SHARED RESOURCES DISTORT CREDIT ASSIGNMENT IN GROUP-RELATIVE AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Group-relative reinforcement learning trains language-model agents by comparing rewards across rollouts of the same prompt. We identify resource-state interference, in which shared quotas, queues, and mutable tool state make a rollout's reward depend on other rollouts' actions. A rollout can thus influence its own evaluation baseline by altering sibling rewards. Our theoretical analysis distinguishes independent-deployment performance from shared-service welfare and provides finite-group constructions in which expected group-relative updates oppose the gradient of either objective. To address this problem, we develop receipt-conditioned credit control, using resource receipts to identify affected rewards and selectively replay trajectories from reference states. This enables either censoring or propensity-corrected credit estimation. Resource receipts and fixed-action paired replay isolate execution effects on outcomes, credit, and parameter gradients. Search audits on Qwen3 and Llama-3.1 show that shared execution distorts raw parameter gradients. In matched training with Qwen3-8B and Llama-3.1-8B, shared execution lowers retrieval NDCG@3 by 10.74 and 35.63 points, respectively, relative to isolation. Partial replay improves raw-gradient alignment with paired isolated execution. In separate matched training experiments, it improves retrieval performance over shared execution while using approximately 38% fewer training retrieval executions than isolation on both models.

1 INTRODUCTION

Group-relative policy optimization has become a standard route to reinforcement learning for large language models. For each prompt, the learner samples a group of trajectories, compares their rewards, and assigns positive or negative credit without a learned critic (Shao et al., 2024). Agentic variants extend this recipe to search, software, and interactive environments (Jin et al., 2025; Feng et al., 2026). The statistical picture is simple. Siblings start from the same prompt, differ through policy sampling, and are treated as providing comparable outcomes.

However, rollouts sampled for the same prompt may interact through the resources used to execute their actions. In an agent RL training report covering models up to 397B, Mercor and SkyRL describe judge API rate limits and tool-server connection failures under high rollout concurrency (Mercor & SkyRL, 2026). When rollouts share quotas, queues, or mutable tool state, one rollout can change the observations and rewards received by its siblings. A rollout can thus influence its own evaluation baseline by altering sibling rewards. We call this dependence resource-state interference. Figure 1 shows this effect in search audits: shared quotas can reverse centered credit and alter it even when a rollout's own reward is unchanged. Interpreting this distortion requires specifying the deployment objective. We distinguish performance under independent execution in a reference resource state from collective welfare under the shared execution regime. We show that resource-state interference can make expected group-relative updates oppose the gradient of either deployment objective.

For independent deployment, the natural reference is the reward each trajectory would receive without sibling-induced changes to resource state. Full trajectory isolation provides these outcomes for every sibling, but requires private tool execution or environment state throughout rollout collection. Censoring affected rewards requires no additional tool calls, but can leave groups with little informative

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.

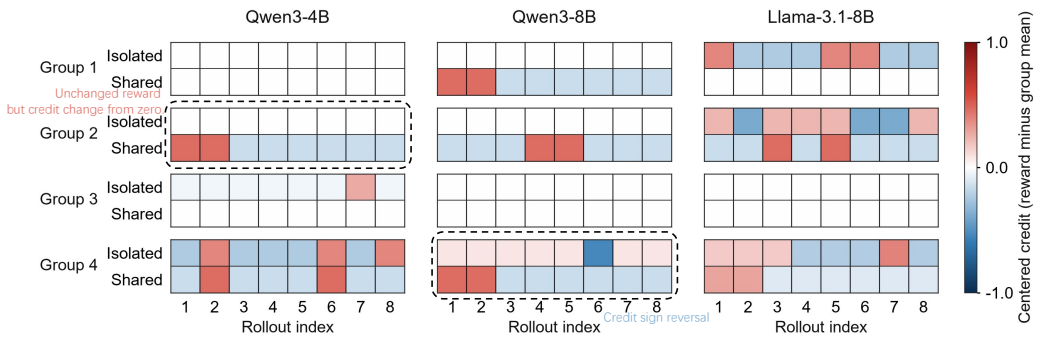


Figure 1: Shared quotas distort group-relative credit. Each row pair compares centered credit ($c_i = R_i - \bar{R}$) for eight fixed rollouts under isolated execution and a shared two-call quota. Sharing can reverse credit signs and alter credit even when a rollout's own reward is unchanged.

credit. Replaying all affected trajectories supplies the missing reference outcomes at the cost of additional tool executions.

To recover this reference comparison, we introduce receipt-conditioned credit control, which uses resource receipts to identify affected rewards and guide censoring or selective reference-state replay. The receipts record each trajectory's resource exposure, while fixed-action paired replay measures how changing that exposure affects outcomes, group credit, and raw parameter gradients. For partial replay, we use known observation propensities to construct credit from the available reference-state rewards. We evaluate the mechanism through search audits on Qwen3 and Llama-3.1, matched policy training, and stateful tasks involving resource competition and shared benefits. In matched training, shared execution lowers retrieval NDCG@3 relative to isolation by 14.58 points on Qwen3-4B, 10.74 points on Qwen3-8B, and 35.63 points on Llama-3.1-8B. Selective reference-state replay improves alignment with paired isolated-execution gradients. In separate matched training experiments on both 8B models, it improves retrieval performance over shared execution while using approximately 38% fewer training retrieval executions than isolation. With matched retrieval budgets, propensity-corrected and unweighted targeted replay achieve similar end-of-training NDCG on the monitoring sets of both 8B models. Our contributions are as follows:

1. **Resource-state interference and gradient mismatch.** We formalize resource-state interference in group-relative agent RL and distinguish independent-deployment performance from shared-service welfare. We provide a gradient decomposition and exact finite-group constructions showing that expected group-relative updates can oppose the gradient of either objective.
2. **Receipt-based paired-replay audit.** We develop a receipt-based paired-replay audit that holds policy actions fixed while varying resource exposure and dispatch order, tracing interference from execution outcomes to group credit and parameter gradients. Mechanism-matched negative controls make the diagnosis falsifiable.
3. **Learning degradation and reward free-riding.** We show empirically that resource-state interference changes rollout outcomes and actual parameter gradients, producing learned-policy loss under resource contention and reward free-riding under persistent state through the same failure of credit attribution.
4. **Receipt-conditioned credit control.** We propose receipt-conditioned credit control, which uses resource receipts either to censor altered labels or to form propensity-corrected credit from selective reference-state replay. Partial replay restores alignment with the isolated-target gradient and recovers held-out policy performance with fewer calls than full replay.

2 Related Work

Group-relative estimation and optimization. GRPO uses within-prompt reward normalization, whereas RLOO constructs leave-one-out baselines (Shao et al., 2024; Ahmadian et al., 2024).

Dr.GRPO revisits normalization-induced length and difficulty biases (Liu et al., 2025). Under i.i.d. output–reward sampling conditional on the prompt, U-statistic analyses characterize the finite-sample and asymptotic behavior of group-relative estimators (Zhou et al., 2026). Token-level studies examine likelihood displacement and develop token reweighting and likelihood-preserving regularization (Deng et al., 2025; 2026b;a).

Structured and counterfactual credit assignment. MT-GRPO combines turn-level and outcome advantages for multi-turn agent training (Wei et al., 2026). AgentPRM learns process rewards that capture the promise and progress of individual decisions (Xi et al., 2026). GiGPO and RTMC construct credit from rollouts with matching intermediate states (Feng et al., 2026; Wang et al., 2026). HGPO incorporates trajectory history, and Tree-GRPO uses intra- and inter-tree comparisons (He et al., 2026b; Ji et al., 2026). COMA uses a centralized critic to marginalize one agent's action while fixing the others' actions, yielding a counterfactual baseline (Foerster et al., 2018). CRAFT reuses sibling rollouts for counterfactual token credit (Meng & Chen, 2026). Branching Policy Optimization forks continuations from sandbox snapshots, whereas C3 restores interaction histories to compare alternative actions (He et al., 2026a; Chen et al., 2026).

Agent environments and rollout systems. Search-R1 trains agents to interleave reasoning and retrieval (Jin et al., 2025), while ToolSandbox and AppWorld evaluate tool use in stateful environments (Lu et al., 2025; Trivedi et al., 2024). Agent-R1 provides modular agent-RL interfaces, while MARS-SQL and LOOP train agents in database and multi-application environments, respectively (Cheng et al., 2025; Yang et al., 2025b; Chen et al., 2025). At the execution layer, Heddle optimizes rollout scheduling and placement for throughput (Zhang et al., 2026), while DeltaBox supports checkpoint and rollback of sandbox file and process state (Dong et al., 2026).

Causal interference. Causal inference studies interference when one unit's potential outcome depends on other units' treatments (Hudgens & Halloran, 2008). Exposure-mapping frameworks relate joint treatment assignments to individual exposures and use inverse-probability weighting to estimate causal effects (Aronow & Samii, 2017). Related work examines interference in networked settings and in the identification of dynamic treatment regimes (Ma & Tresp, 2021; Sherman et al., 2020). Hays and Raghavan model interactions mediated by shared state and develop estimators for direct and global effects (Hays & Raghavan, 2025).

3 METHOD

3.1 TWO DEPLOYMENT OBJECTIVES

For a prompt x, the policy independently samples G action sequences $A_i \sim \pi_\theta(\cdot \mid x)$ before execution. Let H_0 denote the initial resource state and σ the dispatch order. The distribution of H_0 is independent of θ. Execution and dispatch depend on θ only through the sampled actions. Executing the group produces

$$R_i(\mathbf{A}, \sigma, H_0), \qquad \mathbf{A} = (A_1, \ldots, A_G). \tag{1}$$

The outcome includes any verifier reward derived from the resulting observation. A fresh reference execution gives $R_i^{\mathrm{ref}}(A_i; H_{\mathrm{ref}})$. We say that sibling rollouts are non-interfering with respect to the reward when $R_i(\mathbf{A}, \sigma, H_0) = R_i^{\mathrm{ref}}(A_i; H_{\mathrm{ref}})$ for every sibling i, action vector $\mathbf{A}$, and admissible order σ.

These executions induce two possible objectives. Independent deployment evaluates a trajectory in the reference state, whereas a shared service may target mean welfare under group execution:

$$J_{\mathrm{ind}}(\theta) = \mathbb{E}_{x, A \sim \pi_\theta}\left[R^{\mathrm{ref}}(A; H_{\mathrm{ref}})\right], \tag{2}$$

$$J_{\mathrm{joint}}(\theta) = \mathbb{E}_{x, \mathbf{A}, \sigma, H_0}\left[\frac{1}{G}\sum_{i=1}^{G} R_i(\mathbf{A}, \sigma, H_0)\right]. \tag{3}$$

The objectives coincide under reward-level non-interference and can otherwise differ, as illustrated in Appendix A.6. The shared-welfare gradient includes cross-rollout effects. Writing

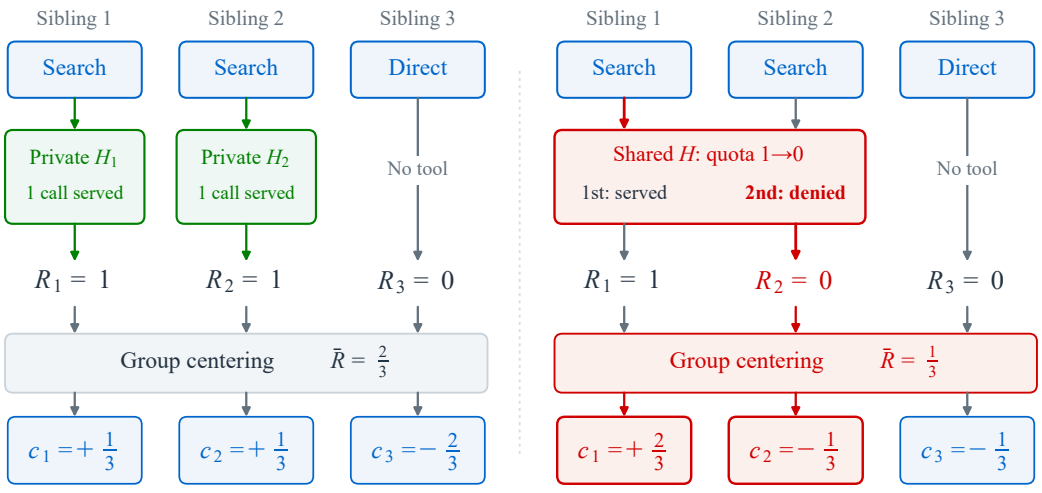


Figure 2: With actions and dispatch order fixed, a shared one-call quota (b) changes centered credit relative to private quotas (a): sibling 2's credit flips sign, while sibling 1's doubles despite its unchanged reward. Rewards are illustrative, with $c_i = R_i - \bar{R}$.

