AUDITING UPDATE-ORDER EFFECTS IN MULTI-AGENT REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Update order can change a multi-agent reinforcement learning (MARL) algorithm even when its objective, data, and optimizer are fixed. We audit the complete learning state, including optimizer memory, to distinguish three claims: updates commute, policies differ, and task utilities are statistically equivalent. The audit groups orders that are equivalent under commuting swaps and compares pairs of training trajectories after correcting for evaluation noise, retaining unresolved cases under explicit coverage assumptions. In three controlled PettingZoo/MPE environments, simplified independent-critic actor–critic updates have zero observed protocol range, whereas sequential shared-critic updates have ranges of 5.11–5.95 return points and two same-sample descriptive ranking reversals. A critic-only probe can miss actor differences: actors read different intermediate states even when the final critic states agree. Freezing shared state restores invariance under disjoint actor updates. In HARL-based experiments at 10M steps per branch, reversing the native loop also reassigns random streams; agent-local streams isolate order and yield heterogeneous sensitivity estimates. Projecting finite update differences onto a surrogate utility gradient ranks held-out surrogate sensitivity within the controlled MPE family (mean within-seed Spearman 0.78); this is not an environment-return predictor, and return-based scheduling and detector transfer remain unestablished. The resulting audit supports reproducible comparisons under the stated runtime protocols and small-seed sampling units. It does not establish a return-improving optimizer or a population-level ranking of update orders.

Keywords: multi-agent reinforcement learning, update order, protocol bias, optimizer state, reproducible evaluation

1 INTRODUCTION

In a sequential MARL loop, an actor may read a shared critic that another update has already changed. The objective and optimizer therefore do not fully specify the algorithm: the order of updates determines which intermediate states each actor reads. Heterogeneous-agent methods make sequential updates explicit (7), but the same issue arises in simpler actor–critic loops. Two updates commute when swapping their order leaves their combined effect unchanged. When they do not commute, order can change the finite-budget algorithm even under identical data and per-block random inputs.

We call the choice of update order an *update protocol*. Measuring its effect requires two distinctions. A native loop can draw from a shared random-number generator (RNG), so reversing the loop changes both order and the random inputs assigned to each agent. Moreover, equal final critic states can hide different actor updates. The audit must therefore specify the complete learning state, including optimizer memory, and the observable being compared. State equality, policy equality, and equality of expected return answer different questions.

We develop an audit that links these questions without treating them as equivalent. A valid commutation relation groups structurally equivalent orders; a noise-corrected comparison of paired trajectories assesses task utility and allows an unresolved outcome. Controlled actor–critic experiments separate invariant from coupled updates and expose the critic-endpoint blind spot. Experiments

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.

based on the Heterogeneous-Agent Reinforcement Learning (HARL) implementation assign random streams by agent to distinguish execution order from broader runtime protocols at 10M steps per branch. The structural grouping specializes a classical correspondence, and the smooth commutator expansion covers a restricted small-step case. Together, the analysis and experiments explain what an order audit can establish; neither sensitivity detection nor restored invariance alone establishes a return-improving schedule.

2 Related Work

Centralized actor–critic and PPO-based MARL provide baselines (5; 6); HARL makes sequential updates explicit (7). Our I-AC/S-AC loops instead isolate shared-critic reads. Classical composition theory motivates the commutator and partial-commutation quotient (2; 3); implementation sensitivity and few-run inference motivate the empirical audit (1; 4). Related MARL work studies credit assignment, value factorization, independent learning, and non-stationarity (8–13). Game-theoretic frameworks, regularized decision processes, and policy optimization address different evaluation and update objects (14–18). Our question is narrower: which orders remain distinguishable when learning state, data, random streams, read schedule, and observable are fixed? We use standard Markov-process and learning-theoretic background (19; 20).

