except single-seed base results marked *. ALFWorld reports success, WebShop reports score and success rate, and Search-QA reports macro EM. GAGPO, ECPO, and PGPO are evaluated at both model scales in all three settings; dashes denote unavailable base results.

Method	ALFWorld		WebShop				Search-QA	
	4B Success	8B Success	4B Score	4B Success Rate	8B Score	8B Success Rate	4B Macro EM	8B Macro EM
Base	61.9*	73.4*	–	–	–	–	20.2*	26.3*
GRPO	68.2 ± 3.8	77.8 ± 3.0	80.5 ± 2.8	67.2 ± 3.9	84.0 ± 2.7	71.5 ± 3.2	37.8 ± 1.7	41.3 ± 1.6
GiGPO	76.4 ± 3.1	85.2 ± 2.7	83.8 ± 2.7	71.2 ± 3.4	87.1 ± 2.4	75.0 ± 2.8	41.5 ± 1.6	45.6 ± 1.4
GAGPO	78.8 ± 2.9	88.3 ± 2.2	85.3 ± 2.5	72.8 ± 3.1	89.5 ± 2.1	77.5 ± 2.5	42.3 ± 1.5	46.4 ± 1.4
ECPO	78.1 ± 3.0	87.2 ± 2.4	84.9 ± 2.6	72.4 ± 3.1	88.8 ± 2.2	77.1 ± 2.6	42.0 ± 1.6	46.1 ± 1.4
PGPO	77.9 ± 3.0	86.8 ± 2.5	85.5 ± 2.4	73.2 ± 3.0	89.7 ± 2.0	78.0 ± 2.4	42.6 ± 1.5	46.7 ± 1.3
RMPO	**79.8 ± 2.7**	**90.5 ± 1.9**	**87.0 ± 2.2**	**74.7 ± 2.8**	**91.5 ± 1.8**	**80.3 ± 2.2**	**43.8 ± 1.4**	**47.9 ± 1.3**

5.3 Ablation Study

Table 2 isolates the Qwen3-8B components under the same training protocol. GRPO + progress adds the normalized environment-side prefix difference to the GRPO episode advantage, without same-state replacement; GRPO + replacement analogously uses replacement alone. We also add the same normalized progress measurement to GiGPO, without replacement. This separates access to environment-side progress from the additional information supplied by same-state replacement. The ALFWorld trajectory-mean control preserves each trajectory's aggregate correction under the policy-loss weighting while removing turn-specific variation; Appendix A.4 gives its definition.

The shuffled variants are retained only as perturbation controls: moving corrections among turns within each trajectory also introduces turn-level noise, so their gap from RMPO does not cleanly isolate the trajectory-level aggregate effect.

Table 2: Ablations of the unified Qwen3-8B method. Entries are mean ± sample standard deviation across six training seeds (%). Each setting reports its primary metric. Shuffled controls permute replay corrections only among turns within the same trajectory.

Setting	Variant	Result
ALFWorld	GRPO	77.8 ± 3.0
	Trajectory-mean correction	82.6 ± 2.8
	Within-trajectory shuffled replay corrections	79.1 ± 3.0
	RMPO	**90.5 ± 1.9**
WebShop	GRPO	84.0 ± 2.7
	GRPO + progress	88.5 ± 2.3
	GiGPO + progress	89.8 ± 2.1
	GRPO + replacement	89.2 ± 2.1
	RMPO: progress + replacement	**91.5 ± 1.8**
	Without per-rule normalization	89.8 ± 2.0
	Within-trajectory shuffled replay corrections	84.7 ± 2.7
Search-QA	GRPO	41.3 ± 1.6
	GRPO + progress	44.7 ± 1.5
	GiGPO + progress	46.6 ± 1.4
	GRPO + replacement	46.1 ± 1.4
	RMPO: progress + replacement	**47.9 ± 1.3**
	Without per-rule normalization	46.5 ± 1.4
	Within-trajectory shuffled replay corrections	41.8 ± 1.6

RMPO improves ALFWorld success over GRPO by 12.7 points, while the trajectory-mean control recovers 4.8 points, or approximately 38% of that gain. Thus, the trajectory-level length effect predicted in Appendix A.4 contributes meaningfully but does not explain most of RMPO's improvement; the within-trajectory placement of deletion corrections carries additional information. The trajectory-mean control also retains nonzero replay updates in reward-homogeneous groups, so it simultaneously controls for that mechanism. Adding progress to GiGPO also improves its WebShop score and Search-QA macro EM, while the combined progress and replacement method remains strongest.

5.4 Replay Availability and Informativeness

Table 3 summarizes where each replay comparison is available. Deletion covers approximately 96% of eligible turns on successful ALFWorld trajectories. In Search-QA, only 26% of measured retrieval turns have non-zero prefix progress, making same-state replacement particularly useful when progress is uninformative. Turns without a valid comparison retain their episode-level advantage.

Table 3: **Replay coverage and boundary cases.** Turns outside the stated support receive $c_{it} = 0$.

Environment	Measurements	Support and coverage
ALFWorld	Deletion	$\sim$96% of eligible turns on successful trajectories; failed trajectories and invalid replays are unmeasured.
WebShop	Progress; product replacement	Navigation and product turns; termination and malformed turns are unmeasured.
Search-QA	Progress; query replacement	Retrieval turns; 26% of progress measurements are non-zero; final-answer turns are unmeasured.

Appendix A discusses the resulting hindsight and coverage properties in more detail.

5.5 Measurement Audit and Replay Cost

Across the signal-specific valid trajectory sets, the mean within-trajectory Spearman correlations are 0.842 for deletion, 0.108 for GiGPO, 0.417 for turn position, and 0.006 after shuffling. Appendix C gives the audit protocol, Table 6 reports the signal-specific valid trajectory counts, and Figure 3 shows the turn-level patterns; both deletion and planner progress use hindsight information unavailable to the acting policy. Under matched budgets in ALFWorld, RMPO reaches 90.5% success versus 87.1% for four-continuation sampling at equal root rollouts and 85.5% at equal GPU-hours (Table 4).

Replay execution cost. Equivalent replay jobs are deduplicated, repeated evaluator scores are memoized, and the remaining jobs are executed in batches. With these optimizations, replay measurement accounts for 92/2758 s (3.3%) of recorded iteration time in ALFWorld, 254/770 s (33.0%) in WebShop, and 10/575 s (1.7%) in Search-QA. Search-QA additionally uses 179 short fixed-reader completions; ALFWorld and WebShop require no additional policy calls.