$S_i = \nabla_\theta \log \pi_\theta(A_i \mid x)$, the gradient is

$$\nabla_\theta J_{\text{joint}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\sum_{j=1}^{G} R_j(\mathbf{A}, \sigma, H_0) S_i\right]. \tag{4}$$

The terms with $j \neq i$ are spillover terms, because an action can affect sibling rewards.

3.2 Gradient Misalignment under Resource Interference

For rewards $R_1, \ldots, R_G$, the sequence-level GRPO estimator is

$$\widehat{A}_i = \frac{R_i - \overline{R}}{s_R + \epsilon}, \qquad \widehat{g}_{\text{GRPO}} = \frac{1}{G}\sum_i \widehat{A}_i \nabla_\theta \log \pi_\theta(A_i \mid x). \tag{5}$$

Here, $\overline{R} = G^{-1}\sum_i R_i$ and $s_R^2 = G^{-1}\sum_i (R_i - \overline{R})^2$. Under shared execution, rewards can differ from their reference-state counterparts, and each action can change its own baseline by altering sibling rewards. Standardization further couples updates through a nonlinear function of the reward vector. The first two mechanisms persist without standardization and can misalign the expected update with either deployment gradient. Figure 2 illustrates this baseline dependence: sharing a quota changes centered credit even for a sibling whose reward is unchanged.

Define

$$g_{\text{own}} = \mathbb{E}\left[\frac{1}{G}\sum_i R_i S_i\right], \qquad g_{\text{spill}} = \mathbb{E}\left[\frac{1}{G}\sum_i \sum_{j \neq i} R_j S_i\right], \tag{6}$$

Here, g_{own} pairs each action with its executed return, while g_{spill} collects cross-sibling score terms. For RLOO, direct expansion gives

$$\mathbb{E}[\widehat{g}_{\text{RLOO}}] = g_{\text{own}} - \frac{1}{G-1} g_{\text{spill}}. \tag{7}$$

Since Equation 4 gives $\nabla_\theta J_{\text{joint}} = g_{\text{own}} + g_{\text{spill}}$, defining $\Delta_{\text{res}} = g_{\text{own}} - \nabla_\theta J_{\text{ind}}$ shows that the errors relative to the independent and joint targets are $\Delta_{\text{res}} - \frac{1}{G-1} g_{\text{spill}}$ and $-\frac{G}{G-1} g_{\text{spill}}$, respectively. Both RLOO errors vanish under non-interference with a matched reference state. In this formulation, the centered update of Dr.GRPO is $(G-1)/G$ times the RLOO update, whereas standardized

GRPO additionally applies a random group-level scale that need not preserve the expected direction. Appendix A.1 gives the derivation.

This mismatch can reverse the expected update direction. For a Bernoulli policy with a scalar logit, we construct binary-action, binary-reward environments with a finite shared resource in which the expected GRPO update has the opposite sign from ∇J_{ind}, and other settings in which it has the opposite sign from $\nabla J_{\mathrm{joint}}$. The same reversals occur for centered and leave-one-out estimators that omit reward standardization.

We establish these claims by exact summation over every action and reward vector. The environment assigns success probability p_{ref} to the first q tool actions, p_{altered} to later tool actions, and p_{direct} to a direct action. Appendix A gives the complete estimator equations, constructive examples, and parameter grid.

3.3 Resource receipts and paired replay

Each tool call emits a versioned resource receipt linking the action, resource exposure, and execution outcome. Stable identifiers support pairing across execution conditions. Receipts record observations and task states through hashes or structured records. Appendix B describes the core receipt schema and pairing protocol.

We sample each trajectory's action sequence once and replay it under two resource-state conditions. The reference execution resets or namespaces the resource state for each trajectory. The coupled execution shares the resource under study across siblings while keeping other execution state trajectory-private. This pairing holds policy-action sampling fixed while allowing resource-dependent observations and rewards to change. We quantify the paired differences at two levels.

For a matched trajectory n, let $Y_n = (O_n, R_n)$ denote its execution outcome. Here, O_n records call status, observations, timeouts, and post-execution task state, and R_n is the reward. We measure the potential-outcome change rate (POCR) and order sensitivity (OS):

$$\mathrm{POCR} = \frac{1}{N}\sum_{n} \mathbb{1}\left[Y_n^{\mathrm{cpl}} \neq Y_n^{\mathrm{ref}}\right], \qquad \mathrm{OS} = \Pr\left[Y_i(\mathbf{A}, \sigma, H_0) \neq Y_i(\mathbf{A}, \sigma', H_0)\right]. \tag{8}$$

Here, N is the number of matched trajectories. The superscripts cpl and ref denote coupled and reference execution. For OS, the two executions hold $\mathbf{A}$ and H_0 fixed and differ only in dispatch order. Reward POCR denotes the same rate computed from the reward component alone.

We then compare group credit and raw parameter gradients across paired executions. For gradient comparisons, we hold policy parameters, the inputs used to compute action log probabilities, and action tokens fixed. Let $\mathbf{z}$ denote either the matched credit vector $\mathbf{c}$, which collects group-relative credit coefficients, or the raw parameter-gradient vector $\mathbf{g}$:

$$\mathrm{Cos}_z = \frac{\langle \mathbf{z}^{\mathrm{cpl}}, \mathbf{z}^{\mathrm{ref}} \rangle}{\|\mathbf{z}^{\mathrm{cpl}}\| \|\mathbf{z}^{\mathrm{ref}}\|}, \qquad \mathbf{z} \in \{\mathbf{c}, \mathbf{g}\}. \tag{9}$$

We report cosine as undefined when either vector has zero norm.

We obtain confidence intervals by cluster-bootstrapping matched execution pairs at the prompt or initial-state level. Mechanism-matched negative controls remove one interference path at a time: a no-tool task removes tool execution, $G = 1$ removes siblings, sufficient capacity removes quota exhaustion, unrelated cache keys remove cache reuse through key sharing, and trajectory-private state removes state sharing.

3.4 Receipt-conditioned credit control

Full trajectory isolation restores the execution regime underlying Equation 2, but requires trajectory-private tool execution or environment state for every sibling. We instead use receipts to decide whether a reward enters credit directly, enters after reference-state replay, or is censored.

The zero-replay option, resource-conditioned censoring (RCC), retains only trajectories whose receipts show no sibling-induced resource change before the outcome is produced and assigns zero advantage to the rest. RCC adds no tool calls, but sparse clean support can leave a group with little or no active credit.

Partial reference-state credit (PRSC) restores this support by replaying each altered trajectory from the reference state with probability $p \in (0, 1]$. Let $\mathcal{O}$ denote the trajectories for which a reference reward is observed. A clean trajectory has observation propensity $q_i = 1$, whereas an altered trajectory is observed only when selected for replay and therefore has propensity $q_i = p$.

Define $w_i = q_i^{-1}$ and $W = \sum_{j \in \mathcal{O}} w_j$. For $W > 0$, following self-normalized counterfactual estimation (Swaminathan & Joachims, 2015), the reference-state group moments are

$$\widehat{\mu} = \frac{1}{W} \sum_{i \in \mathcal{O}} w_i R_i^{\mathrm{ref}}, \qquad \widehat{s}^2 = \frac{1}{W} \sum_{i \in \mathcal{O}} w_i (R_i^{\mathrm{ref}} - \widehat{\mu})^2. \tag{10}$$

The corresponding credit coefficient is

$$\widehat{A}_i^{\mathrm{PRSC}} = \mathbb{1}[i \in \mathcal{O}] \frac{w_i (R_i^{\mathrm{ref}} - \widehat{\mu})}{\widehat{s} + \epsilon}. \tag{11}$$

When $W = 0$, all PRSC credit coefficients are set to zero. PRSC targets the full reference-state group credit and recovers it exactly at $p = 1$. Replay probability controls expected tool cost, while partial observation and self-normalization affect finite-group bias and variance (Appendix A.5).

False negatives admit altered rewards as reference outcomes, while false positives increase expected replay cost. Our receipt-noise experiments show greater sensitivity to false negatives, supporting high-recall detection (Section 4.5).

4 Experiments

Fixed-action paired replay first measures execution effects on outcomes and credit, with search audits tracing these effects to raw gradients. Matched search training then tests whether this distortion affects learned-policy performance and whether receipt-conditioned control mitigates it. Equal-cost replay comparisons further test the benefit of propensity weighting.

4.1 Experimental setup

Tasks and models. We evaluate search on HotpotQA, using its annotated supporting-document titles as relevance labels for retrieval evaluation (Yang et al., 2018). The agent generates a query, retrieves evidence from a local BM25 index of 2018 Wikipedia (Robertson & Zaragoza, 2009), and answers from the retrieved passages. Search experiments use Qwen3-1.7B, Qwen3-4B, Qwen3-8B (Yang et al., 2025a), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). To examine interference through persistent tool state, we use ToolSandbox's cellular and Wi-Fi setting tasks, with milestone-based evaluation of the resulting environment state (Lu et al., 2025). We also evaluate Qwen3-8B on multistep inventory-reservation and cache-refresh tasks, which capture resource competition and shared benefits, respectively.

Interference protocols. We replay fixed actions under isolated and shared execution. Search quota audits use eight rollouts per prompt. Isolation retrieves every valid query in a trajectory-private resource state. Shared execution orders queries by token length, serves the first two valid queries per group, and returns quota-exhausted observations for the rest. We vary dispatch order and use negative controls with capacity matched to group size or trajectory-private namespaces. Queue audits use two shared concurrent retrieval slots and deadlines of 75, 125, and 200 ms. ToolSandbox keeps conversations private and either resets setting state per rollout or carries it across siblings. Inventory and cache tasks use fixed-action replay under private and shared states, with separate adaptive runs allowing later actions to respond to tool observations. Appendix J gives sample sizes, seeds, and task-specific protocols.

Online training and evaluation. Online search training updates rank-8 LoRA adapters on the attention query and value projections using the policy loss on generated query tokens (Hu et al., 2022). Each update uses four prompts with eight rollouts per prompt, and matched conditions share initial parameters, prompt batches, and policy-sampling seeds. We compare shared execution, trajectory isolation, and receipt-conditioned credit control over 40- and 160-update runs. At matched retrieval cost, we also compare propensity-corrected and unweighted replay. Trained policies are evaluated

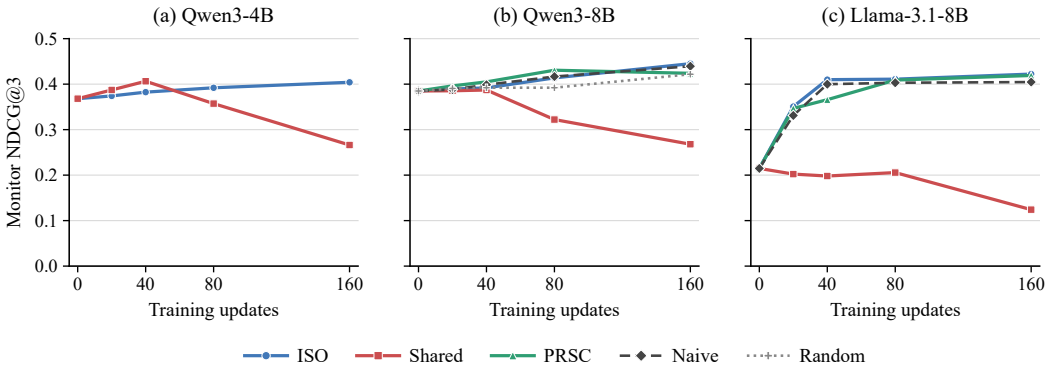


Figure 3: Learning curves under matched training, evaluated under isolated retrieval on the 64-question monitor. Shared execution yields lower final monitor NDCG@3 than isolation across all three models, while reference replay improves performance on both 8B models.

under isolated retrieval, with supporting-title NDCG as the primary metric. Additional evaluations measure answer containment, token F1, and exact match, and assess transfer to 2WikiMultihopQA (Ho et al., 2020). Appendix J provides reward and metric definitions, evaluation splits, training seeds, and hyperparameters.

4.2 Shared resources change outcomes and credit

Exact enumeration shows that shared resource state can make the expected GRPO update oppose the independent-deployment gradient. Across the parameter grid, the fraction of settings with sign reversals increases from 0.0% at $G = 2$ to 5.1% at $G = 4$ and 8.6% at $G = 8$. A separate Monte Carlo evaluation at $G = 16$ finds sign reversals in 26 of 50 cache and quota settings. Dr.GRPO (Liu et al., 2025) and RLOO (Ahmadian et al., 2024) also exhibit sign reversals. Appendix A gives the complete grid and validation.