3 Update Protocols and Protocol Bias

Definition 1 (Update block)**.** *Let $z = (\theta, h)$ be the complete learning state: parameters θ and memory h, including optimizer moments, step counters, and any mutable state used by later updates. Conditional on specified data and random inputs, an update block $U_i : \mathcal{Z} \to \mathcal{Z}$ is a deterministic training operator. Its write set may overlap another block's write set; an agent update can modify both its actor and a shared critic.*

Definition 2 (Update protocol)**.** *An update protocol is an ordered sequence or distribution over ordered sequences of update blocks. A deterministic protocol is $\pi = (i_1, \dots, i_m)$ and induces the composed update*

$$U_\pi = U_{i_m} \circ \cdots \circ U_{i_1}.$$

A randomized protocol samples $\pi \sim q$ at each training round.

We write $\theta_t^{\pi,s}$ for the parameter projection of the complete state $z_t^{\pi,s}$. The complete state determines future updates, whereas an evaluation functional measures a selected consequence of those updates. A parameter-based evaluation $J(\theta)$ can hide differences in optimizer memory that affect later training. Likewise, equal policy outputs on a fixed collection of observations do not imply equal policies everywhere. We thus distinguish complete-state invariance, agreement on an observation bank, and equality of expected return before defining utility-based sensitivity.

Execution order versus runtime protocol. An execution-order intervention is isolated only if every update block receives the same data and random stream under every tested permutation. We therefore write a runtime protocol as (π, ρ, β), where ρ maps random draws to named blocks and β records backend semantics. With one global random stream, reversing a loop changes both π and ρ because successive random-permutation draws are reassigned to different agents. Our agent-local intervention holds ρ fixed by agent and changes only π; native runs measure the broader implementation protocol, not execution order alone.

Fixed-batch probes match the actual data across orders. For on-policy trajectories, matching random inputs does not keep rollout data identical after policies diverge: data collection must be included in the complete round map, with matched exogenous random inputs. The accumulation relation below otherwise applies only to a common externally supplied data sequence.

Definition 3 (Protocol range)**.** *For a checkpoint t, seed s, and evaluation functional J, the protocol range of a candidate set $\mathcal{P}$ is*

$$\mathcal{B}_t(\mathcal{P}; s) = \max_{\pi \in \mathcal{P}} J(\theta_t^{\pi,s}) - \min_{\pi \in \mathcal{P}} J(\theta_t^{\pi,s}).$$

Definition 4 (Windowed L_2 protocol sensitivity)**.** *For a registered checkpoint window W,*

$$\mathcal{S}_{W,2}(\mathcal{P}) = \left(\mathbb{E}_s \left[|W|^{-1} \sum_{t \in W} \mathcal{B}_t(\mathcal{P}; s)^2 \right] \right)^{1/2}.$$

We call the registered set ε-equivalent when $\mathcal{S}_{W,2}(\mathcal{P}) \leq \varepsilon$, and ε-dependent otherwise. This is a finite-budget root-mean-square (RMS) criterion, not an expected-range criterion. For an endpoint pair, $\mathbb{E}_s|d_s| \leq \sqrt{\mathbb{E}_s d_s^2}$, so RMS equivalence is a conservative sufficient condition for expected absolute-gap equivalence, but not conversely. We report seed-wise ranges separately for ranking diagnostics.

Observed return differences also contain evaluation noise. An evaluation bank is a group of rollout episodes with its own random seed; paired protocols use matched episode seeds within each bank. For exactly two paired protocols, training seed s, checkpoint t, and independently seeded evaluation bank e, let X_{ste} be the paired return difference and $\mu_{st} = \mathbb{E}_e[X_{ste}]$. To assess differences across training rather than at the endpoint alone, we audit the latent trajectory estimand for a registered checkpoint window W:

$$Q_W = \mathbb{E}_s \left[|W|^{-1} \sum_{t \in W} \mu_{st}^2 \right].$$

Squaring prevents sign reversals from cancelling. When $\mathcal{P}$ consists of exactly two protocols and the same paired utility, checkpoint window, and evaluation-bank coupling are used, $Q_W = \mathcal{S}_{W,2}(\mathcal{P})^2$, so the certificate targets the defined sensitivity. For a library of three or more protocols, the best–worst range is a descriptive multi-protocol estimand; it is not identified with this two-protocol Q_W certificate. If D_{st} is the mean over E independent banks and V_{st} is its estimated measurement variance, then