6 Conclusion and Limitations

The main empirical finding is that executable replay can provide useful credit beyond rollout-derived estimates or dense progress shaping. Across three environments and two model scales, RMPO improves group-based policy optimization; in ALFWorld, deletion aligns with symbolic-planner progress and outperforms sampled continuations under matched budgets. The replay interface is the mechanism that makes these measurements usable during agent training.

Limitations. The interface requires resettable task conditions and reproducible comparisons. It suits simulators, frozen retrieval systems, and offline replicas, but not live websites or physical environments that cannot be reset. Stochastic settings require controlled random seeds or repeated replays; otherwise transition noise is confounded with the changed action.

Replay measurements also encode hindsight and environment semantics. Deletion measures necessity for one realized suffix, not ex-ante decision quality, so it can downweight reasonable exploration, miss redundant action sets, and learn permissive simulator preconditions. Normalized c_{it} is not generally potential-based; RMPO is therefore a biased shaping objective and need not preserve the original reward's optimal policy.

Coverage and evaluator quality vary: failed ALFWorld trajectories receive no deletion credit; WebShop adds substantial CPU cost and lacks a compute-matched continuation comparison; Search-QA inherits reader errors and leaves answer turns unmeasured. Signed failures, stochastic comparisons, and interacting deletions could broaden coverage at additional cost.

ETHICS STATEMENT

RMPO is evaluated only in established simulated benchmarks. It does not collect personal data or interact with live users or external services. The method can make agent training more effective, so deployments should still be reviewed for task-specific safety, privacy, and misuse risks.

REPRODUCIBILITY STATEMENT

We provide anonymous code at `https://anonymous.4open.science/r/rmpo-CD48`. The paper specifies the replay comparison rules, training protocol, hyperparameters, evaluation settings, six-seed aggregation, and measurement-cost profile needed to reproduce the reported comparisons.

AI USE STATEMENT

AI assistance was used to organize the manuscript, develop its exposition, and edit the LaTeX draft.

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

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/16105fb9cc614fc29e1bda00dab60d41-Abstract.html.

Lars Buesing, Theophane Weber, Yori Zwols, Sebastien Racaniere, Arthur Guez, Jean-Baptiste Lespiau, and Nicolas Heess. Woulda, coulda, shoulda: Counterfactually-guided policy search. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJG0voC9YQ.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/420c9f777c0b4f78d515e53cf74d58b2-Abstract-Conference.html.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11794.

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

Malte Helmert. The Fast Downward planning system. *Journal of Artificial Intelligence Research*, 26:191–246, 2006. doi: 10.1613/jair.1705.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. arXiv:2106.09685, 2021. URL https://arxiv.org/abs/2106.09685.

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. arXiv:2503.09516, 2025. URL https://arxiv.org/abs/2503.09516.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590, 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Yuanfan Li, Qi Zhou, Wenjing Duan, and Lu Chen. When denser credit is not enough: Evidence-calibrated policy optimization for long-horizon LLM agent training. arXiv:2606.05885, 2026. URL https://arxiv.org/abs/2606.05885v1.

Sicong Liao, Zhi Chen, and Yaohua Tang. TCPO: Turn-level credit policy optimization. arXiv:2608.01667, 2026. URL https://arxiv.org/abs/2608.01667v1.

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. arXiv:2503.20783, 2025. URL https://arxiv.org/abs/2503.20783.

Thomas Mesnard, Théophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S. Stepleton, Nicolas Heess, Arthur Guez, Éric Moulines, Marcus Hutter, Lars Buesing, and Rémi Munos. Counterfactual credit assignment in model-free reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7654–7664, 2021. URL https://proceedings.mlr.press/v139/mesnard21a.html.

Andrew Y. Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 278–287, 1999. URL https://ai.stanford.edu/~ang/papers/shaping-icml99.pdf.

Bo Qian, Yuting Wu, Shuang Zeng, Huaiyu Wan, Dalin Zhang, and Jiqiang Liu. MileGPO: Milestone inference with local evidence for graph-based policy optimization of long-horizon LLM agents. arXiv:2608.19803, 2026. URL https://arxiv.org/abs/2608.19803v1.

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: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://arxiv.org/abs/2010.03768.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, Eli Gottlieb, Monica Lam, 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:2504.20073, 2025. URL https://arxiv.org/abs/2504.20073.

David H. Wolpert and Kagan Tumer. An introduction to collective intelligence. arXiv:cs/9908014, 1999. URL https://arxiv.org/abs/cs/9908014.

David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(02n03):265–279, 2001.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. arXiv:2505.09388, 2025. URL https://arxiv.org/abs/2505.09388.

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. URL https://papers.nips.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, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru 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, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38,

2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Xuhui Zheng, Kang An, Ziliang Wang, Yuhang Wang, and Yichao Wu. StepSearch: Igniting LLMs search ability via step-wise proximal policy optimization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL https://aclanthology.org/2025.emnlp-main.1106/.

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

Siyuan Zhu, Chao Yu, Rongxin Yang, Zongkai Liu, Jinjun Hu, Qiwen Chen, and Yibo Zhang. GAGPO: Generalized advantage grouped policy optimization. arXiv:2605.13217, 2026. URL https://arxiv.org/abs/2605.13217v1.

A Mathematical Details

A.1 Relation to GRPO

When $c_{it} = 0$, Equation (7) reduces exactly to GRPO. Otherwise, RMPO retains the same episode-level advantage and adds the average normalized correction from the valid replay comparisons at that turn.

A.2 Fixed-Evaluator Replay Comparisons

Let C denote a fixed completion or evaluation procedure. For a history h and action a, define

$$Q^C(h, a) = \mathbb{E}[\text{Score}(C(h, a))]. \tag{8}$$

For any reference $b_C(h)$ that depends on the pre-action history but not on a, define

$$\delta_C(h, a) = Q^C(h, a) - b_C(h). \tag{9}$$

Proposition 1. The reference term does not affect the expected score-function update:

$$\mathbb{E}_{a\sim\pi_\theta(\cdot|h)}\left[\nabla_\theta \log \pi_\theta(a \mid h)\delta_C(h, a)\right] = \mathbb{E}_{a\sim\pi_\theta(\cdot|h)}\left[\nabla_\theta \log \pi_\theta(a \mid h)Q^C(h, a)\right]. \tag{10}$$

Proof. The result follows from $\mathbb{E}_{a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a \mid h)] = 0$.