Fixed-action replay on search trajectories shows how interference changes with group size. With the shared retrieval quota fixed at two, outcome POCR rises from 0% at $G = 1$ to 87.5% at $G = 16$. As group size increases from $G = 4$ to $G = 16$, F1-credit cosine with isolated execution decreases from 0.236 to 0.105. Matching quota capacity to group size makes both outcomes and F1-based credit identical to those under isolated execution (Appendix Table 8).

Dispatch order also changes credit allocation. Two random dispatch orders preserve the generated actions and nearly preserve aggregate answer quality, yet yield F1-credit cosines of 0.073 and 0.163 relative to token-length order.

4.3 Effects across models and resource mechanisms

Quota interference persists across model scales and families (Table 1). With eight siblings sharing two retrieval calls, shared execution reduces supporting-title NDCG and distorts group-relative credit across all four models. Outcome POCR reaches 75.0% on Qwen3-4B and Qwen3-8B and 61.7% on Llama-3.1-8B. Matching capacity to group size or using trajectory-private resource states eliminates these outcome and credit differences in these three models.

Table 1: Shared-quota effects in fixed-action audits. ΔNDCG is shared minus isolated (points). Credit reports cosine alignment with isolated credit. Grad. reports cosine alignment with isolated raw LoRA gradients.

Model	ΔNDCG	Credit	Grad.
Qwen3-1.7B	−29.01	0.070	−0.178
Qwen3-4B	−27.43	0.212	0.037
Qwen3-8B	−29.89	0.065	−0.062
Llama-3.1-8B	−14.91	0.273	0.015

The credit distortion reaches the raw LoRA gradients. Paired comparisons show negative alignment between shared and isolated gradients on Qwen3-1.7B and Qwen3-8B and near-orthogonal alignment on Qwen3-4B and Llama-3.1-8B.

Queue contention in BM25 retrieval also changes rollout outcomes and distorts group-relative credit. Extending the deadline from 75 to 200 ms reduces outcome differences and improves credit alignment between shared and isolated execution.

In ToolSandbox, persistent shared state allows later rollouts to receive rewards for settings changed by earlier siblings. On Qwen3-1.7B, this free-riding raises mean reward by 13.70 points, and about half of the active credit entries differ in sign from their isolated counterparts. Qwen3-4B attempts the setting change in every rollout. Later shared calls return an already-disabled error, but all rollouts receive full reward under both execution conditions, leaving group-relative credit at zero.

Multistep tasks expose both resource competition and shared benefits (Table 2). Inventory success is 37.5% under fixed shared replay and 75.0% when agents can adapt to tool observations, compared with 100% under isolation. Shared caching maintains 100% success while reducing origin calls from eight to one per group and increasing group welfare from 7.2 to 7.9. Reversing dispatch order changes credit allocation without changing group welfare in either task.

Table 2: Multistep tasks on Qwen3-8B. Welfare sums the eight rollout rewards. Order cos. is the credit cosine between forward and reverse dispatch orders.

Task	Execution	Welfare	Order cos.
Inventory	ISO	8.0	—
	Shared (fixed)	3.0	−0.600
	Shared (adaptive)	6.0	−0.333
Cache	ISO	7.2	—
	Shared (fixed/adapt.)	7.9	−0.143

4.4 Learned-policy effects and receipt-conditioned control

We evaluate trained policies under isolated retrieval to assess how resource-state interference affects learning. After 160 updates, policies trained in isolation improve over initialization, whereas those trained with shared resources fall below their initial performance on average. Relative to isolation, shared execution lowers retrieval NDCG@3 by 14.58 points on Qwen3-4B, 10.74 points on Qwen3-8B, and 35.63 points on Llama-3.1-8B (Table 3).

Table 3: NDCG@3 (0–100) before and after training. Naive uses unweighted targeted replay, while Random uses unweighted random replay.

Model	Initial	ISO	Shared	PRSC	Naive	Random
A. Effects of shared execution: historical evaluation, 448 questions						
Qwen3-4B	38.90	43.51	28.94	—	—	—
Qwen3-8B	42.66	44.07	33.33	—	—	—
B. Replay comparison						
Qwen3-8B	38.49	44.50	26.79	42.38	43.94	42.18
Llama-3.1-8B	24.16	49.86	14.24	49.93	49.23	47.99

Selective reference replay improves retrieval performance (Figure 3). In a separate Qwen3-8B study, PRSC raises monitor NDCG@3 by 15.59 points over shared execution, partially closing the gap to isolation. On Llama-3.1-8B, it raises final NDCG@3 by 35.69 points and brings performance within one point of isolation. It uses 37.73% and 37.80% fewer training retrieval executions than isolation on Qwen3-8B and Llama-3.1-8B, respectively.

Reference replay also improves raw-gradient alignment with isolated execution (Figure 4). In the 40-update Qwen3-8B audit, PRSC and Shared training checkpoints have mean raw-gradient cosines of 0.982 and 0.291 with their respective paired isolated references. PRSC has lower relative gradient error than Shared and RCC at every checkpoint. At Llama-3.1-8B initialization, PRSC raises this cosine from 0.337 to 0.922. Appendices D and I report the paired audits.

At equal retrieval cost, unweighted replay exceeds PRSC by 1.55 NDCG@3 points at the Qwen3-8B monitor endpoint, while PRSC has the higher learning-curve AUC. On Llama-3.1-8B, PRSC leads targeted unweighted replay and random replay by 0.70 and 1.94 points, respectively, on the independent final evaluation. Targeted unweighted replay has a lower monitor endpoint and AUC than PRSC. Both weighted and unweighted reference replay improve over shared execution in these matched studies, with their relative policy performance varying across evaluations. In the finite-group audit, PRSC has lower estimator bias and higher single-draw mean squared error than equal-budget unweighted replay (Appendix A.5).

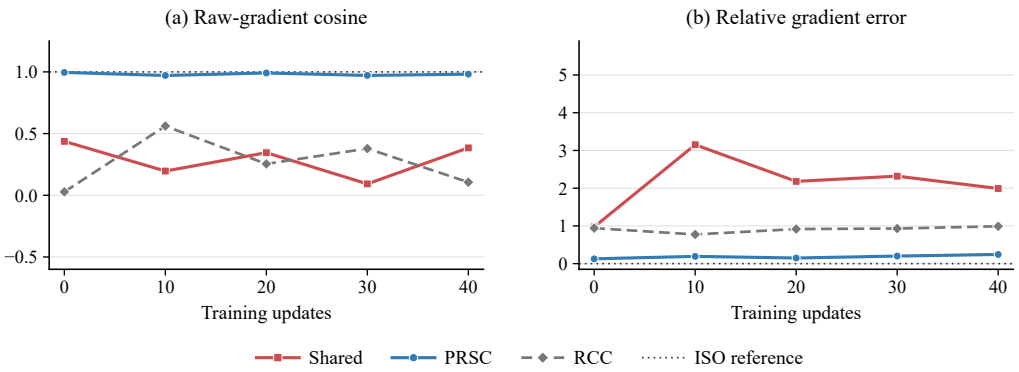


Figure 4: Raw-gradient correction across Qwen3-8B training checkpoints. PRSC has higher cosine alignment and lower relative error than Shared and RCC at every checkpoint.

The earlier 40-update studies show more variable policy effects. On Qwen3-1.7B, shared execution lowers NDCG by 2.79 points and PRSC recovers 95.7% of the gap (Appendix Table 12). Reevaluating the Qwen3-4B and Qwen3-8B models gives shared-minus-isolated NDCG@3 differences of -0.23 and $+0.54$ points. PRSC finishes 0.50 and 0.10 points below isolation, respectively.

4.5 Robustness and Scope

Receipt errors affect credit correction asymmetrically. Under independent replay, injecting 40% false negatives reduces mean-estimator NDCG-credit cosine from 0.987 to 0.674 on Qwen3-4B and from 0.991 to 0.531 on Qwen3-8B. The same false-positive rate leaves alignment nearly unchanged. Missed exposure leaves resource-altered rewards in credit estimation, whereas false positives mainly incur unnecessary replay (Table 4). These results favor high-recall exposure detection when selecting trajectories for reference replay.

Table 4: Receipt quality. Cosine similarity between isolated credit and PRSC credit averaged over independent replay draws.

Receipts	Qwen3-4B	Qwen3-8B
Correct	0.987	0.991
40% false negatives	0.674	0.531
40% false positives	0.989	0.993
40% stale epochs	0.831	0.779

Replay dependence also affects estimation error. Mean-credit alignment captures the direction of the average estimate, while single-draw error measures the accuracy of individual replay estimates. Even with correct receipts, batch-correlated replay increases single-draw relative mean squared error on both models while retaining high alignment of the mean credit estimate. Stale resource epochs and misspecified observation propensities further affect the estimates. Replay sampling therefore matters alongside receipt accuracy (Appendix F.3).

Policy effects also vary across evaluation tasks. In the original Qwen3-1.7B transfer evaluation on 2WikiMultihopQA, Shared-minus-isolated and PRSC-minus-isolated NDCG differences are $+0.038$ and $+0.266$ points, respectively. The point estimates remain close to isolation on this benchmark (Appendix Table 22).

5 Conclusion

Shared resources make sibling rewards interdependent and can distort group-relative credit. Our gradient decomposition and exact finite-group constructions show that expected group-relative updates can oppose the gradient of either independent-deployment performance or shared-service welfare. Our audits and matched training studies connect this mismatch to gradient distortion and retrieval degradation on Qwen3 and Llama-3.1. Selective reference replay improves learning over shared execution while using fewer retrieval executions than isolation. These findings make execution state an explicit part of training design and motivate matching credit estimation to the intended deployment objective.

AI USE STATEMENT

We used generative AI tools for language polishing, literature search and reference checking, and table preparation. The authors take responsibility for the final content, including the methods, experimental results, and conclusions.

REPRODUCIBILITY STATEMENT

Source code is provided in the supplementary material. Appendix A provides the theoretical derivations and finite-group constructions. Experimental protocols and evaluation procedures are documented in Appendices B–E and I. Appendix J summarizes experiment configurations and evaluation splits.

REFERENCES

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

Peter M. Aronow and Cyrus Samii. Estimating average causal effects under general interference, with application to a social network experiment. *The Annals of Applied Statistics*, 2017.

Kevin Chen, Marco Cusumano-Towner, Brody Huval, Aleksei Petrenko, Jackson Hamburger, Vladlen Koltun, and Philipp Krähenbühl. Reinforcement learning for long-horizon interactive LLM agents. *arXiv preprint arXiv:2502.01600*, 2025.

Yanjun Chen, Yirong Sun, Hanlin Wang, Jinghan Wang, Xinming Zhang, Xiaoyu Shen, Wenjie Li, and Wei Zhang. Exact is easier: Credit assignment for cooperative LLM agents. *arXiv preprint arXiv:2603.06859*, 2026.

Mingyue Cheng, Shuo Yu, Daoyu Wang, Qingchuan Li, Xiaoyu Tao, Jie Ouyang, Yucong Luo, Yitong Zhou, Qi Liu, and Enhong Chen. Agent-R1: A unified and modular framework for agentic reinforcement learning. *arXiv preprint arXiv:2511.14460*, 2025.

Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025*, 2025.

Wenlong Deng, Yushu Li, Boying Gong, Yi Ren, Christos Thrampoulidis, and Xiaoxiao Li. On group relative policy optimization collapse in agent search: The lazy likelihood-displacement. In *International Conference on Machine Learning*, 2026a.

Wenlong Deng, Yi Ren, Yushu Li, Boying Gong, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. Token hidden reward: Steering exploration-exploitation in group relative deep reinforcement learning. In *International Conference on Learning Representations*, pp. 122109–122133, 2026b.

Yunpeng Dong, Jingkai He, Shiqi Liu, Yuze Hou, Dong Du, Zhonghu Xu, Si Yu, Baochuan Yang, Yubin Xia, and Haibo Chen. DeltaBox: Scaling stateful AI agents with millisecond-level sandbox checkpoint/rollback. *arXiv preprint arXiv:2605.22781*, 2026.

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, pp. 46375–46408, 2026.

Jakob N. 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.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Chris Hays and Manish Raghavan. Double machine learning for causal inference under shared-state interference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 22318–22350. PMLR, 2025.

Bowei He, Yankai Chen, Xiaokun Zhang, and Xue Liu. Branching policy optimization: Sandbox-native language agent reinforcement learning. *arXiv preprint arXiv:2607.14171*, 2026a.

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

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625, 2020.

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. In *International Conference on Learning Representations*, 2022.

Michael G. Hudgens and M. Elizabeth Halloran. Toward causal inference with interference. *Journal of the American Statistical Association*, 103(482):832–842, 2008.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. In *International Conference on Learning Representations*, pp. 87362–87388, 2026.

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 preprint arXiv:2503.09516*, 2025.

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. In *Second Conference on Language Modeling*, 2025.