$$D_{st}^2 - V_{st} = \frac{1}{E(E-1)} \sum_{e \neq f} X_{ste} X_{stf}$$

is an unbiased order-two U-statistic for μ_{st}^2: it averages products from distinct banks, removing the evaluation-noise contribution to the square. We cluster checkpoints within training seed and compare one-sided confidence bounds on Q_W with ε^2: an upper bound below the threshold certifies L_2 equivalence, a lower bound above it certifies L_2 dependence, and overlap yields *undetermined* rather than a forced conclusion. The certificate applies only to the registered protocols, budget window, utility, and tolerance. Our implementation uses Student-t bounds on independent training-seed scores, with checkpoints clustered within seed. These bounds are exact for Gaussian seed scores and otherwise rely on an asymptotic approximation; unbiasedness of the squared-effect estimator alone does not guarantee finite-sample coverage. The classifications are therefore model-dependent statistical evidence, especially with five heterogeneous seeds. Additional evaluation banks reduce measurement noise but do not increase the number of independent training seeds.

Removing order sensitivity need not preserve performance. We therefore evaluate a mitigation by a separate utility estimand. For an intervention M and baseline algorithm A, define its paired utility difference as

$$\Delta_B(M \mid A) = \mathbb{E}_s \left[J(\theta_B^{M,s}) - J(\theta_B^{A,s}) \right].$$

We call M (ε, δ)-admissible when its endpoint sensitivity $\mathcal{S}_{\{B\},2}$ is at most ε and $\Delta_B(M \mid A) \geq -\delta$ under the same interaction budget. Sensitivity, utility, and optimizer cost must be reported separately: exact invariance alone does not imply a better learner.

4 Why Update Order Matters

We first ask when order changes the complete learning state, before interpreting any observed utility difference. The commutator measures the difference between composing two updates in opposite orders. For smooth near-identity parameter maps, meaning updates that approach the identity as the step size vanishes, the leading order difference is $\eta^2[F_i, F_j]$; Appendix H gives the expansion, utility projection, and its controlled affine-system check. This is not a general Adam scaling law: at fixed momentum coefficients, memory and counters need not approach the identity as the learning rate vanishes, and clipping is not globally smooth.

Finite updates and accumulation. For an actual complete-state round map T_t^π, define $D_t(z) = T_t^\pi(z) - T_t^{\pi'}(z)$ under matched inputs. For two paths let $e_t = \|z_t - z_t'\|$ and $\delta_t = \|D_t(z_t')\|$. If ℓ_t bounds the change of T_t^π between these two inputs, the triangle inequality gives

$$e_{t+1} \le \ell_t e_t + \delta_t, \qquad e_B \le e_0 \prod_{t=0}^{B-1} \ell_t + \sum_{t=0}^{B-1} \delta_t \prod_{r=t+1}^{B-1} \ell_r.$$

One can always use the realized difference quotient when $e_t > 0$ and set $\ell_t = 0$ when the inputs coincide. This is then a retrospective path decomposition, not a predictive stability guarantee. A uniform bound requires independent control of these factors. We do not assume a uniform $O(\eta^3)$ remainder for the actual Adam implementation or equate local parameter differences with long-horizon utility differences. State norms require fixed coordinate scaling; our diagnostics report parameter, memory, and policy differences separately.

Proposition 1 (Commuting updates imply protocol invariance)**.** *If all update blocks commute on every state reachable from the initial state by the tested protocols and their adjacent-swap intermediates, i.e., $U_iU_j(z) = U_jU_i(z)$ for every such complete state, pair i, j, and training round t, then every deterministic protocol over the same multiset of blocks produces the same iterate after B rounds. Consequently, $\mathcal{S}_{W,2}(\mathcal{P}) = 0$ for every window and evaluation functional J.*

Proof. Any two permutations of the same multiset can be transformed into one another by adjacent swaps. Commutativity makes every adjacent swap leave the composed operator unchanged, so all protocols induce the same training map at each round and therefore the same final iterate. □

Pairwise commutation can also reduce the number of interventions to audit: orders connected by swaps of commuting blocks form one equivalence class. The collection of these classes is the protocol quotient. Let the dependency graph G contain one vertex per update block and an edge for each pair not certified to commute.