Corollary 1 (fixed-completion policy improvement). Suppose C executes the selected action and then follows a fixed completion policy π_C under the same reward, so that $Q^C = Q^{\pi_C}$. The correction-only score-function semi-gradient therefore favors actions with larger $Q^{\pi_C}(h, a)$. The greedy policy

$$\pi'(h) \in \arg\max_a Q^{\pi_C}(h, a)$$

satisfies $J(\pi') \geq J(\pi_C)$ by the policy-improvement theorem. If π_C is optimal, the ranking induced by Q^C has the same maximizing actions as Q^*.

Scope. Prefix progress has this form when the before- and after-turn scores are produced by the same fixed evaluator. Same-state replacement has the same interpretation when its candidate reference is fixed by the pre-action state. In WebShop, the recorded product is added when it falls outside the first five candidates, so the reference is action dependent on those turns and Proposition 1 does not apply exactly. In Search-QA, the candidate set always includes the recorded query and is deduplicated, so its reference can likewise depend on the selected action and does not strictly satisfy the premise. Search-QA also uses reader F1 for replay measurement while the episode reward is EM, so the policy-improvement interpretation applies to the reader score rather than directly to EM. Finally, RMPO normalizes measurements using sampled rollout-group statistics; these statistics are detached during the policy update, making the analysis a semi-gradient interpretation rather than the gradient of a fixed global objective.

A.3 Success-Conditioned Action Deletion

ALFWorld deletion is evaluated only for successful recorded trajectories. Let

$$I(\tau) = \mathbf{1}\{R(\tau) = 1\},$$

and define the on-policy action value under the binary reward as the success probability

$$Q^\pi(h, a) = \Pr_\pi(R(\tau) = 1 \mid h, a) = \mathbb{E}_\pi[R(\tau) \mid h, a].$$

Let $\tau^{\setminus t}$ denote the execution obtained by removing the recorded action at turn t and replaying the recorded suffix. Define the zero-extended raw deletion measurement

$$\widetilde{d}^{\text{del}}(h, a) = I(\tau)\left[R(\tau) - R(\tau^{\setminus t})\right]. \tag{11}$$

Because $R(\tau) \in \{0, 1\}$ in ALFWorld,

$$\mathbb{E}[\widetilde{d}^{\mathrm{del}} \mid h, a] = Q^{\pi}(h, a) - B_{\mathrm{del}}(h, a), \tag{12}$$

where

$$B_{\mathrm{del}}(h, a) = \mathbb{E}\Big[I(\tau)R(\tau^{\setminus t}) \mid h, a\Big]. \tag{13}$$

To include both group normalization and the fact that failed trajectories receive no deletion correction, define the zero-extended normalized correction

$$\widetilde{c}^{\mathrm{del}}(h, a) = \frac{I(\tau)\left[R(\tau) - R(\tau^{\setminus t}) - \mu_{\mathcal{G}}^{\mathrm{del}}\right]}{\sigma_{\mathcal{G}}^{\mathrm{del}} + \epsilon}. \tag{14}$$

This definition applies only to valid deletion replays; invalid replays (approximately 4% of eligible turns) instead use $c_{it} = 0$.

Proposition 2. Treating the detached group statistics as fixed,

$$\mathbb{E}\big[\widetilde{c}^{\mathrm{del}}(h, a) \mid h, a\big] = \frac{\big(1 - \mu_{\mathcal{G}}^{\mathrm{del}}\big)\, Q^{\pi}(h, a) - B_{\mathrm{del}}(h, a)}{\sigma_{\mathcal{G}}^{\mathrm{del}} + \epsilon}. \tag{15}$$

Thus, if $B_{\mathrm{del}}(h, a)$ is independent of a for a fixed history h, the expected normalized deletion correction is a nonnegative rescaling of the on-policy action value plus a history-dependent baseline. In general, $B_{\mathrm{del}}(h, a)$ is action dependent because the recorded action affects which suffix is realized, so deletion measures necessity relative to that realized suffix rather than an unbiased on-policy advantage.

Proof. Because the reward is binary, $I(\tau)R(\tau) = R(\tau)$. Moreover, by the success-probability definition of Q^{π}, $\mathbb{E}[I(\tau) \mid h, a] = Q^{\pi}(h, a)$. Taking the conditional expectation of the definition above and applying this identity and the definition of B_{del} gives Equation (15).

Information-gathering actions. Consider an action that changes the agent's observation but not the latent task state. If the recorded suffix remains executable and successful after that action is removed, then $d^{\mathrm{del}} = 0$ even when the observation affected how the policy chose the suffix in the original interaction. This is the case for several search actions in Figure 7: deletion evaluates the realized plan after the search has already influenced later decisions.

A.4 Aggregate Effect of Normalized Deletion

For a successful ALFWorld trajectory i, let n_i be the number of valid deletion measurements and let K_i be the number with $d_{it}^{\mathrm{del}} = 1$. From Equation 5,

$$\sum_{t=1}^{n_i} c_{it} = \frac{K_i - \mu_{\mathcal{G}}^{\mathrm{del}} n_i}{\sigma_{\mathcal{G}}^{\mathrm{del}} + \epsilon}. \tag{16}$$

Thus, when successful trajectories contain similar numbers of necessary actions, their aggregate replay correction decreases as additional non-necessary turns are added.

In the training group shown in Figure 7, $\mu_{\mathcal{G}}^{\mathrm{del}} = 0.42$ and $\sigma_{\mathcal{G}}^{\mathrm{del}} = 0.49$. Both the 12-turn and 30-turn successful trajectories contain seven actions with $d^{\mathrm{del}} = 1$. Their unweighted aggregate corrections are approximately

$$7(1.18) - 5(0.85) = +4.0$$

and

$$7(1.18) - 23(0.85) = -11.3,$$

respectively.