Jiarui Lu, Thomas Holleis, Yizhe Zhang, Bernhard Aumayer, Feng Nan, Felix Bai, Shuang Ma, Shen Ma, Mengyu Li, Guoli Yin, Zirui Wang, and Ruoming Pang. ToolSandbox: A stateful, conversational, interactive evaluation benchmark for LLM tool use capabilities. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 1160–1183, 2025.

Yunpu Ma and Volker Tresp. Causal inference under networked interference and intervention policy enhancement. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 3700–3708. PMLR, 2021.

Zibin Meng and Kani Chen. CRAFT: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning. *arXiv preprint arXiv:2606.29476*, 2026.

Mercor and SkyRL. Training frontier knowledge work agents: A 397B RL training guide with SkyRL. Mercor Blog, 2026. URL `https://www.mercor.com/blog/training-frontier-knowledge-work-agents-a-397b-rl-training-guide-with-skyrl/`. Published September 1, 2026.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 4(1-2):1–174, 2009. doi: 10.1561/1500000019.

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

Eli Sherman, David Arbour, and Ilya Shpitser. General identification of dynamic treatment regimes under interference. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pp. 3917–3927. PMLR, 2020.

Adith Swaminathan and Thorsten Joachims. The self-normalized estimator for counterfactual learning. In *Advances in Neural Information Processing Systems*, volume 28, 2015.

Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. AppWorld: A controllable world of apps and people for benchmarking interactive coding agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16022–16076, 2024.

Tao Wang, Suhang Zheng, and Xiaoxiao Xu. RTMC: Step-level credit assignment via rollout trees. *arXiv preprint arXiv:2604.11037*, 2026.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via fine-grained reward structure and credit assignment. arXiv preprint arXiv:2505.11821v3, 2026.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, 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. In *Proceedings of the ACM Web Conference 2026*, pp. 4184–4195. ACM, 2026. doi: 10.1145/3774904.3792551.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Haolin Yang, Jipeng Zhang, Zhitao He, Alexander Zhou, and Yi R. Fung. MARS-SQL: A multi-agent reinforcement learning framework for text-to-SQL. *arXiv preprint arXiv:2511.01008*, 2025b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, 2018.

Zili Zhang, Yinmin Zhong, Chengxu Yang, Chao Jin, Bingyang Wu, Xinming Wei, Yuliang Liu, and Xin Jin. Heddle: A distributed orchestration system for agentic RL rollout. *arXiv preprint arXiv:2603.28101*, 2026.

Hongyi Zhou, Kai Ye, Erhan Xu, Jin Zhu, Ying Yang, Shijin Gong, and Chengchun Shi. Demystifying group relative policy optimization: Its policy gradient is a U-statistic. *arXiv preprint arXiv:2603.01162*, 2026.

APPENDIX

A EXACT FINITE-GROUP ANALYSIS

A.1 GENERAL CENTERED-CREDIT DECOMPOSITION

For $G > 1$, let $\widehat{g}_{\mathrm{Dr.GRPO}} = \frac{1}{G}\sum_i (R_i - \overline{R})S_i$ and $\widehat{g}_{\mathrm{RLOO}} = \frac{G}{G-1}\widehat{g}_{\mathrm{Dr.GRPO}}$. Expanding the group mean and using the definitions in Equation 6 gives

$$\begin{aligned}
\mathbb{E}[\widehat{g}_{\mathrm{Dr.GRPO}}] &= g_{\mathrm{own}} - \mathbb{E}\left[\frac{1}{G}\sum_i \overline{R}S_i\right] \\
&= g_{\mathrm{own}} - \frac{1}{G^2}\mathbb{E}\left[\sum_i \sum_j R_j S_i\right] \\
&= g_{\mathrm{own}} - \frac{1}{G}\left(g_{\mathrm{own}} + g_{\mathrm{spill}}\right) \\
&= \frac{G-1}{G} g_{\mathrm{own}} - \frac{1}{G} g_{\mathrm{spill}}.
\end{aligned}$$

Multiplying by $G/(G-1)$ recovers Equation 7:

$$\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] = g_{\mathrm{own}} - \frac{1}{G-1} g_{\mathrm{spill}}.$$

Define $\Delta_{\mathrm{res}} = g_{\mathrm{own}} - \nabla_\theta J_{\mathrm{ind}}$. Using $\nabla_\theta J_{\mathrm{joint}} = g_{\mathrm{own}} + g_{\mathrm{spill}}$, the differences between the expected RLOO update and the two objective gradients are

$$\begin{aligned}
\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - \nabla_\theta J_{\mathrm{ind}} &= \Delta_{\mathrm{res}} - \frac{1}{G-1} g_{\mathrm{spill}}, \\
\mathbb{E}[\widehat{g}_{\mathrm{RLOO}}] - \nabla_\theta J_{\mathrm{joint}} &= -\frac{G}{G-1} g_{\mathrm{spill}}.
\end{aligned}$$

A.2 ENVIRONMENT AND ESTIMATORS

The exact environment has one binary policy action per sibling. Action $A_i = 1$ invokes the tool and $A_i = 0$ takes a direct route. We parameterize the policy by a scalar logit θ, with $p = \mathrm{sigmoid}(\theta)$. Each sibling independently samples $A_i \sim \mathrm{Bernoulli}(p)$. Indexing siblings in their fixed dispatch order, the rank of a tool action is

$$K_i = \sum_{j \le i} A_j. \tag{12}$$

The success probability is

$$\Pr(R_i = 1 \mid \mathbf{A}) = \begin{cases} p_{\mathrm{direct}} & A_i = 0, \\ p_{\mathrm{ref}} & A_i = 1 \text{ and } K_i \le q, \\ p_{\mathrm{altered}} & A_i = 1 \text{ and } K_i > q. \end{cases} \tag{13}$$

Conditional on the action vector, rewards are independent Bernoulli draws with the specified success probabilities. The threshold q separates tool calls in the reference and altered resource states. When $p_{\mathrm{altered}} < p_{\mathrm{ref}}$, this models quota exhaustion. When $p_{\mathrm{altered}} > p_{\mathrm{ref}}$, it models cache activation or another beneficial state transition.

For each action vector $\mathbf{a} \in \{0,1\}^G$ and reward vector $\mathbf{r} \in \{0,1\}^G$, we compute their exact joint probability. The policy score with respect to the logit is $\nabla_\theta \log \pi_\theta(a_i) = a_i - p$. Estimator coefficients are

$$c_i^{\mathrm{GRPO}}(\mathbf{r}) = \frac{r_i - \bar{r}}{s_r}, \tag{14}$$

$$c_i^{\mathrm{Dr.GRPO}}(\mathbf{r}) = r_i - \bar{r}, \tag{15}$$

$$c_i^{\mathrm{RLOO}}(\mathbf{r}) = \frac{G}{G-1}(r_i - \bar{r}), \tag{16}$$

$$c_i^{\mathrm{own}}(\mathbf{r}) = r_i. \tag{17}$$

Here $\bar{r} = G^{-1}\sum_i r_i$ and $s_r^2 = G^{-1}\sum_i (r_i - \bar{r})^2$. Exact enumeration uses $\epsilon = 0$ for nonzero-variance groups and assigns zero GRPO credit when $s_r = 0$. The exact expected update is

$$g_c = \sum_{\mathbf{a},\mathbf{r}} P_\theta(\mathbf{a}, \mathbf{r}) \frac{1}{G} \sum_i c_i(\mathbf{r})(a_i - p). \tag{18}$$

Under independent deployment, each tool action starts in the reference state, giving

$$g_{\text{ind}} = p(1-p)(p_{\text{ref}} - p_{\text{direct}}). \tag{19}$$

The shared-welfare gradient is

$$g_{\text{joint}} = \nabla_\theta J_{\text{joint}} = \mathbb{E}\left[\bar{R} \sum_i (A_i - p)\right]. \tag{20}$$

The numerical audit reports the scaled quantity $\widetilde{g}_{\text{joint}} = g_{\text{joint}}/G$. This positive scaling preserves the gradient sign. The shared-welfare counts below apply the magnitude threshold to this scaled quantity.

A.3 Constructive counterexamples

The sign-reversal claims in the main paper can be checked with explicit finite settings. In a beneficial cache setting, let

$$G = 4, \quad q = 1, \quad p = .25, \quad p_{\text{direct}} = .35, \quad p_{\text{ref}} = .225, \quad p_{\text{altered}} = .90. \tag{21}$$

Exact summation gives $g_{\text{ind}} = -0.02344$ and $g_{\text{GRPO}} = +0.00909$. Centered Dr.GRPO is $+0.00417$ and RLOO is $+0.00557$. All three reverse the independent target.

For a damaging quota, let

$$G = 4, \quad q = 1, \quad p = .75, \quad p_{\text{direct}} = .50, \quad p_{\text{ref}} = .82, \quad p_{\text{altered}} = .05. \tag{22}$$

Here $g_{\text{ind}} = +0.06000$ while $g_{\text{GRPO}} = -0.03111$. Dr.GRPO and RLOO are -0.01591 and -0.02121. The shared-welfare gradient is also negative, so all three updates oppose the independent-deployment gradient while aligning with the shared-welfare gradient.

A separate setting produces updates that oppose the shared-welfare gradient while aligning with the independent-deployment gradient. With $G = 2$, $q = 1$, $p = .7$, $p_{\text{direct}} = .2$, $p_{\text{ref}} = .1$, and $p_{\text{altered}} = .3$, exact summation gives $g_{\text{ind}} = g_{\text{GRPO}} = -0.0210$, while $g_{\text{joint}} = +0.0084$ and $\widetilde{g}_{\text{joint}} = +0.0042$. Dr.GRPO yields -0.0105, and RLOO yields -0.0210. All three updates therefore oppose the shared-welfare gradient.

A.4 Grid and validation

Table 5 reports independent-target reversals by group size. A setting is material when the target magnitude is at least 10^{-3}. A reversal additionally requires an estimator magnitude of at least 10^{-3} and the opposite sign. The target is g_{ind} for the independent comparison and $\widetilde{g}_{\text{joint}}$ for the shared-welfare comparison.

Table 5: Exact phase grid and independent-target sign reversals

Group size	Material settings	GRPO reversals	Rate
2	700	0	0.00%
4	1,050	54	5.14%
8	1,050	90	8.57%
Total	2,800	144	5.14%

The raw grid contains 3,000 settings with $G \in \{2, 4, 8\}$. We use $p_{\text{direct}} \in \{.2, .5, .8\}$, p_{ref} and $p_{\text{altered}} \in \{.1, .3, .5, .7, .9\}$, and $p \in \{.1, .3, .5, .7, .9\}$. Capacity is drawn from $\{1, G/2, G\}$ after removing duplicates. The independent-target filter excludes 200 settings with zero target gradient. On the same raw grid, GRPO reverses the scaled shared-welfare target in 202 settings that meet the magnitude thresholds. Dr.GRPO and RLOO reverse the independent target in 142 settings each.

The own-return estimator also reverses the independent target in 198 settings, because its executed rewards differ from the reference-state rewards.

For $G = 16$, exhaustive reward enumeration is expensive. We evaluate 50 canonical points with one million sampled groups per point. The cache mechanism reverses 14 of 25 points and quota reverses 12 of 25. We validate the Monte Carlo implementation at $G = 8$ against the exact calculation. The maximum absolute error over mean reward and all gradient estimators is 2.44×10^{-4}.

A.5 Finite-group estimator audit

For a fixed group of trajectories, the target is the standardized credit computed from all reference rewards. The weights in the moment estimates account for unequal reward-observation probabilities, while the weights in the credit coefficients account for missing trajectory contributions. The audit evaluates the resulting estimator over replay masks with the reference rewards and action-score contributions held fixed.

The fixed-population audit varies group size $G \in \{4, 8, 16\}$, nominal clean fraction in $\{.125, .25, .5, 1\}$, replay probability $p \in \{.1, .25, .5, .75, 1\}$, and harmful quota or beneficial cache interference. We evaluate the 120 settings with nine estimators: Shared, RCC, random same-budget replay, naive targeted replay, PRSC/Hájek, HT moments, Full, oracle-receipt PRSC, and actual-receipt PRSC. This yields 1,080 estimator cells. Each setting contains eight fixed groups. Actual clean counts are recorded when the nominal fraction cannot be represented exactly at a given group size. We enumerate the replay masks exactly, holding the finite population and action-score contributions fixed. For each fixed group, let $\widehat{g}$ and g_{ref} denote gradients formed from estimated and full-reference credit using the same fixed 12-dimensional score vectors. We define relative bias and single-draw relative MSE as

$$\text{Relative bias} = \frac{\|\mathbb{E}[\widehat{g}] - g_{\mathrm{ref}}\|_2}{\max(\|g_{\mathrm{ref}}\|_2, 10^{-15})},$$

$$\text{Relative MSE} = \frac{\mathbb{E}[\|\widehat{g} - g_{\mathrm{ref}}\|_2^2]}{\max(\|g_{\mathrm{ref}}\|_2^2, 10^{-15})}.$$

Expectations are taken over the replay-mask distribution. The mean-gradient cosine compares $\mathbb{E}[\widehat{g}]$ with g_{ref}.