Theorem 1 (Partial-commutation protocol quotient)**.** *Let $\mathcal{C}$ be a symmetric pairwise commutation relation that is valid on every state reachable by the tested protocols and their adjacent-swap intermediates, and let G contain exactly the pairs not in $\mathcal{C}$. Two permutations belong to the same class under adjacent swaps of commuting blocks if and only if they induce the same relative order on every edge of G. The classes are therefore in one-to-one correspondence with acyclic orientations of G.*

Proof. Swapping a nonedge preserves every edge orientation, so each commutation class has one orientation. Conversely, two linear extensions of the same acyclic orientation are connected by adjacent swaps of incomparable vertices, which are nonadjacent in G and hence commute. □

This specializes the classical trace-monoid correspondence, which groups sequences by swaps of commuting elements, to protocols without repeated blocks (3). The quotient is exact only when a symmetric, state-valid commutation relation has been supplied and verified for every reachable intermediate state; a heuristic or unverified "not observed" relation does not justify quotienting. It collapses all orders for disjoint frozen-state actors to one class, but gives no reduction for a complete dependency graph; it avoids factorial sweeps without assuming that unknown pairs commute.

Proposition 2 (Frozen shared-state counterfactual)**.** *At each training round, suppose actor i has disjoint local state (parameters and optimizer memory) and reads only that local state plus frozen shared data/state, every actor reads the same frozen critic ψ_t without modifying it, and one critic map updates ψ_t from a permutation-independent joint loss after all actor updates. Then every actor permutation has the same round map and hence $\mathcal{S}_{W,2}(\mathcal{P}) = 0$ for every window and evaluation functional J.*

Proof. With shared state fixed and random inputs assigned by actor, actor maps modify disjoint local coordinates and therefore commute pairwise. Their composition is permutation-invariant, and the subsequent joint critic map receives the same actor parameters, batch, and joint loss. Induction over training rounds gives identical iterates for all protocols. □

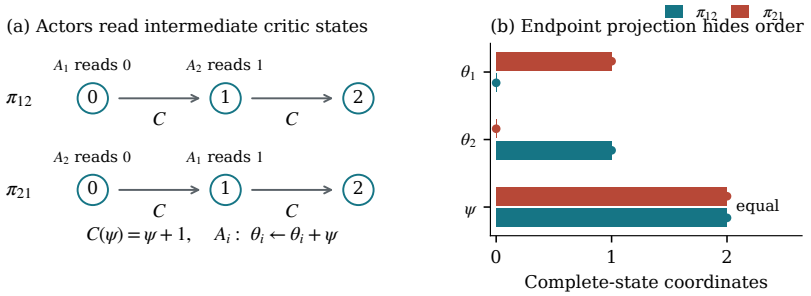


Figure 1: Exact two-actor blind-spot construction. With $C(\psi) = \psi + 1$ and $A_i(\theta_i; \psi) = \theta_i + \psi$, reversing the actor order changes the actor endpoints from $(0, 1)$ to $(1, 0)$ while both protocols end at critic state $\psi = 2$. A critic-only endpoint therefore cannot certify complete-state or policy invariance.

This proposition gives an exact source intervention rather than a heuristic schedule. It does not guarantee utility preservation, and it can fail under actor parameter sharing or any order-dependent mutation of replay, normalization, or target-network state.

A critic-endpoint blind spot. Let C update the complete critic state and A_i update actor i using the current critic. Suppose each critic step implements the same map C of critic state alone, with the same batch, targets, loss, and matched random inputs. Both actor orders then end at $C^m(\psi)$ in the critic projection. However, the actor at position k reads $C^{k-1}(\psi)$. Reordering therefore changes its input even when the final critic states agree. A constructive two-actor example takes scalar states, $C(\psi) = \psi + 1$, and $A_i(\theta_i; \psi) = \theta_i + \psi$. Starting at zero, the interleaved orders give actor endpoints $(0, 1)$ and $(1, 0)$, with critic endpoint 2 in both. Thus critic-endpoint equality cannot certify complete-update invariance. Section 6.3 tests this distinction in the implemented actor–critic loop; the construction alone says nothing about return.