Equation 16 describes the sum of turn corrections. Its effect on optimization depends on the reduction used in the policy loss. For a loss with normalized turn weights ρ_{it}, the corresponding quantity is $\sum_t \rho_{it} c_{it}$. To isolate turn-specific variation, let $\mathcal{T}_i$ be the valid replay turns and define

$$\bar{c}_i^{\rho} = \frac{\sum_{t \in \mathcal{T}_i} \rho_{it} c_{it}}{\sum_{t \in \mathcal{T}_i} \rho_{it}}, \qquad c_{it}^{\mathrm{mean}} = \begin{cases} \bar{c}_i^{\rho}, & t \in \mathcal{T}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

Then

$$\sum_t \rho_{it} c_{it}^{\text{mean}} = \sum_t \rho_{it} c_{it}. \tag{18}$$

The trajectory-mean control in Section 5.3 uses these same loss weights, preserving the trajectory-level aggregate correction while removing within-trajectory variation.

B Experimental Protocol

B.1 Training and Evaluation Settings

All controlled methods use the same optimization and adaptation settings within each environment. We use AdamW with a learning rate of 1×10^{-5}. Rollouts are sampled with temperature 1.0 using groups of eight trajectories in all three environments, with 16 groups collected per policy update. Evaluation uses temperature 0.4 for ALFWorld and WebShop and greedy decoding for Search-QA. We use KL coefficients of 0.01 for ALFWorld and WebShop and 0.001 for Search-QA.

All models are adapted with LoRA of rank 32 and scaling 64 on the attention and MLP projections. Training uses bf16 precision with gradient checkpointing. Within each environment, GRPO, GiGPO, and RMPO use the same rollout budget, optimizer, adaptation setup, and checkpoint-selection rule.

For the progress-matched GiGPO control, we use

$$A_{it}^{\text{GiGPO+prog}} = A_{it}^{\text{GiGPO}} + f(d_{it}^{\text{prog}}), \tag{19}$$

with exactly the same progress normalization as RMPO and no replacement signal.

B.2 Replay Instantiations

ALFWorld. We use action deletion on successful recorded trajectories. The recorded suffix is replayed after removing the selected action, and the episode reward provides the replay score. Failed trajectories receive no deletion comparison and therefore use $c_{it} = 0$ at every turn. This instantiation measures hindsight necessity within successful plans; it does not use deletion-to- success events from failed plans as signed negative credit.

WebShop. We use prefix progress and action replacement with the same stopping score: $S(x)$ is the score of purchasing the best product–option combination selected in x, and zero if no purchase is possible. Because earlier selections remain purchasable, $P_{it} = S(x_{it})$ is the maximum, over prefixes up to turn t, of the score obtained by purchasing immediately; leaving a product page or selecting a suboptimal option therefore does not decrease P_{it}. Action replacement applies the completion C before scoring: for each candidate a, C opens the product from the restored result page and selects its options by matching the instruction's requested attributes, without model calls, and $V_{it}(a)$ is the resulting purchase score. The candidate set contains the first five executable product choices on the restored result page, together with the recorded choice if it falls outside this set.

Search-QA. We use prefix progress for retrieval turns and replacement among alternative queries. The stopping score $S(x)$ is the token-level F1 of a fixed reader's answer given the evidence retrieved in x, so P_{i0} is the reader's closed-book score; this replay score differs from the EM episode reward. Because retrieved passages can be read at once, the completion C is the identity: $V_{it}(a) = S(x_{i,t-1} \oplus a)$ scores the evidence gathered before turn t together with the passages returned by query a, and hence $V_{it}(a_{it}) = P_{it}$. The replacement candidate set contains the recorded query and the distinct queries issued by already collected sibling rollouts that share the same pre-action evidence state. At the first retrieval turn, where all rollouts share the same evidence state, the original question is added as a fixed candidate. If deduplication leaves only the recorded query, $d_{it}^{\text{rep}} = 0$; this measurement is retained as valid and enters the group statistics in Eq. (5). No additional policy sampling is used to construct the candidate set.

B.3 Replay Execution and Cost

Restoration and validation. Each comparison restores the recorded task condition, environment version, evaluator, and pre-action state required by the replay rule. Stochastic components use matched random seeds; repeated execution would be required when randomness cannot otherwise be controlled. We first verify that re-executing the recorded interaction reproduces its outcome. Comparisons that fail this check are discarded. After the intervention, deletion additionally requires a replayable suffix and replacement requires executable candidates from the same restored state. The validation work is included in the timing measurements.

Deduplication and memoization. Equivalent replay comparisons are executed once. ALFWorld deletion replays are keyed by the resulting action sequence, Search-QA candidates by their retrieved evidence set, and fixed-reader or WebShop completion values by the task and restored interaction state. These optimizations are included in the timing measurements below.

Batched timing configuration. ALFWorld deletion replays use asynchronous batched environments in chunks of 96 slots. The timing profile uses 32 rollout workers and 64 annotation workers under the same server allocation as the reported runs.

Matched-budget continuation comparison. Table 4 compares deletion replay with policy-continuation sampling in ALFWorld. The equal-root-rollout setting keeps the collected root trajectories fixed; the equal-GPU-hours setting reduces root rollouts to match the continuation cost.

Table 4: ALFWorld Qwen3-8B comparison with continuation sampling. Entries are success mean $\pm$ sample standard deviation across six training seeds (%).

Method	Success	Extra policy continuations
GiGPO	85.2 ± 2.7	0
Continuation-MC ($k = 4$), equal root rollouts	87.1 ± 2.5	4 / eligible prefix
Continuation-MC ($k = 4$), equal GPU-hours	85.5 ± 2.8	4 / eligible prefix
RMPO	$\mathbf{90.5 \pm 1.9}$	0

Timing profile. Table 5 reports the per-iteration cost of the replay measurements used in the reported runs.

Table 5: Recorded per-iteration measurement profile. Timings use 32 rollout workers and 64 annotation workers with the replay execution optimizations above. Additional model calls exclude ordinary policy rollouts.

Environment	Measurement / total (s)	Fraction	Additional model calls
ALFWorld	92 / 2,758	3.3%	0
WebShop	254 / 770	33.0%	0
Search-QA	10 / 575	1.7%	179 reader completions

The accounting separates ordinary policy rollouts from the additional work required to construct replay measurements. The Search-QA reader calls score retrieved evidence and do not sample policy continuations. Table 4 separately reports the cost-controlled comparison with four trained-policy continuations per eligible prefix. Absolute timing depends on the execution hardware and worker configuration. The matched-compute continuation comparison is available only for ALFWorld; the reported WebShop results therefore do not establish an advantage under equal wall-clock or GPU-hour budgets.