Excluding $p = 1$ and fully clean controls leaves 72 settings per estimator. We first average each metric over the eight fixed groups within each setting, then average over the 72 settings. PRSC has mean relative bias 0.244, compared with 0.543 for same-budget targeted replay, 0.648 for random replay, 0.881 for shared credit, and 0.915 for RCC. Its mean-gradient cosine is 0.960. The corresponding single-draw relative MSE is 1.511, compared with 0.542 for targeted replay and 0.643 for random replay. Across all 72 settings, PRSC has lower bias and equal or higher single-draw MSE than targeted replay. At $p = 1$, PRSC and full replay agree exactly.

A.6 Independent and shared-welfare targets

We compare deployment objectives in an exactly enumerable cache environment with $G = 4$. Enumerating all 16 action vectors, 24 dispatch orders, and four initial states produces 1,536 outcomes. The two settings represent harmful eviction and beneficial refresh. Across nine gradient checks, the joint-welfare gradient agrees with exact enumeration and central finite differences to a maximum absolute error of 2.45×10^{-10}. Under the exact reference construction, PRSC and isolated-target training have identical per-sibling credit, gradients, and 60-step learning trajectories.

Reported welfare is the sum of the four sibling rewards. In the harmful-eviction setting, joint-target training reaches shared-deployment welfare 1.681470, compared with 1.621644 for isolated-target and PRSC training. In the beneficial-refresh setting, joint-target, isolated-target/PRSC, and group-centered training reach 2.954400, 2.956130, and 2.960326, respectively. These results distinguish restoring reference-state credit from optimizing shared-service welfare. The exact reference construction here is separate from the finite-mask estimator audit above.

B Receipt Schema and Audit Protocol

B.1 Receipt fields

The core receipt schema, `sri.receipt.v1`, records action identity, resource exposure, and execution outcomes. Its fields are listed in Table 6.

Table 6: Resource receipt schema

Field group	Contents
Schema and pairing	schema version, stable sample key, and call identifier
Identity	run, step, prompt, group, trajectory, and sibling identifiers
Action	tool name, semantic action hash, and resource namespace
Resource version	resource epoch, backend hash, and account hash
Scheduling	dispatch sequence, arrival, queue entry, service start, and service end times
Cache and quota	cache key hash, cache hit, cache age, quota before, and quota after
Task state	task-state hashes before and after the call
Outcome	timeout, retry count, status, observation hash, reward, and verifier details

Receipts are append-only JSON lines, with required strings and timestamps validated at write time. Stable hashes replace raw query text when pairing requires only equality. The original validation suite writes 2,624 receipts and reproduces its output hash under a fixed configuration.

B.2 Fixed-action paired replay

Fixed-action audits use the following protocol.

1. Sample policy actions once and assign a stable sample key to each sampled trajectory.
2. Execute every action with a trajectory-private resource namespace.
3. Replay the same actions under a declared shared namespace and dispatch order.
4. Pair receipts by stable sample key and call identifier, then validate complete pairing.
5. Compare status, observation hash, timeout, post-execution task state, reward, and group-relative credit.
6. For reported confidence intervals, cluster-bootstrap matched execution pairs at the prompt or initial-state level, retaining all siblings and both execution conditions together.

For multistep tasks, fixed-action replay holds the complete action sequence constant. Separate adaptive runs generate each next action from the observations produced under the current resource regime (Appendix E).

Validation cases with a shared one-entry cache or finite quota produce outcome POCR of 87.5% and 75.0%, respectively. Simultaneous cache misses without single-flight coordination, unrelated cache keys, and sufficient capacity produce zero POCR. Changing dispatch order while holding actions fixed changes every assigned cache reward in the order-sensitivity case.

B.3 Why reward change alone is insufficient

The receipt records exposure even if a scalar verifier is insensitive to it. In ToolSandbox, outcome POCR is 33.69% for Qwen3-1.7B, while reward POCR is 27.39%. For Qwen3-4B, outcome POCR is 87.5% because later tool actions encounter a different status, while reward POCR is zero. Receipt-based auditing thus captures changes in tool outcomes even when scalar rewards remain unchanged.

C Search Agent Details

C.1 Retriever and retrieval protocols

The local corpus contains 21,015,324 Wikipedia 2018 passages. In the initial retriever check on 128 HotpotQA development questions, top-10 BM25 retrieves at least one supporting title for 75.78% of

questions. Mean recall over the two supporting titles is 52.15%, and answer containment is 49.22%. Warm throughput rises from 19.6 queries per second with one worker to 270.3 with 16 workers and 293.2 with 32 workers. At 32 workers, mean and P95 latency are 82.5 and 209.2 ms. These measurements motivated the 75, 125, and 200 ms queue deadlines.

Table 7 distinguishes the retrieval conventions used across studies. The original 1.7B results below retain their original evaluator. Appendix I reports the revised evaluations and 160-update results, with training rewards and evaluation splits specified in Appendix J.

Table 7: Retrieval and scoring protocols. Passage rankings and rankings of unique document titles are distinguished explicitly.

Study	Retrieval	Scoring
Original 1.7B study and 4B transfer	10 passages	Original supporting-title NDCG
Protocol-v2 frozen audits and 40-update 4B/8B training	3 passages	Supporting titles scored at their first occurrence in the passage ranking
Revised evaluations, Qwen3-8B optimizer/replay study, and Llama training	20 candidates	Deduplicate normalized titles in retrieval order, then compute binary NDCG@3 on the first three unique documents

C.2 Frozen-policy protocol

The policy generates a search query from the question. Each query is sampled once and replayed across resource conditions. In the original Qwen audit, a deterministic fallback extracts queries when search tags are malformed, and retrieval returns ten passages. The answer model then receives the question, query, and passages. The later protocol-v2 audits use three retrieved passages. The Llama parser supports valid XML, recovered XML, and plain-text fallback, while unpaired malformed XML is retained as an unexecuted failure with zero retrieved documents. Failures remain in the evaluation denominator.

In the main quota intervention, the isolated condition retrieves for every valid query. The shared condition orders siblings by query token length and allows the first two valid queries. Remaining queries receive a quota-exhausted observation. Alternative order experiments use sibling identifier and two independently hashed permutations. The original 1.7B frozen set contains 128 prompts and 1,024 trajectories. The protocol-v2 audits use 256 prompts and 2,048 trajectories per sampling seed at Qwen3-4B, Qwen3-8B, and Llama-3.1-8B.

The random dispatch orders with seeds 6801 and 6802 yield F1-credit cosines of 0.073 and 0.163, respectively, relative to token-length order.

Table 8: Frozen quota dose response for Qwen3-1.7B with group size eight

Quota	Outcome POCR	EM Δ	F1 Δ	F1 credit cosine	Credit activation change
1	87.50%	-3.32	-2.10	0.179	19.53%
2	75.39%	-3.13	-1.94	0.109	19.53%
4	50.00%	-2.05	-1.16	0.255	17.97%
8	0.00%	0.00	0.00	1.000	0.00%

Paired outcome and task effects decrease as capacity increases and vanish at quota eight. Credit cosine varies non-monotonically across the smaller quotas.

C.3 Queue and deadline intervention

We submit fixed queries to the BM25 server through a capacity-two gate. The isolated condition gives each trajectory its own queue exposure. The shared condition queues siblings together. Table 9 reports the original Qwen3-1.7B results, and Table 10 reports the subsequent Qwen3-4B and Qwen3-8B audits.

At 75 ms, the Qwen3-1.7B outcome-POCR interval is $[9.96, 18.65]\%$. EM and F1 change by -0.68 and -0.04 points. The larger-model audits run on an otherwise idle host and average three fixed-

Table 9: Original Qwen3-1.7B BM25 queue audit. Rates are percentages of trajectories.

Deadline	Isolated timeout	Shared timeout	Outcome POCR
75 ms	2.44%	16.21%	14.26%
125 ms	0.98%	7.52%	6.84%
200 ms	0.00%	2.93%	2.93%

action sampling batches per model. Longer deadlines reduce timeouts and paired outcome changes while increasing credit alignment.

Table 10: Qwen3-4B and Qwen3-8B queue audits over three sampling batches. Timeout and POCR columns are percentages. ΔNDCG is Shared minus isolated in points. Credit cosine compares the paired group-credit vectors.

Model	Deadline (ms)	ISO timeout	Shared timeout	POCR	ΔNDCG	Credit cos.
Qwen3-4B	75	7.7	26.9	20.3	−7.48	.573
	125	1.0	13.2	12.5	−4.97	.711
	200	0.1	4.5	4.7	−1.95	.866
Qwen3-8B	75	5.6	27.9	23.1	−9.72	.474
	125	0.8	13.0	12.6	−4.57	.666
	200	0.3	4.2	4.2	−1.26	.862

C.4 Online optimization

Table 11 records the original 40-update settings. Optimization updates only generated query tokens. Subsequent training protocols and results are given in Appendices J and I.

Table 11: Original 40-update search RL configuration

Parameter	Value
Base models	Qwen3-1.7B and Qwen3-4B
Adapter	LoRA rank 8 on query and value projections
Group size	8
Prompts per update	4
Updates	40
Shared quota	2 calls per prompt group
Retrieval depth	10 passages
Training reward	supporting-title NDCG
Trainable span	generated query tokens
Main 1.7B seeds	6522, 6523, 6524
4B boundary seed	6851
Evaluation checkpoints	0, 10, 20, 30, 40 updates
Monitor prompts	first 64 HotpotQA development prompts
Final prompts	disjoint remaining 448 prompts

Every method begins from the same base weights and newly initialized adapter for a given seed. Conditions use the same prompt batches and model sampling seeds. Each run generates 1,280 training trajectories. Full reference-state replay reexecutes all quota-altered retrievals and produces a final adapter byte-identical to isolated training after 40 updates for seed 6522.

C.5 Aggregate 40-update results

Table 12 and Figure 5 summarize the original Qwen3-1.7B training study. RCC reaches comparable held-out NDCG without supplemental retrieval, with 2.08% of credit entries active. PRSC retains a broader credit signal and uses 63.83% as many retrieval executions as isolation.

Table 12: Original Qwen3-1.7B online training over three seeds on the 448-question held-out HotpotQA set, disjoint from the 64-question monitor, using the original evaluator. NDCG and active credit are percentages. Calls are relative to isolation. Intervals are hierarchical 95% bootstrap intervals for the NDCG delta.

Method	NDCG (%)	Δ (pt.)	95% CI	Calls (% ISO)	Active (%)
Shared	36.954	−2.794	$[-3.909, -1.665]$	25.00%	62.71%
RCC	39.726	−0.022	$[-0.701, +0.527]$	25.00%	2.08%
PRSC $p = .5$	39.628	−0.120	$[-0.502, +0.264]$	63.83%	15.81%
Isolated	39.748	0	n/a	100.00%	25.52%

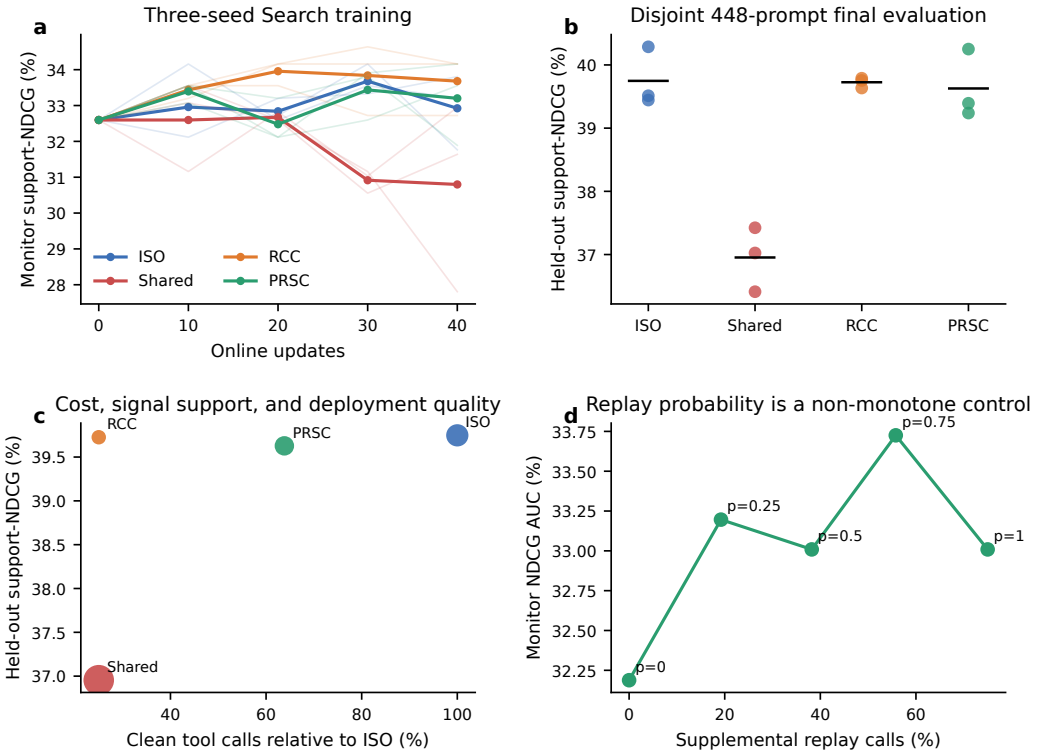


Figure 5: Qwen3-1.7B online training. Shared-resource training lowers held-out retrieval NDCG, while RCC and PRSC close the gap with different tradeoffs in credit support and tool calls.