These results give a criterion for deciding when order needs an empirical audit. Coupled sequential blocks that fail to commute lose the structural guarantee of invariance. This does not imply that every MARL algorithm, or every utility measured from those blocks, is protocol-dependent.

Changing order can also reverse a comparison with a fixed baseline whenever its utility lies between the two protocol outcomes. A constructive example and proof appear in Appendix H.1; the empirical ranking comparisons below concern observed sample means, not population reversals.

5 The Protocol-Bias Audit

The audit treats update order as an experimental variable in existing sequential or block-decomposable MARL methods. It specifies which orders to compare, which observable to measure, and when the evidence remains inconclusive, without introducing a new optimizer.

For larger libraries, the structural quotient can remove redundant protocols before selecting a pair. Separate pairwise intervals do not provide a simultaneous certificate for the entire library without an additional multiple-comparison procedure and a matching library-level estimand.

Detection versus selection. A magnitude detector predicts where order has a large effect; a signed selector predicts which order is better. Commutator magnitude can prioritize expensive paired interventions, while equivalence is assessed by the noise-corrected comparison above. Selecting an order also requires a stable sign, so failure of signed transfer does not rule out sensitivity or predictive information in its magnitude.

Algorithm 1 Protocol Equivalence Audit

Require: Complete-state blocks $\{U_i\}_{i=1}^m$, a pre-specified pair $\mathcal{P} = \{\pi, \pi'\}$, seeds S, window W, bank count $E \geq 2$, tolerance ε, confidence procedure

Specify data, per-block random inputs, utility, and independent sampling units
Report any exact equivalence justified by a valid commutation relation; otherwise retain both protocols
for each seed $s \in S$ **do**
  for each protocol $\pi \in \mathcal{P}$ **do**
    Train with paired randomness and record E independently seeded evaluation-bank returns at each $t \in W$
  end for
end for
Estimate Q_W with the order-two U-statistic and seed-cluster confidence bounds
Return pairwise EQUIVALENT, DEPENDENT, or UNDETERMINED at ε, conditional on the confidence procedure's coverage assumptions
if protocol-dependent **then**
  Apply a source counterfactual that freezes or jointly updates shared state, when available
  Report residual sensitivity, paired utility difference, and optimizer-step cost
end if
Optionally use commutator magnitude to prioritize additional paired probes; never treat it as a certificate
Report protocol classes, bank independence unit, window, tolerance, status, utility, ranking flips, and compute

6 EXPERIMENTS

We organize the experiments around four predefined evaluation targets. Q1 tests whether the commutator view predicts sensitivity even when the sign of local order preference does not transfer. Q2 tests whether protocol sensitivity appears in real MARL training loops. Q3 tests whether the measured sensitivity is explained by coupled sequential shared-state updates rather than by random implementation noise. Q4 tests whether a source intervention can restore invariance without sacrificing utility.

6.1 CONTROLLED SANITY CHECK

The structural analysis supplies a controlled check for the audit. If update blocks commute, adjacent swaps leave the composed map unchanged. In the smooth near-identity special case, the second-order term is the Lie bracket; it is the leading term only when nonzero. Shared writes alone do not imply either noncommutativity or this scaling in a general optimizer. This motivates an independent-critic actor–critic (I-AC) negative control with near-commuting updates and a sequential shared-critic actor–critic (S-AC) positive control with coupled updates. The comparison tests whether the audit separates invariant and protocol-dependent loops under matched budgets.

6.2 CONTROLLED LOCAL REAL-MARL GRID

The local PettingZoo/MPE grid uses simplified actor–critic controls rather than standard PPO, IPPO, or MAPPO training: I-AC has independent local critics; S-AC has independent actors and a shared critic updated after each actor. Both take one Adam actor step per episode on $-\log \pi_i(a_i \mid o_i)\widehat{A}_i$ with entropy regularization and gradient-norm clipping. They do not implement the PPO likelihood-ratio objective, ratio clipping, or repeated minibatch epochs. These control names do not denote standard IPPO or MAPPO. In S-AC, later actors recompute their advantages using the intermediate critic, creating coupled update blocks. We evaluate fixed-order libraries and a uniform randomized-order baseline over 10 paired seeds; selector variants are reported separately as stress tests. The canonical table uses 1500 training episodes, evaluation every 100 episodes, and ten evaluation episodes per checkpoint. Returns average the last ten checkpoints. These curve means support descriptive comparisons; bank-corrected equivalence is evaluated separately on the HARL data.