C Symbolic-Planner Progress Audit

We use symbolic-planner progress to audit whether turn-level training signals agree with independent task progress in ALFWorld. For a simulator state s, Fast Downward returns the length of an

optimal remaining plan when the goal is solvable (Helmert, 2006). We define

$$\ell(s) = \begin{cases} 0, & \text{if } s \text{ satisfies the goal,} \\ \varnothing, & \text{if the planner finds no solution,} \\ \text{optimal remaining plan length,} & \text{otherwise.} \end{cases} \tag{20}$$

For a transition from s_t to s_{t+1}, planner progress is

$$p_t = \begin{cases} \ell(s_t) - \ell(s_{t+1}), & \ell(s_t), \ell(s_{t+1}) \neq \varnothing, \\ \text{undefined,} & \text{otherwise.} \end{cases} \tag{21}$$

Thus positive values move toward the goal, zero makes no optimal-plan progress, and negative values move away.

Unsolvable states. Undefined planner-progress values account for 6.8% of audited turns and are mapped to zero in the reported correlation analysis, treating transitions involving an unsolvable endpoint as no measured planner progress. Planner progress is used only for this audit.

Aggregation and uncertainty. For each signal, we compute Spearman rank correlation separately within each trajectory, using average ranks for ties. We omit trajectories with fewer than three turns, a constant planner-progress sequence, or a constant compared-signal sequence, because rank correlation is then undefined. We report the unweighted arithmetic mean over the remaining trajectories, so longer trajectories receive no extra weight. Table 6 reports the signal-specific valid trajectory counts. The audit pool contains 3,695 turns from 192 rollouts before this filtering. Because filtering depends on the compared signal, the reported means use different trajectory subsets and are not a matched-subset comparison.

Table 6: Symbolic-planner progress audit in ALFWorld. n is the number of valid trajectories after signal-specific filtering.

Signal	Spearman ρ	n
Deletion measurement	0.842	158
GiGPO step term	0.108	135
Position only	0.417	171
Shuffled deletion measurement	0.006	158

The deletion measurement remains strongly aligned with planner progress, whereas the shuffled control is near zero. Position alone retains moderate agreement. Planner progress uses the simulator's full symbolic state, and deletion evaluates a completed suffix; their agreement therefore partly reflects a shared hindsight perspective rather than information available to the policy at decision time.

Planner progress is computed from the simulator's symbolic state, whereas the deletion measurement uses terminal outcomes from replayed interactions. Figure 3 evaluates whether these independently obtained quantities rank turns similarly within a trajectory.

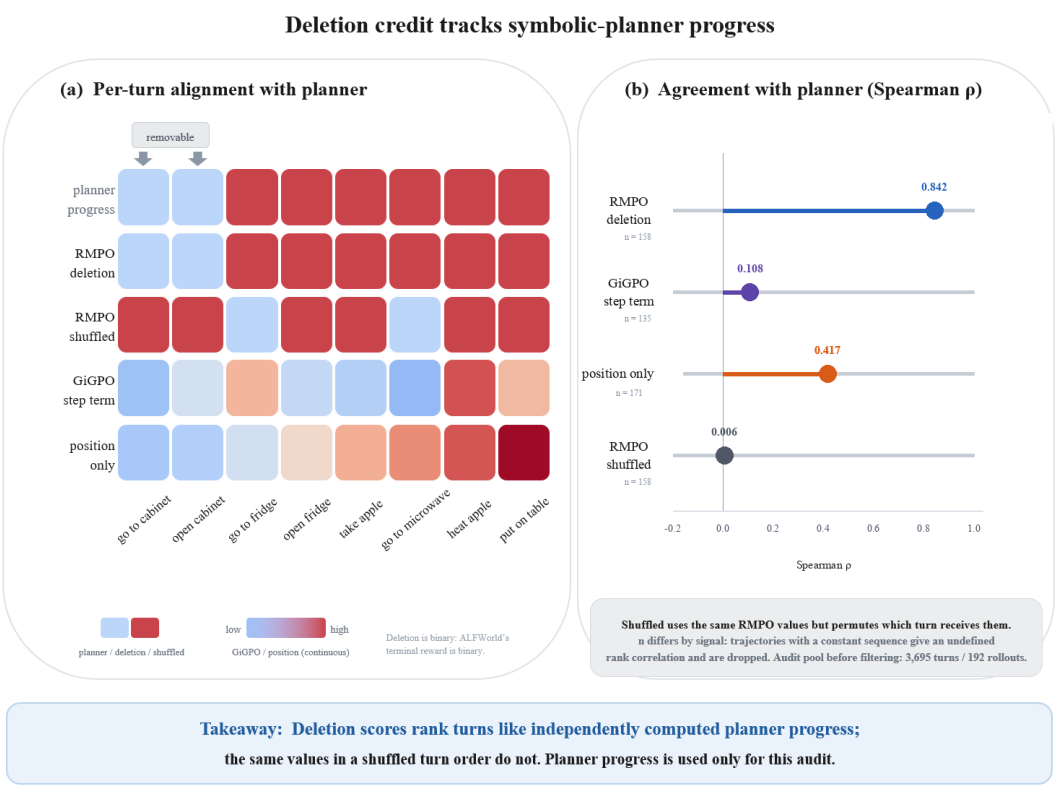


Figure 3: **The deletion measurement tracks symbolic-planner progress.** Left: per-turn values for an example trajectory; both quantities assign low values to removable turns. Right: mean within-trajectory Spearman correlation over the ALFWorld audit. Shuffling preserves the measured values but changes their turn assignments.

D PROMPTS

All three environments use a chat format: one system prompt, then one user message per turn carrying the current observation. The model must end each reply with a single `Action:` line. The prompts below are reproduced verbatim from the code; fields shown in `{...}` are filled in at run time. Figures 4, 5, and 6 give the prompts for ALFWorld, WebShop, and Search-QA.