C.6 Per-seed final results

PRSC restores 95.7% of the Shared–isolated gap under the original evaluator. Its difference from isolation has a seed-level 95% interval of $[-0.331, +0.090]$ points and a hierarchical interval of $[-0.502, +0.264]$ points.

D Gradient Audit

D.1 Gradient calculation and original Qwen3-1.7B audit

Within each paired audit, model parameters, inputs, and generated action tokens are fixed. We backpropagate the policy loss through the query and value LoRA adapters, changing only the resource-derived credit coefficients. Cosine compares the resulting raw gradients with the paired isolated gradient. Relative error is $\|g - g_{\text{ISO}}\|_2 / \|g_{\text{ISO}}\|_2$. Cosine is undefined if either vector is zero, and relative error is undefined if the reference vector is zero.

Table 13: Held-out HotpotQA retrieval NDCG by training seed

Seed	Isolated	Shared	RCC	PRSC $p = .5$
6522	40.286	37.025	39.790	40.251
6523	39.444	36.412	39.635	39.239
6524	39.513	37.424	39.754	39.394
Mean	39.748	36.954	39.726	39.628

The original Qwen3-1.7B audit uses 1,024 fixed queries and the sequence log-probability loss

$$\mathcal{L}_{\text{audit}} = -\sum_i \widehat{A}_i \log \pi_\theta(A_i \mid x_i), \qquad \log \pi_\theta(A_i \mid x_i) = \sum_{t=1}^{T_i} \log \pi_\theta(a_{it} \mid x_i, a_{i,<t}). \tag{23}$$

Here, T_i is the number of generated action tokens. Gradients cover approximately 1.6 million trainable LoRA parameters. Table 14 reports the original three adapter-initialization seeds.

Table 14: Original Qwen3-1.7B paired gradient audit. The Shared-only rerun is reported separately in Table 1.

Seed	Shared cosine	Shared relative error	PRSC cosine	PRSC negative modules
6515	-0.418	1.936	0.964	0%
6516	-0.425	1.931	0.943	0%
6517	-0.436	1.940	0.957	0%

Between 94.64% and 98.21% of effective modules have negative Shared cosine in this original audit. Every effective module has nonnegative cosine under PRSC. With sparse exact-match credit, the Shared cosine is -0.470.

D.2 Qwen3-8B training-checkpoint audit

The checkpoint audit follows training seeds 6861–6865 at updates 0, 10, 20, 30, and 40. It uses rank-8 query/value LoRA with scaling parameter 16 and zero dropout, covering 3,833,856 trainable parameters. The probe contains 64 HotpotQA prompts disjoint from training, calibration, monitor, and final evaluation, with eight sampled actions per prompt. At each checkpoint, one fixed action batch is replayed under the run condition, isolated execution, and full reference replay. Conditions use their own training checkpoints, with the paired comparisons evaluated at the same parameters within each checkpoint.

Table 15 gives the values underlying Figure 4. Each entry is the mean and sample standard deviation across five training seeds. PRSC has higher mean cosine and lower mean relative error than Shared and RCC at every checkpoint. ISO and full reference replay have cosine one and relative error zero throughout.

Averaging the five checkpoint means gives raw-gradient cosines of 0.2910 for Shared and 0.9821 for PRSC. In these diagnostic probes, PRSC uses 128 primary and 286.12 supplemental retrieval executions per checkpoint on average, totaling 414.12 compared with 512 for full reference replay. This is a 19.12% reduction in diagnostic executions. Full replay uses 384 supplemental executions. Optimizer-history comparisons are reported separately in Appendix I.

E Stateful Tool Environments

We study persistent settings in ToolSandbox and two multistep tasks involving inventory and caching. The multistep tasks use Qwen3-8B with sampling seed 9401 and 64 groups of eight siblings per task. Fixed-action replay executes complete sequences generated under isolation. Adaptive execution lets each agent choose subsequent actions from its tool observations. Shared execution uses round-robin dispatch in forward sibling order (0–7) or reverse order (7–0), with at most one call per active sibling per round and inactive slots retained. A trajectory-private resource control matches ISO in both branches. Group welfare sums the eight trajectory rewards, and failures remain in the denominator.

Table 15: Qwen3-8B raw-gradient alignment with each condition's paired isolated reference. Values are mean ± sample standard deviation across five training seeds.

Condition	Update	Cosine	Relative error
Shared	0	0.4370 ± 0.0721	0.9740 ± 0.0995
	10	0.1964 ± 0.5179	3.1518 ± 2.4172
	20	0.3451 ± 0.3517	2.1808 ± 1.4669
	30	0.0923 ± 0.2348	2.3189 ± 1.2322
	40	0.3841 ± 0.2724	1.9918 ± 0.8277
RCC	0	0.0277 ± 0.4897	0.9412 ± 0.1538
	10	0.5609 ± 0.6203	0.7732 ± 0.1967
	20	0.2542 ± 0.5380	0.9169 ± 0.1423
	30	0.3786 ± 0.4034	0.9302 ± 0.1617
	40	0.1059 ± 0.2734	0.9900 ± 0.0309
PRSC	0	0.9957 ± 0.0018	0.1271 ± 0.0434
	10	0.9710 ± 0.0489	0.1924 ± 0.1745
	20	0.9911 ± 0.0100	0.1486 ± 0.0615
	30	0.9714 ± 0.0326	0.2005 ± 0.1491
	40	0.9816 ± 0.0213	0.2432 ± 0.0865

In the fixed-action branch, PRSC selects state-exposed trajectories with probability 0.5, replays their sequences from reference states, and forms propensity-corrected credit. Across both tasks and schedules, 960 of 1,920 exposed trajectories receive replay, adding 2,353 tool calls to 5,120 primary calls. The replay corrects credit while retaining the realized shared outcomes. Adaptive runs compare ISO, private, shared, and task-specific resource controls.

E.1 TOOLSANDBOX SETTING TASKS

We use the official `cellular_off` and `wifi_off` setting scenarios. The initial world state has the relevant setting enabled. ACT invokes the official disabling tool. NOOP leaves the setting unchanged. The official milestone evaluator scores the final setting. The audits use 256 groups (2,048 trajectories) for Qwen3-1.7B and 64 groups for Qwen3-4B, with eight siblings per group.

In isolated execution, every sibling starts from the same snapshot. In shared execution, sibling conversations remain private while the external setting database persists in dispatch order. This design isolates tool-side state from context sharing.

For Qwen3-1.7B, ACT frequency is 13.96%. The cellular scenario has reward POCR 12.99% and reward increase 6.49 points. The Wi-Fi scenario has reward POCR 41.80% and reward increase 20.90 points. Across scenarios, reward POCR is 27.39% with interval $[23.83, 31.01]$. Mean reward rises 13.70 points with interval $[11.91, 15.50]$.

The first ACT in every shared group succeeds. Later ACT calls fail with the expected already-disabled status in every case. NOOP siblings after a prior ACT inherit the successful final state and receive reward without taking the action. This trace pattern localizes the effect to external state persistence.

Partial reference replay in the free-riding setting reaches a mean-estimator credit cosine of 0.996 relative to isolated credit.

Qwen3-4B emits ACT for every trajectory. Seven of eight shared siblings encounter the repeated-action error. Yet isolated and shared final rewards are both 100%. The credit vector is identically zero on both sides, so its cosine is undefined.

State injection modifies only non-empty data rows, preserving ToolSandbox's schema sentinel. All reported results use this corrected implementation.

E.2 INVENTORY RESERVATION

Each trajectory must reserve and commit two units of the specified item under its own reservation identity. The primary and backup warehouses initially contain six units each. The tools expose current

stock, atomically reserve available units, commit an owned live reservation, and release an owned reservation. A successful commitment earns reward 1 and failure earns 0. The sufficient-capacity control raises primary stock to 16 units.

Isolated sequences reserve and commit at the primary warehouse using two calls. Shared replay allows three siblings per group to succeed, reducing success from 100% to 37.5% and welfare from 8 to 3. Adaptive agents can use the backup warehouse after rejection, raising shared success to 75% and welfare to 6. This attains the six-reservation limit imposed by 12 available units. Both branches reach 100% success and welfare 8 under the sufficient-capacity control, which also removes the order effect.

Reversing dispatch preserves group welfare but changes 75% of sibling rewards under fixed actions and 50% under adaptive execution. The corresponding credit cosines are -0.6000 and -0.3333. Fixed-action PRSC reduces the RMS difference between estimated and paired ISO credit from 0.999793 to zero. ISO rewards are uniformly 1, so the reference credit is zero and its cosine is undefined. The adaptive reverse condition retains one task-constraint failure among its 512 trajectories.

E.3 Cache refresh

The cache starts cold at version 1, while the origin holds version 2. A fetch returns the current cache state without changing it. A refresh accesses the origin when the cache is cold or stale and otherwise returns a cache hit. Success requires the trajectory's own fetch observation to contain fresh version 2. Reward is the success indicator minus 0.1 for each external origin call. The resource control starts with a prewarmed version-2 cache.

Isolated trajectories fetch, refresh, and fetch again. Under shared round-robin execution, all initial fetches precede the first refresh, so adaptive agents retain the same three-action sequences as ISO. Only the first refresh in each group accesses the origin. Both branches retain 100% success while reducing origin calls from eight to one per group, or from 512 to 64 per condition. Mean reward rises from 0.9 to 0.9875 and welfare from 7.2 to 7.9. The prewarmed control removes origin calls and the order effect, yielding reward 1 and welfare 8. Fixed replay still executes all three actions, while adaptive trajectories finish after the first fresh fetch.

Reversing dispatch changes which sibling pays the origin cost. In both branches, 25% of sibling rewards change and the forward/reverse credit cosine is -0.1429, while success and group welfare remain unchanged. Fixed-action PRSC reduces the RMS difference between estimated and paired ISO credit from 0.996985 to 1.35×10^{-12} in forward order and 1.15×10^{-12} in reverse order. ISO rewards are uniformly 0.9, giving zero reference credit. RMS difference measures correction toward this reference, while forward/reverse cosine measures order sensitivity between the two nonzero shared credit vectors.

F Method Ablations

F.1 Replay probability

On frozen Qwen3-4B trajectories, we evaluate replay probabilities $p = .10, .25, .50, .75$ using 2,000 independent replay masks. Supplemental retrieval executions amount to 7.50%, 18.78%, 37.48%, and 56.21% of all trajectories, respectively. The corresponding mean-estimator credit cosines with isolated credit are 0.885, 0.960, 0.988, and 0.998. These cosines compare the credit vector averaged over masks with the isolated vector. Single-mask alignment is lower in aggregate.

A separate online sweep uses Qwen3-1.7B and training seed 6522 at $p = .25, .50, .75$. Table 16 reports the original retrieval NDCG on the 448-question evaluation set, disjoint from the monitor. The $p = .5$ run finishes closest to isolation, while $p = .75$ uses more replay and has a larger performance gap. The original three-seed Qwen3-1.7B study also uses $p = .5$ (Table 12).

The later Qwen3-4B and Qwen3-8B 40-update studies use model-specific gradient calibration on 128 prompts. For each model, the selector chooses the smallest $p \in \{.10, .25, .50, .75, 1\}$ with mean ESS/$G \geq .5$ and a 95% lower confidence bound on raw-gradient cosine of at least .9. The bound is computed by exact enumeration of ordered bootstrap resamples of three adapter-and-replay seeds.

Table 16: Original Qwen3-1.7B replay-probability sweep, seed 6522. NDCG differences are in points on the original 448-question evaluation. Supplemental executions are percentages of all training trajectories. Recovery is the fraction of the Shared-to-ISO NDCG gap closed.