Table 1: Controlled protocol-bias audit on PettingZoo/MPE at 1500 episodes. I-AC is the independent-critic negative control; the S-AC shared-critic loop is the coupled-update positive control. The paired protocol range computes best-minus-worst fixed return within each seed and then averages across seeds; returns and ranges use the last ten checkpoints of the canonical raw grid. All $\pm$ values are SEM across ten training seeds.

Environment	Algorithm	Episodes	Uniform return	Paired range
simple_adversary	I-AC	1500	-3.43 ± 0.69	0.00 ± 0.00
simple_adversary	S-AC	1500	2.47 ± 0.95	5.95 ± 0.62
simple_spread	I-AC	1500	-28.02 ± 0.90	0.00 ± 0.00
simple_spread	S-AC	1500	-27.86 ± 0.65	5.67 ± 0.30
simple_tag	I-AC	1500	0.63 ± 0.44	0.00 ± 0.00
simple_tag	S-AC	1500	-0.16 ± 0.77	5.11 ± 0.51

Table 1 separates the two controls: protocol choice has zero observed effect with independent critics and a large effect with the sequential shared critic. This contrast is consistent with sensitivity arising from coupled update blocks within the tested implementation and budget; it does not identify the mechanism beyond that setting. The table is descriptive. Each checkpoint supplies one evaluation mean, without the independent-bank variance estimate needed for an ε-equivalence certificate. A separate five-checkpoint trajectory audit also finds zero I-AC differences and nonzero absolute S-AC tail differences. It likewise uses curve means rather than banks and is not a Q_W certificate; Appendix G distinguishes the inputs. The I-AC result is thus a registered zero-difference control, while the shared-critic result makes order a necessary part of the reported algorithm specification. The equivalent/dependent/undetermined labels below instead use bank-corrected seed-level t intervals under the stated coverage and independence assumptions. The practical implication is to audit invariance before leaving order unspecified, not to randomize every update loop.

Protocol choice also changes the sample-mean comparison with I-AC (Table 4). This is a same-sample descriptive ranking diagnostic: the best and worst orders are selected and evaluated on the same ten-seed cohort, so it is not a population ranking reversal or evidence that a fixed order is better. Because the same seeds select and evaluate the extreme orders, population ranking reversals require confirmation on independent seeds or simultaneous inference over the full order library. On *Simple Adversary*, S-AC training outperforms I-AC under both the best and worst fixed protocols, so there is no rank reversal despite a sizable fixed-order gap. Ranking flips are therefore a possible consequence of protocol dependence, not its definition.

The paired ranges in Table 1 are larger than the gaps between globally best and worst fixed protocols, which are 1.31, 1.56, and 3.14 on *Simple Spread*, *Simple Tag*, and *Simple Adversary*, respectively. The larger within-seed ranges show why an effect of order need not imply that one fixed order is best across seeds.

6.3 Mechanism Diagnostics

We examine the role of shared-state reads by varying the critic-update loop. The coupling sweep varies $\lambda \in [0, 1]$, where $\lambda = 0$ moves the critic update outside the per-agent order loop and $\lambda = 1$ performs the fully sequential shared-critic update. For each seed we take the endpoint return range over all six fixed orders, then average over five seeds. At 500 episodes this range is zero at $\lambda = 0$ and 11.75, 9.19, 13.95, and 14.40 at $\lambda = 0.25, 0.50, 0.75, 1.00$, respectively. This is a composite implementation intervention: loss scaling under Adam does not equal proportional step-size scaling, and the three-agent endpoints and interior settings use one, three, and four critic steps, respectively. The sweep therefore does not isolate a continuous coupling-strength effect. A separate budget sweep gives endpoint ranges 23.18, 13.92, 12.12, and 8.55 at 100, 300, 800, and 1500 episodes. The range decreases across these four budgets but remains nonzero; this does not establish a universal attenuation law. These endpoint ranges differ from the canonical tail-10 range in Table 1.