ALFWorld

system prompt | user message per turn | thinking mode on

System prompt

```
You are an agent in a text-based household environment.
You are given a household task. You must explore the rooms, find the objects you
need, and manipulate them to complete the task.
Each turn, look at the observation and issue exactly ONE action, on its own line:
Action: <your action>
You will be shown the list of admissible actions each turn. Your action MUST be one of them, copied
exactly. Anything else is rejected and wastes the step.
Guidance:
- Receptacles must be opened before you can see inside them. 'go to <recep>'
  moves you there; if it is closed, 'open <recep>' before taking anything.
- You can carry only ONE object at a time. Take it, then go to the destination,
  then put it there.
- 'put <obj> in/on <recep>' requires you to be holding <obj> and to be at
  <recep>.
- Cleaning uses a sinkbasin, heating a microwave, cooling a fridge. The pattern
  is: hold the object, go to the appliance, then use the clean/heat/cool action.
- 'examine <obj>' and 'look' inspect without consuming the object.
- If you cannot find an object, systematically visit the receptacles that would
  plausibly hold it rather than repeating the same one.
RESPONSE FORMAT -- this is strict:
Your reply must END with a line of exactly the form
Action: <one action>
Nothing may follow that line, and it must be a command, not prose.
```

✓ *Correct:*

```
Action: go to cabinet 4
Action: take mug 1 from countertop 1
Action: clean mug 1 with sinkbasin 1
```

✗ *Wrong (these are spent as wasted steps):*

```
Alternatively I could check the drawer
I think the mug is in the cabinet
Action: let me look around first
```

User message, first turn

```
Task: {goal}
Observation:
{observation}
Admissible actions:
  {admissible action 1}
  {admissible action 2}
  ...
(step 0/{max_turns})
```

Later turns: the same message without the `Task:` *line, with the step counter advanced.*

Figure 4: **ALFWorld prompts.** System prompt and per-turn user message, reproduced verbatim from the code. Fields in **red** are filled in at run time.

WebShop — system prompt · user message per turn · thinking mode off

System prompt

```
You are shopping on a text-based web store.
You are given an instruction describing what to buy. Search for it, open a
product, choose the options the instruction asks for, and buy it.
Actions look like:
  search[<query>]        only on the search page
  click[<text>]          the exact text of a clickable item shown below
You will be shown the clickable items each turn. Your action MUST be one of the
listed forms, copied exactly.
Think briefly, then end your reply with exactly one line:
Action: <action>
Nothing may follow that line.
```

User message, every turn

```
Step {t}/{max_turns}
{page text}
You may search: search[<your query>]
Clickable: click[{item 1}], click[{item 2}], ...
Your action:
```

The `You may search` *line appears only on the search page.*

Figure 5: **WebShop prompts.** Fields in **red** are filled in at run time.

Search-QA — system prompt · user message per turn

System prompt

```
You answer questions by searching Wikipedia.
Each turn, reply with exactly one line, either
  Action: search[your query]
  Action: answer[your final answer]
Search returns the three most relevant passages. You have a limited number
of turns, so search only as many times as you need, then answer. Keep the
answer short: a name, a date, a number, or yes/no.
```

User message, first turn

```
Question: {question}
```

User message after a search

```
Results (turn {t} of {max_turns}):
[{title 1}] {passage 1}
[{title 2}] {passage 2}
[{title 3}] {passage 3}
```

Each passage is prefixed by its title and cut to at most 1200 characters.

Figure 6: **Search-QA prompts.** Fields in **red** are filled in at run time.

E Worked Examples of Replay Measurement and Credit

Figures 7–11 illustrate representative replay measurements and, for selected trajectories, the resulting turn-level advantages. The examples use groups of eight rollouts of one task, sampled from the initial Qwen3-8B policy. The measurements were recorded by the replay code, and the corrections c_{it} and turn-level advantages A_{it} are obtained with Eqs. (5)–(7). The GRPO advantage, which repeats the episode advantage on every turn, is shown for comparison.

ALFWorld · action deletion

`pick_heat_then_place_in_recep-Potato-None-Fridge-19`

Task: put a hot potato in fridge. **Group of 8 rollouts.** ✓12 ✓13 ✓13 ✓13 ✓17 ✓19 ✓30 ✗30 turns per rollout
Episode advantage: $+0.38$ for each success, -2.65 for the failure. Every successful rollout ends with the same seven needed actions, preceded by 4 to 23 turns of searching for the potato.

Successful rollout 1, turn by turn

bars: turn-level advantage; GRPO RMPO

Turn	Action	Observation after the action (abridged)	d^{del}	c_{it}	GRPO	RMPO
1	`go to countertop 1`	On the countertop 1, you see a butterknife 1, a cup 3, a fork 1, a knife 1, a pan *[...]*	0	-0.85	$+0.38$	-0.47
2	`go to countertop 2`	On the countertop 2, you see a egg 2, a fork 3, a fork 2, a plate 2, and a soapbottle *[...]*	0	-0.85	$+0.38$	-0.47
3	`go to drawer 1`	On the drawer 1, you see nothing.	0	-0.85	$+0.38$	-0.47
4	`go to drawer 2`	On the drawer 2, you see nothing.	0	-0.85	$+0.38$	-0.47
5	`go to diningtable 1`	On the diningtable 1, you see a apple 2, a apple 1, a bread 1, *[...]* a potato 1, *[...]*	$+1$	$+1.18$	$+0.38$	$+1.56$
6	`take potato 1 from diningtable 1`	You pick up the potato 1 from the diningtable 1.	$+1$	$+1.18$	$+0.38$	$+1.56$
7	`go to microwave 1`	The microwave 1 is closed.	$+1$	$+1.18$	$+0.38$	$+1.56$
8	`open microwave 1`	The microwave 1 is open. In it, you see nothing.	0	-0.85	$+0.38$	-0.47
9	`heat potato 1 with microwave 1`	You heat the potato 1 using the microwave 1.	$+1$	$+1.18$	$+0.38$	$+1.56$
10	`go to fridge 1`	The fridge 1 is closed.	$+1$	$+1.18$	$+0.38$	$+1.56$
11	`open fridge 1`	The fridge 1 is open. In it, you see a bowl 1, a lettuce 1, and a mug 1.	$+1$	$+1.18$	$+0.38$	$+1.56$
12	`move potato 1 to fridge 1`	You move the potato 1 to the fridge 1.	$+1$	$+1.18$	$+0.38$	$+1.56$