Condition	Supplemental executions	NDCG Δ vs. ISO	Gap recovered
Shared $p = 0$	0.00%	-3.262	0.0%
PRSC $p = .25$	19.22%	-1.098	66.3%
PRSC $p = .50$	38.20%	-0.036	98.9%
PRSC $p = .75$	55.78%	-0.551	83.1%
Full replay	75.00%	0.000	100.0%

Both models select $p = .75$. At this probability, mean ESS/G and the cosine lower bound are .800 and .930 for 4B, and .806 and .962 for 8B. At $p = .5$, both models pass the ESS criterion but fall below the cosine threshold. This selection uses only calibration gradients. The separate 160-update training protocols and Llama calibration are reported in Appendix I.

F.2 Receipt noise

The original Qwen3-1.7B audit perturbs exposure labels at replay probability $p = .5$ and evaluates 2,000 replay masks. A false negative labels an exposed trajectory as clean, admitting its altered reward without reference replay. A false positive labels a clean trajectory as exposed, adding it to the replay candidates. Table 17 distinguishes the cosine for individual replay masks from the cosine of the credit vector averaged over masks. All comparisons use paired isolated credit.

Table 17: Original Qwen3-1.7B receipt-noise audit at $p = .5$. Single-mask cosine summarizes individual replay draws, while mean-estimator cosine compares the averaged credit vector with isolated credit. Supplemental executions are percentages of all trajectories.

Noise	Single-mask cosine	Mean-estimator cosine	Supplemental executions
Clean	0.633	0.811	37.56%
5% false negative	0.526	0.803	35.60%
10% false negative	0.462	0.788	33.74%
20% false negative	0.380	0.731	30.00%
10% false positive	0.634	0.838	38.80%
20% false positive	0.634	0.864	40.02%
10% symmetric	0.463	0.818	35.02%
20% symmetric	0.376	0.780	32.54%

False negatives reduce both replay coverage and alignment in this audit. False positives increase replay executions while preserving single-mask alignment and improving mean-estimator alignment.

F.3 Extended receipt and replay stress audit

The extended audit crosses two models, Qwen3-4B and Qwen3-8B, two reward metrics, retrieval-support NDCG and token F1, 29 stress scenarios, and two replay designs. Independent Bernoulli replay samples trajectories independently, while batch-correlated replay introduces dependence between their replay decisions. The resulting 232 cells each use 2,000 replay draws. Logged quota receipts identify 1,536 exposed and 512 unexposed trajectories for each model, with no false positives or false negatives. The stress conditions then inject exposure-label errors, stale resource epochs, or errors in the observation propensities supplied to the estimator.

Table 18 reports credit based on the audit's original retrieval-support NDCG rewards. Mean-estimator cosine measures the direction of the averaged credit vector, while single-draw relative MSE measures the error of individual replay estimates. False negatives reduce alignment much more than false positives. For 4B and 8B, the corresponding relative biases are 0.196 and 0.173 with correct receipts, 0.743 and 0.882 with 40% false negatives, and 0.203 and 0.182 with 40% false positives.

Table 18: Extended receipt and replay stress on NDCG-based credit. Panel A reports the cosine of mean credit under independent Bernoulli replay. Panel B reports single-draw relative MSE with correct receipts. All receipt errors are injected. Each cell uses 2,000 replay draws.

Condition	Qwen3-4B	Qwen3-8B
A. Mean-estimator credit cosine		
Correct receipts	0.987	0.991
40% false negatives	0.674	0.531
40% false positives	0.989	0.993
40% stale resource epochs	0.831	0.779
B. Single-draw relative mean squared error		
Independent replay	0.646	0.701
Batch-correlated replay	0.759	0.794

With correct receipts, batch-correlated replay increases relative MSE while retaining high mean-estimator alignment. Stale resource epochs reduce alignment in both models. Underestimating observation propensities by 25% increases MSE, while overestimating them by 25% reduces single-draw variance and shifts estimator norm and bias. These comparisons separate receipt classification from the replay sampling and weighting used to form credit.

The Qwen3-8B token-F1 population for seed 6881 has zero isolated credit. Its 29 scenarios under two replay designs account for 58 cells with undefined cosine and relative error, leaving 174 evaluable cells. An independent Monte Carlo rerun preserves all 232 cell identities and their evaluability flags.

F.4 Group size

In the original search audit with quota fixed at two, outcome POCR is 0%, 50%, 75%, and 87.5% for $G = 1, 4, 8, 16$. F1-credit cosine is undefined at $G = 1$, where group-relative credit is zero. For $G = 4, 8, 16$, credit cosine is 0.236, 0.113, and 0.105, and active sign disagreement is 6.25%, 10.00%, and 13.54%, respectively. The runs at different group sizes also have different sampled query distributions and sparse answer rewards. Task-score loss is nonmonotone across these runs despite the increasing exposure rate.

G System Cost and Signal Support

G.1 Retrieval executions and credit support

We count primary retrieval executions and supplemental reference executions separately. Their sum gives total retrieval cost. Quota refusals do not execute retrieval. Relative cost divides this total by the executions used by isolation in the same study. Development, evaluation, and gradient-probe executions are accounted for separately from training.

Table 19 reports the original Qwen3-1.7B 40-update study over seeds 6522–6524. Each run generates 1,280 trajectories, so isolation executes 1,280 retrievals. Active credit is the fraction of trajectory credit entries that are nonzero.

Table 19: Original Qwen3-1.7B training cost and signal support, averaged over three seeds. Total and supplemental executions are percentages of isolated executions. Active credit is a percentage of trajectories.

Method	Total (% ISO)	Supplemental (% ISO)	Active (%)	Time (s)
Shared	25.00	0.00	62.71	249
RCC	25.00	0.00	2.08	261
PRSC $p = .5$	63.83	38.83	15.81	266
Isolated	100.00	0.00	25.52	238

Quota exhaustion creates reward variation and raises the active-credit fraction under Shared execution. RCC removes affected labels, leaving nonzero credit on only 2.08% of trajectories. Partial replay

increases this fraction to 15.81% by supplying reference rewards. Active credit measures signal availability, while paired credit and gradient comparisons measure its agreement with isolation.

Mean wall time is 238 seconds for ISO and 266 seconds for PRSC despite the latter's lower retrieval count. Local model generation dominates runtime and retrieval runs asynchronously, so the reduction in executions does not translate into a measured training speedup here.

G.2 Longer-run replay budgets

The 160-update studies in Table 20 generate 5,120 trajectories per run. Shared execution performs 1,280 primary retrievals, while isolation performs 5,120. The table lists their realized PRSC budgets. In the Qwen3-8B equal-budget comparison, Naive and Random each match PRSC's supplemental budget within each seed, averaging 1,908.33 executions. In the Llama comparison, Naive and Random match PRSC's budget within each seed, averaging 1,904.67 supplemental executions. The corresponding performance results appear in Appendix I.

Table 20: PRSC training retrieval budgets after 160 updates. Each row uses 1,280 primary executions and compares total cost with 5,120 isolated executions. Savings are $1 - \mathrm{Total}/5{,}120$. The last two rows report means over three training seeds.

Study	p	Supplemental	Total	Savings
Qwen3-4B/8B, $\beta_1 = 0$ recovery	.75	2,874	4,154	18.87%
Qwen3-8B, replay comparison	.50	1,908.33	3,188.33	37.73%
Llama-3.1-8B, matched training	.50	1,904.67	3,184.67	37.80%

For Llama, PRSC averages 7,024.67 requests per run across three training seeds, comprising 3,184.67 retrieval executions and 3,840 quota refusals. The reported 37.80% saving uses execution counts. The checkpoint-probe saving of 19.12% in Appendix D instead compares 414.12 diagnostic executions with 512 for full replay. Multistep tool-call costs are reported with their task definitions in Appendix E.

H Model and Dataset Transfer

H.1 Original Qwen3-4B configuration transfer

The original Qwen3-4B transfer study applies the 1.7B configuration with PRSC $p = .5$. Its four conditions share initial weights, seed 6851, prompt order, and 40 updates. Table 21 reports scores on the 448-question held-out set under the original evaluator. The later calibrated five-seed 4B study and its unique-title reevaluation are reported separately in Appendix I.

Table 21: Original Qwen3-4B configuration transfer after 40 updates with seed 6851. Scores use the original evaluator and are on a 0–100 scale.

Method	NDCG	Answer hit	Token F1	Exact match
Isolated	37.888	40.625	15.330	9.598
Shared	37.316	39.732	15.689	10.268
RCC	36.603	39.063	14.463	8.482
PRSC $p = .5$	36.418	39.063	14.419	8.482

The Shared, RCC, and PRSC NDCG intervals relative to isolation all include zero. Under this evaluator, RCC and PRSC exact match are each 1.116 points lower, with interval $[-2.232, -0.223]$. Many RCC updates have almost no active credit. The direct configuration transfer yields lower NDCG point estimates for RCC and PRSC than for isolation.

H.2 2WikiMultihopQA

We evaluate the original Qwen3-1.7B runs from seeds 6522–6524 on 448 2WikiMultihopQA questions using their original evaluator. This dataset is not used for training or checkpoint selection. Table 22 reports mean differences from the isolated adapter with the same training seed.

Table 22: Original Qwen3-1.7B evaluation on 2WikiMultihopQA over three training seeds and 448 prompts. Entries are differences from isolation in points under the original evaluator.

Method	NDCG Δ	Token F1 Δ	Exact match Δ
Shared	+0.038	+1.076	+1.339
RCC	+0.075	−0.529	−0.223
PRSC $p = .5$	+0.266	−0.029	+0.149

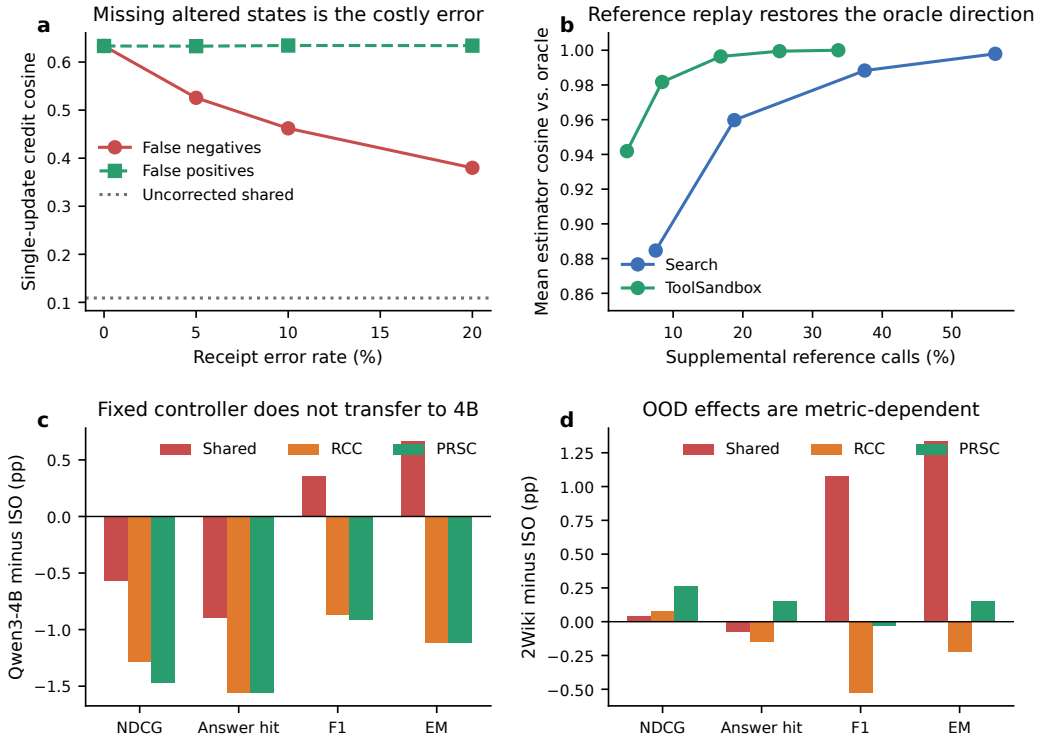


Figure 6: Original receipt-noise, reference-replay, and transfer studies. False-negative receipts reduce PRSC credit alignment more than false positives. Reference replay restores credit alignment in search and ToolSandbox. The Qwen3-4B configuration transfer and Qwen3-1.7B 2WikiMultihopQA panels retain their original evaluators.

All NDCG intervals include zero. Shared exact match increases by 1.339 points, with a seed-level 95% interval of $[0.785, 1.894]$ and a hierarchical interval of $[0.223, 2.455]$ under the original evaluator. Retrieval and answer metrics thus show different transfer effects. Figure 6 summarizes these original evaluations alongside the earlier receipt-noise and configuration-transfer studies.

I Additional Training Results

Retrieval evaluations use isolated execution and binary supporting-title NDCG@3 over the first three unique normalized titles among 20 BM25 candidates. Training rewards are specified by study. Scores and normalized AUC use the 0–1 scale, while differences and confidence intervals are in points.