Exact counterfactual and utility preservation. We implement Proposition 2 by freezing the shared critic during all disjoint actor updates and applying one permutation-independent joint critic update afterward. Forward, reverse, and randomized actor schedules are checkpoint-wise identical

Table 2: Frozen shared-state counterfactual. Moving the shared critic update outside the ordered actor loop restores exact protocol invariance. Tail-10 differences use 95% paired percentile-bootstrap intervals (100,000 resamples over training seeds). Non-inferiority uses a one-return-point margin; failure to establish it does not establish a utility loss.

Environment	Seeds	Critic-step reduction	Tail-10 utility delta	Non-inferior?
simple_spread	30	$3\times$	$+0.39\ [-0.84, 1.66]$	yes
simple_tag	10	$4\times$	$-0.43\ [-1.71, 0.87]$	no
simple_adversary	10	$3\times$	$-1.04\ [-2.78, 0.67]$	no

Table 3: Critic endpoints can miss actor-order effects. "Detected" means policy TV exceeds 10^{-8} on the observation bank. These small local effects do not establish a return difference.

Interleaved update targets	Critic probe zero	Policy detected	Mean TV
Real batch targets	17/30	30/30	1.2452×10^{-4}
Equal-target control	30/30	30/30	1.2907×10^{-4}

in an implementation test, while critic optimizer steps fall from one per agent to one per episode at unchanged interaction budget.

Restoring invariance also reduces critic optimizer steps (Table 2). Return non-inferiority is supported on spread and unresolved on tag and adversary; their intervals allow both improvement and loss. The intervention changes critic update count as well as actor exposure to intermediate critic states.

Sensitivity detection and the boundary of signed selection. Finite-update utility projections rank held-out sensitivity of the frozen surrogate within controlled MPE (mean within-seed Spearman 0.78), but this is not an environment-return predictor. Independent-data sign selection and HARL detector transfer fail their respective checks. Appendix A retains the full cohorts, tests, controls, and transfer limitations.

Complete-update measurement. We compare critic-only endpoint distance with a complete forward/reverse actor–critic probe in the controlled CPU spread loop. Models and optimizers are cloned together; each branch starts from identical state and matched random inputs. Actor parameters, critic parameters, optimizer memory, and policy outputs are compared separately. Ten training seeds contribute checks before updates 1, 10, and 20; these are 30 repeated checkpoints, not 30 independent training trials. Policy total variation (TV) is averaged over agents and a fixed observation bank collected before training.

The policy comparisons expose effects that the critic endpoint misses (Table 3). Controls that update all actors before one or three mean-loss critic steps have zero policy difference at all 30 checkpoints in each target condition. The complete probe passed original-state and gradient preservation checks in 180 cases, with ten additional exception-cleanup checks. Its distances compare floating model/optimizer tensors, not arbitrary non-tensor metadata. The comparison covers only forward and reverse order; zero TV on this bank does not prove global commutativity. Each complete probe costs six actor and six critic optimizer steps, versus six critic steps for the old probe, in addition to cloning and policy evaluation. The wider measurement scope thus adds computation.

Appendix Figure 2 visualizes the distinct protocol-range and frozen-intervention estimands in Tables 1 and 2.

Return-based scheduling remains unresolved. The completed 10M-step spread grid does not establish a scheduling advantage: the diagnostic-minus-uniform return difference is $+0.529$ with a 95% interval $[-2.411, 3.470]$. Cost, intervention frequency, missing summaries, and overlapping reset seeds prevent causal attribution. Appendix A retains the full comparison and its limitations; Table 6 reports both frozen primary metrics.