Agent · Turn 5 *where d^{del} first becomes 1*

<think>
Okay, so I need to put a hot potato in the fridge. Let me think about how to approach this. First, I need to find a potato. [...] The user has already checked countertops 1 and 2, and drawers 1 and 2, but didn't find any potatoes.
Next, I should consider other locations where a potato might be. Common places could be cabinets, the fridge itself (though maybe it's not there yet), or maybe the dining table. The user hasn't checked the dining table yet. [...] So the next step should be to go to the dining table. If the potato isn't there, then systematically check the cabinets. [...] So first, find the potato, then heat it in the microwave, then put it in the fridge.
So first, find the potato. The user hasn't checked the dining table yet. Let's go there next.
</think>
`Action: go to diningtable 1`

How the correction is obtained

Deletion is measured on every turn of the seven successful rollouts (117 turns, 49 with $d^{\text{del}} = 1$); their mean is 0.42 and standard deviation $\sigma = 0.49$, so in every rollout
needed ($d = 1$): $c = (1 - 0.42)/0.49 = +1.18$
other ($d = 0$): $c = (0 - 0.42)/0.49 = -0.85$

Rollout 6: succeeds on its last turn

30 turns, 23 of them searching; deletion marks the same seven actions.

turns	c_{it}	GRPO	RMPO
7 needed	$+1.18$	$+0.38$	$+1.56$
23 search	-0.85	$+0.38$	-0.47

Figure 7: **Action deletion in ALFWorld.** One training group of eight rollouts. Replaying rollout 1 without each action shows which turns the plan needed ($d^{\text{del}} = 1$, shaded). After normalization the needed actions are lifted above the episode advantage, while the four search turns and the redundant `open microwave 1` (turn 8: ALFWorld permits heating without first opening the appliance) fall below it. This is hindsight necessity, not ex-ante decision quality: the unsuccessful searches were reasonable under partial observability, while `go to diningtable 1` is rewarded because the potato is there in this realized layout.

WebShop · measurements `ws_train_6756` `8 rollouts` `scores 1.00, 0.64, 0 × 6`

Instruction

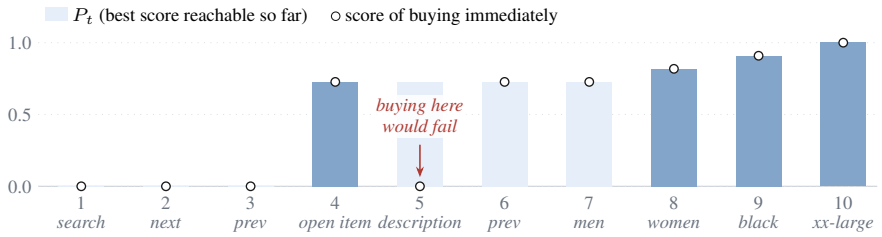
```
WebShop [SEP] Instruction: [SEP] Find me officially licensed, machine wash men's t-shirts & tanks with
polyester heathers, heathers cotton, cotton heather, needle sleeve, classic fit with color: black, and
fit type: women, and size: xx-large, and price lower than 60.00 dollars [SEP] Search
```

Action replacement: the items on the results page the rollouts returned to

	Candidate	*Item*	$V(a')$	*- - - mean*	d^{rep}	*opened by*
1	`click[b08226ndzw]`	Disney Winnie The Pooh Tigger Upside Down Portrait Tank Top	1.000		+0.44	rollouts 1, 5
2	`click[b09rvf4jp1]`	Once Upon A Time Queen Born In 1982 Tank Top	0.909		+0.35	rollout 5
3	`click[b07jvvdj6l]`	Canadian Tuxedo T Shirt	0.000		−0.56	rollout 5
4	`click[b09p7h5yk7]`	It's In My DNA Palestinian Shirt Tank Top	0.909		+0.35	rollout 5
5	`click[b07xpr3r7n]`	Marvel Avengers: Endgame Captain America T-Shirt	0.000		−0.56	--
not among the first five:						
6	`click[b09p39qn2w]`	I Tackled 100 Day Of School Football Boy 100th Day School T-Shirt	0.000		−0.47	rollouts 0, 4, 6, 7

$V(a')$: score of opening the item and deterministically completing its options (no model calls). The first five items form the candidate set (mean 0.564); an opened item outside them joins the set (mean 0.470).

Prefix progress along successful rollout 1



d^{prog} is the increase of P_t: $+0.73$ at the item click (turn 4), $+0.09$ at each of `women`, `black`, `xx-large`, and 0 elsewhere; leaving the product page (turn 5) or choosing the wrong fit type (turn 7) costs nothing.

Figure 8: **Replay measurements in WebShop.** One training group. *Top*: action replacement compares an opened item with the items listed on the same page. *Bottom*: prefix progress scores each prefix of rollout 1 by the best purchase reachable so far; this score only rises at the turns that change what can be bought. Figure 9 shows how these measurements change the turn-level advantages.

WebShop · turn-level advantages

`ws_train_6756` same group as Figure 8

The rollout scores 1.00, 0.64, and 0 (six times) give episode advantages $+2.17$, $+1.19$, and -0.56. Across all valid measurements in this group, the rounded per-rule statistics are $\mu_{\mathcal{G}}^{\text{prog}} = 0.04$, $\sigma_{\mathcal{G}}^{\text{prog}} = 0.20$ and $\mu_{\mathcal{G}}^{\text{rep}} = -0.10$, $\sigma_{\mathcal{G}}^{\text{rep}} = 0.44$. Thus progress values 0, 0.09, and 0.73 normalize to -0.20, $+0.25$, and $+3.45$, while replacement values $+0.44$, -0.47, and -0.56 normalize to $+1.23$, -0.84, and -1.05.

Successful rollout 1

bars: turn-level advantage; GRPO RMPO

Rollout	*Turns*	*Actions*	d^{prog}	d^{rep}	c_{it}	*GRPO*	*RMPO*
1	1–3	`search; next; prev`	0.00	*none*	-0.20	$+2.17$	$+1.97$
1	4	`open Disney Tigger tank top`	$+0.73$	$+0.44$	$+2.34$	$+2.17$	$+4.51$
1	5–7	`description; prev; men`	0.00	*none*	-0.20	$+2.17$	$+1.97$
1	8–10	`women; black; xx-large`	$+0.09$	*none*	$+0.25$	$+2.17$	$+2.42$

Rollout 0: an unhelpful outside item

The opened item is outside the first five and has $d^{\text{prog}} = 0$, $d^{\text{rep}} = -0.47$. Hence $c = (-0.20 - 0.84)/2 = -0.52$, and its turn-level advantage is $-0.56 - 0.52 = \mathbf{-1.08}$ rather than GRPO's -0.56.