Matched 160-update comparisons use rank-8 query/value LoRA, a common fresh adapter and empty optimizer states, and four prompts with eight siblings per update. The main 8B training comparisons use three seeds (Tables 26 and 27), while the optimizer ablations retain their original seed. Three-seed summaries report means and sample standard deviations. Shared execution serves two queries per group. Monitor AUC uses trapezoidal integration over updates 0, 20, 40, 80, and 160 divided by 160. Costs count training retrieval executions, excluding refusals, development, evaluation, and probes.

I.1 REEVALUATION OF THE 40-UPDATE MODELS

We reevaluate Qwen3-4B and Qwen3-8B adapters from seeds 6851–6855 and 6861–6865 on HotpotQA development rows 64–511, a historical 448-question set already used for analysis. Training used the original depth-three passage-ranking reward and PRSC $p = .75$. Reevaluation applies the unique-title metric and corrected answer prompt to the initial models and ISO, Shared, and PRSC endpoints.

Table 23 reports paired-t intervals over five training seeds and secondary nested-bootstrap intervals from 5,000 resamples of paired seeds and prompts.

Table 23: Forty-update reevaluation. Δ is relative to ISO. Both intervals are 95% intervals in points.

Model	Method	Step 0	Step 40	Δ	Seed t interval	Nested interval
Qwen3-4B	ISO	.384518	.413592	0	—	—
	Shared	.384518	.411252	$-.2340$	$[-1.8362, +1.3682]$	$[-1.4593, +1.0612]$
	PRSC	.384518	.408580	$-.5012$	$[-.8819, -.1204]$	$[-1.0118, +.0326]$
Qwen3-8B	ISO	.424746	.428490	0	—	—
	Shared	.424746	.433912	$+.5422$	$[-.1773, +1.2617]$	$[-.2408, +1.3573]$
	PRSC	.424746	.427526	$-.0965$	$[-.5524, +.3595]$	$[-.5155, +.3647]$

ISO improves by 2.9073 points at 4B and 0.3744 at 8B. Both Shared-minus-ISO seed intervals include zero. All five 4B PRSC-minus-ISO differences are negative, with the paired-t interval below zero and the nested interval extending slightly above it.

I.2 FIRST-MOMENT ABLATION IN FRESH TRAINING

The seed-8701 study trains Qwen3-4B and Qwen3-8B with AdamW $\beta_1 = .9$ or 0, learning rate 2×10^{-5}, $\beta_2 = .95$, $\epsilon = 10^{-8}$, and zero weight decay. Training uses the original depth-three reward. Table 24 evaluates the resulting policies on historical448 with unique-title NDCG@3. The PRSC extension uses $\beta_1 = 0$ and $p = .75$, with the same initialization and prompt schedules as its ISO and Shared controls.

Table 24: Seed-8701 bridge448 results at update 160. PRSC was run only with $\beta_1 = 0$. $\Delta_{\mathrm{S-I}}$ is Shared minus ISO in points.

Model	β_1	Step 0	ISO	Shared	PRSC	$\Delta_{\mathrm{S-I}}$
Qwen3-4B	.9	.388950	.435108	.289350	—	-14.5758
	0	.388950	.416062	.326515	.425886	-8.9547
Qwen3-8B	.9	.426620	.439687	.329936	—	-10.9750
	0	.426620	.441235	.303216	.455964	-13.8019

Setting $\beta_1 = 0$ changes Shared-minus-ISO by $+5.6211$ points at 4B and -2.8269 at 8B. PRSC improves over its matched Shared condition by 9.9371 and 15.2748 points. Each recovery run uses 1,280 primary and 2,874 supplemental retrieval executions, totaling 4,154 compared with 5,120 for ISO, a reduction of 18.87%. Shared uses 1,280 executions. On monitor64, PRSC finishes 1.0835 points below ISO at 4B and 2.1670 points above it at 8B.

I.3 FIXED-STATE OPTIMIZER PROBES

At fixed Qwen3-8B Shared checkpoints from seeds 6861–6865 and updates 10, 20, 30, and 40, paired Shared/ISO raw-gradient cosine averages 0.2545. Applying both gradients to cloned copies of the saved AdamW state gives one-step update cosine 0.9891. Setting $\beta_1 = 0$ while retaining second-moment history lowers it to 0.2224. Parameters and actions are fixed. The probes use learning rate 2×10^{-5}, zero weight decay, and inactive norm clipping at 1. Update zero is excluded because its history is empty.

I.4 QWEN OPTIMIZER AND EQUAL-BUDGET REPLAY COMPARISONS

The seed-6861 Qwen3-8B study uses unique-title NDCG@3 for both training reward and evaluation. Three 40-update ISO development candidates per optimizer select learning rates 6×10^{-5} for AdamW, 1.123×10^{-4} for AdamW with $\beta_1 = 0$, and 0.2225506 for momentum-free SGD, shared across execution conditions. Fixed-action development selects PRSC $p = .5$. Table 25 reports monitor64 endpoints and AUC from initial score .384926. There is no separate final evaluation.

Table 25: Seed-6861 Qwen3-8B monitor64 results. Executions count training retrieval only.

Optimizer	Method	Step 160	AUC	Executions
AdamW	ISO	.457006	.418057	5,120
	Shared	.322653	.349940	1,280
	PRSC	.412640	.410359	3,186
AdamW, $\beta_1 = 0$	ISO	.382644	.402326	5,120
	Shared	.229130	.311105	1,280
	PRSC	.425756	.386323	3,186
SGD	ISO	.422220	.417081	5,120
	Shared	.280568	.313757	1,280
	PRSC	.441381	.408705	3,186

Each PRSC run in this optimizer study uses 1,280 primary plus 1,906 supplemental executions, 37.77% fewer than ISO.

PRSC AUC remains below ISO under all three optimizers. The SGD-minus-AdamW difference in the Shared penalty is -0.7299 endpoint points.

Table 26 reports the three-seed training comparisons. Naive samples resource-altered valid trajectories and Random samples all valid trajectories. Both use unweighted reference-reward credit and match PRSC's supplemental budget in all 640 groups per seed.

Table 26: Qwen3-8B results at update 160 with standard AdamW. NDCG@3 means and sample standard deviations use three training seeds. In the replay comparison, mean monitor AUC is .416955 for PRSC and .413415 for Naive.

Method	Mean	SD
A. Shared-execution effects: historical448, seeds 8701–8703		
ISO	.440680	.001665
Shared	.333309	.007523
B. Replay comparison: monitor64, seeds 6861–6863		
ISO	.444993	.010405
Shared	.267946	.073222
PRSC	.423817	.015398
Naive	.439366	.014485
Random	.421802	.030854

I.5 LLAMA MATCHED TRAINING AND GRADIENT PROBES

The five matched Llama-3.1-8B-Instruct conditions use seeds 19601–19603, AdamW $(\beta_1, \beta_2) = (.9, .95)$, learning rate 6×10^{-5}, and PRSC $p = .5$, selected on development seed 19501. Training uses the unique-title reward. Independent final448 is disjoint from training, development, monitor, probe, and the earlier frozen audit. One initial evaluation is shared across conditions. Their step-160 final evaluations follow completion of training. Table 27 includes all five conditions.

Shared finishes 35.6254 points below ISO and 9.9263 below initialization. PRSC improves over Shared by 35.6933 points and exceeds ISO by .0679 points. It exceeds Naive by .6952 points on final448, while Naive has the lower monitor endpoint and AUC. Random reaches 47.99 points, approximately 1.87 below ISO and 1.94 below PRSC.

Table 27: Llama-3.1-8B evaluations. The five matched final scores use update 160. Initial and final scores use final448. Monitor endpoint and AUC use monitor64. Results are means over three training seeds, with sample standard deviations shown for final scores.

Method	Initial	Final	Monitor	AUC	Executions
ISO	.241627	$.498618 \pm .001440$	.422145	.393750	5,120
Shared	.241627	$.142364 \pm .093280$	.124088	.183956	1,280
PRSC	.241627	$.499298 \pm .007305$	.419027	.383467	3,184.67
Naive	.241627	$.492346 \pm .007264$	.404581	.382081	3,184.67
Random	.241627	$.479941 \pm .008763$	.420966	.380682	3,184.67

PRSC, Naive, and Random each use 1,280 primary plus 1,904.67 supplemental executions on average, totaling 3,184.67. Their budgets match in all 640 groups per seed, saving 37.80% relative to ISO.

For seed 19601, at initialization, paired raw-gradient cosine is .337443 for Shared, .922318 for PRSC, and .896178 for Naive. All comparisons use the same fixed action batch, with PRSC and Naive sharing the replay mask and reference rewards. At the Shared step-160 checkpoint, the isolated reference gradient is zero, so all cosines relative to it are undefined.

J Reproducibility Manifest

Table 28 records the training and multistep study identities. Appendix I gives the optimizer settings, replay probabilities, retrieval definitions, and complete training results. Table 11 retains the original 1.7B settings and the initial 4B transfer configuration.

Table 28: Study identities and evaluation roles. Seed ranges identify separate training runs. The stateful study uses a sampling seed and evaluates execution outcomes without policy updates.

Study	Seed(s)	Updates	Evaluation
Original Qwen3-1.7B training	6522–6524	40	HotpotQA448 and 2Wiki448, original evaluator
Calibrated Qwen3-4B training	6851–6855	40	Historical bridge448, revised endpoint evaluation
Calibrated Qwen3-8B training	6861–6865	40	Historical bridge448, revised endpoint evaluation
Qwen3-4B/8B optimizer-history study	8701	160	Monitor64 and historical bridge448
Qwen3-8B optimizer and replay comparisons	6861	160	Monitor64
Qwen3-8B ISO/Shared comparison	8701–8703	160	Monitor64 and historical bridge448
Qwen3-8B replay comparison	6861–6863	160	Monitor64
Llama-3.1-8B five conditions	19601–19603	160	Monitor64 and independent final448
Qwen3-8B inventory and cache	9401	—	64 groups of eight siblings per task

The five-seed 40-update Qwen studies compare ISO, Shared, RCC, PRSC, and Full. Model-specific calibration selects PRSC $p = .75$ on independent 128-prompt splits. The later unique-title reevaluation covers ISO, Shared, PRSC, and the initial models, while retaining the original depth-three training reward. The initial 4B transfer run with seed 6851 uses its original, separately specified configuration. These experiments and the subsequent seed-8701 runs share a historical bridge whose results had already been analyzed.

The seed-6861 optimizer study selects learning rates from three 40-update ISO candidates per optimizer on development128 and selects replay probability through fixed-action development probes. Llama uses development seed 19501 for its three 40-update learning-rate candidates and replay calibration. Its five training conditions are ISO, Shared, PRSC, Naive, and Random. Table 27 reports their three-seed results. Additional seeds retain the original study-specific hyperparameters, with the

additional Qwen3-8B ISO/Shared training runs executed sequentially on one GPU in place of the original cross-GPU layout.

Audit seeds and sampling. The original exact-audit seed schedule begins at 6510. Frozen search uses eight siblings per prompt, with 128 prompts at Qwen3-1.7B and 256 prompts at Qwen3-4B, Qwen3-8B, and Llama-3.1-8B. The larger models use action-sampling seeds 6881–6883, producing 2,048 trajectories per seed. Original frozen-gradient comparisons use three adapter initializations per model. Qwen3-1.7B uses initialization seeds 6515–6517, and Llama uses 6931–6933 with actions from sampling seed 6881. These schedules describe the original audits. The later Shared-only rerun in Table 1 is distinct from these original audits. ToolSandbox uses seeds 6701 and 6702, dispatch-permutation audits use 6801 and 6802, and the multistep protocol is specified in Appendix E.

Evaluation and statistical units. Checkpoint evaluation uses isolated, normally operating retrieval. The 40-update HotpotQA endpoint contains 448 development questions disjoint from monitor64. The seed-8701 bridge is evaluated at updates 0, 40, and 160. The 160-update monitor grid is 0, 20, 40, 80, and 160, with normalized trapezoidal AUC as a secondary measure. Llama final448 is disjoint from training, development, monitor, gradient probes, and the earlier frozen audit. A common initial evaluation is shared across the five conditions, and their endpoint evaluations follow completion of training. Answer containment, token F1, and exact match supplement retrieval scores where reported.

Prompt bootstrap resampling preserves condition pairing. Earlier multi-seed training reports use seed-level intervals and, where specified, hierarchical resampling of seeds and paired prompts. The five-seed historical-bridge reevaluation reports paired-t intervals and a secondary nested bootstrap. Three-seed 160-update summaries use means and sample standard deviations across training seeds, with differences paired within each seed. Single-seed training results are point estimates. Failed calls, timeouts, and invalid trajectories remain in the corresponding audit and evaluation denominators.