Rollout 5: revisiting an inferior item

Once a better completion is already available, opening the Canadian Tuxedo item gives $d^{\text{prog}} = 0$ and $d^{\text{rep}} = -0.56$. Thus $c = (-0.20 - 1.05)/2 = -0.63$, lowering the advantage from $+1.19$ to $+\mathbf{0.56}$ for that turn.

Figure 9: **Turn-level advantages in WebShop**, from the measurements in Figure 8. Group-level normalization strongly rewards the product click that makes a high-scoring purchase reachable, gives smaller positive corrections to the option choices that improve the completion, and downweights zero-progress or below-average product choices. When both progress and replacement are valid at an item click, c_{it} is their average.

Search-QA · measurements

HotpotQA `train_26369` EM: 7 of 8

Question

`Question: Stony Awards presented a posthumous award to what American cannabis rights activist and the author of "The Emperor Wears No Clothes",?`
Gold answer: **`Jack Herer`** *closed-book reader answer:* `Rudolph Giuliani` ($V = 0$)

Sibling queries from the same evidence state, and the value V of the evidence they return

	First query	*Rollouts*	*Passages retrieved (titles)*	V	d^{rep}
q_a	**`Stony Awards posthumous award cannabis rights activist`**	0, 3, 4, 5	Asma Jahangir; Stony Awards; List of cannabis competitions	0.00	−0.50
q_b	**`Stony Awards posthumous award cannabis activist`**	1, 7	Stony Awards; List of cannabis competitions; High Times	0.00	−0.50
q_c	**`Stony Awards cannabis activist "The Emperor Wears No Clothes"`**	2, 6	**Jack Herer** (two passages); The Emperor Wears No Clothes	1.00	+0.50
q_0	*the question text itself (fixed candidate)*	–	**Jack Herer** (two passages); Chris Conrad (author)	1.00	–

The candidate set at the first turn is $\{q_a, q_b, q_c, q_0\}$ (mean $V = 0.5$). Second queries are compared the same way: after q_a both follow-ups (q'_a, q''_a) retrieve the Jack Herer article ($V = 1$); after q_b both follow-ups return the same passages again ($V = 0$); with no spread among siblings, $d^{\text{rep}} = 0$ for every second query.

What rollout 0 saw before its second search

```
Results (turn 2 of 4):
[Asma Jahangir] [...] she was awarded United
Nations Human Rights Award posthumously [...]
[Stony Awards] [...] Thomas King Forçade Award
Named after High Times founder Tom Forçade, the
award is for "stony achievement" in film. [...]
2006: Jeff "The Dude" Dowd for The Big Lebowski
[...]
[List of cannabis competitions] [...] Stony
Awards Taste of Texas Hemp Cup [...]
Action: search[American cannabis rights activist
The Emperor Wears No Clothes]
```

Why the two searches of rollout 0 differ

Turn 1. From these three passages the reader answers "Jeff *The Dude* Dowd": $V(E_1) = 0$. A sibling query from the same state (q_c) found the answer at once, so the query is scored below its siblings.

Turn 2. The results now open with *"[Jack Herer] Jack Herer (June 18, 1939 – April 15, 2010), sometimes called the Emperor of Hemp, was an American cannabis rights activist [...]"*; the reader answers Jack Herer: $V(E_2) = 1$, $d^{\text{prog}} = +1$.

Figure 10: **Replay measurements in Search-QA.** One training group. *Top*: queries issued from the same evidence state are compared by the value V of the evidence they return (the F1 of the answer read off that evidence). *Bottom*: prefix progress along rollout 0; Figure 11 shows the resulting advantages.

Search-QA · turn-level advantages — HotpotQA `train_26369` — same group as Figure 10

Four representative rollouts — *bars: turn-level advantage; GRPO RMPO*

Rollout	Turn	Action	d^{prog}	d^{rep}	c_{it}	GRPO	RMPO
0 (✓)	1	`search[Stony Awards posthumous award cannabis rights activist]`	0.00	−0.50	−0.95	+0.38	−0.58
	2	`search[American cannabis rights activist The Emperor Wears No Clothes]`	+1.00	0.00	+0.83	+0.38	+1.21
	3	`answer[Jack Herer]`	*none*	*none*	0	+0.38	+0.38
2 (✓)	1	`search[Stony Awards cannabis activist "The Emperor Wears No Clothes"]`	+1.00	+0.50	+1.59	+0.38	+1.97
	2	`answer[Jack Herer]`	*none*	*none*	0	+0.38	+0.38
7 (✓)	1	`search[Stony Awards posthumous award cannabis activist]`	0.00	−0.50	−0.95	+0.38	−0.58
	2	`search[Stony Awards posthumous award cannabis activist author]`	0.00	0.00	−0.20	+0.38	+0.18
	3	`search[Stony Awards posthumous award cannabis activist book]`	0.00	0.00	−0.20	+0.38	+0.18
	4	`answer[Jack Herer]`	*none*	*none*	0	+0.38	+0.38
1 (✗)	1	`search[Stony Awards posthumous award cannabis activist]`	0.00	−0.50	−0.95	−2.65	−3.60
	2	`search[Stony Awards posthumous award cannabis activist author]`	0.00	0.00	−0.20	−2.65	−2.84
	3	`search[Stony Awards posthumous award cannabis activist book]`	0.00	0.00	−0.20	−2.65	−2.84
	4	`answer[James O'Keefe]`	*none*	*none*	0	−2.65	−2.65

Figure 11: **Turn-level advantages in Search-QA**, from the measurements of Figure 10; c_{it} averages the normalized progress and replacement measurements of a turn. GRPO gives every turn of rollout 0 the same advantage; RMPO separates its two searches: the first is scored against a sibling query that answered the question at once, the second is where the evidence arrived. Rollout 2's single query, which found the answer at once, receives the largest correction. Rollout 7 answered correctly although its retrievals never helped the reader; its first query is scored below its siblings and its repeated queries slightly below the group average. Answer turns have no replay comparison and keep the episode advantage. The other rollouts of the group repeat rollouts 0 or 2 up to the wording of the second query.