6.4 HARL-Base Runtime Audit

The HARL audit uses a fixed upstream revision and the discrete Simple Spread environment, with disclosed changes for the environment, backend, evaluation banks, and random-stream instrumentation. Exact revision identifiers and source paths are provided in the supplementary artifact. In the native loop, the global Torch RNG assigns minibatch permutations by loop position, so reversing the loop changes both execution order and random-stream assignment. The primary HAPPO intervention instead serializes persistent agent-local streams, giving each agent identical draws under both orders. Five paired seeds run to 10M steps per branch with 100 checkpoints and ten paired evaluation banks (100M training steps; 20,000 returns). The estimated trajectory sensitivity is large, but the effects vary in magnitude and sign across seeds. This agent-local extension was analyzed after the original MPE registration and is exploratory. In separate native HAPPO audits on spread, reference, and speaker–listener, full trajectories and late-window behavior differ: the two two-agent tasks have certified tail/endpoint equivalence, whereas intervals for three-agent spread remain wide throughout. These native comparisons characterize sensitivity to the runtime protocol over different windows; they do not identify a universally better HAPPO order. The full audit table appears in Appendix D. A separate completed cohort, seeds 6–10, has full-trajectory RMS 9.693 $[1.161, 13.658]$ and endpoint RMS 9.742 $[0, 14.879]$. Table 8 reports all windows. Both cohorts are post-registration exploratory evidence, without retrospective claims of confirmation.

7 Limitations

Each agent-local HARL cohort has five heterogeneous seeds, so sensitivity in this implementation is not a population-level HAPPO result. Native comparisons also change RNG assignment and cannot identify a trust-region cause. The quotient requires valid commutation; the local commutator prioritizes audits without certifying long-horizon utility gaps. The full-update probe uses a controlled CPU loop and fixed observation bank, not a standard learner's return. Surrogate projection, policy TV, and return are distinct. Scheduling has no established stable benefit; null intervals establish neither equivalence nor a cause of failure.

8 Conclusion

Execution order can change coupled MARL updates even with fixed data and random streams. An audit must follow the complete learning state: equal critic endpoints can coexist with different actor policies. Structural commutation justifies grouping equivalent orders; a noise-corrected utility comparison assesses a specified pair, window, and tolerance under stated coverage assumptions. Keeping these claims distinct makes order comparisons reproducible without turning local sensitivity into a claim of better scheduling.

Reproducibility Statement

The supplementary package contains the current source and PDF, an evidence index, a machine-readable input manifest, and an offline verification entry point. Appendix G maps the main claims to their cohorts and estimands, including missing historical artifacts.

AI Use Statement

Generative AI tools assisted with mathematical revisions, code development, and analysis of experimental results. The authors take responsibility for the final content.

References

[1] Agarwal, R., Schwarzer, M., Castro, P. S., Courville, A., and Bellemare, M. G. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, 2021.

[2] Blanes, S., Casas, F., and Murua, A. Splitting methods for differential equations. *Acta Numerica*, 33:1–161, 2024.

[3] Cartier, P. and Foata, D. *Problèmes combinatoires de commutation et réarrangements*. Springer, 1969.

[4] Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. Implementation matters in deep policy gradients: A case study on PPO and TRPO. *arXiv preprint arXiv:2005.12729*, 2020.

[5] Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Advances in Neural Information Processing Systems*, 2017.

[6] Yu, C., Velu, A., Vinitsky, E., Gao, Y., Wang, Y., Bayen, A., and Wu, Y. The surprising effectiveness of PPO in cooperative multi-agent games. In *Advances in Neural Information Processing Systems*, 2022.

[7] Zhong, Y. et al. Heterogeneous-agent reinforcement learning. *Journal of Machine Learning Research*, 2024.

[8] Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 32:2974–2982, 2018.

[9] Rashid, T. et al. QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4295–4304, 2018.

[10] Sunehag, P. et al. Value-decomposition networks for cooperative multi-agent learning. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 2085–2087, 2018.

[11] Lauer, M. and Riedmiller, M. An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 535–542, 2000.

[12] Tan, M. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 330–337, 1993.

[13] Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. Dealing with non-stationarity in multi-agent deep reinforcement learning. arXiv preprint arXiv:1906.04737, 2019.

[14] Lanctot, M. et al. A unified game-theoretic approach to multiagent reinforcement learning. In *Advances in Neural Information Processing Systems*, 30:4190–4203, 2017.

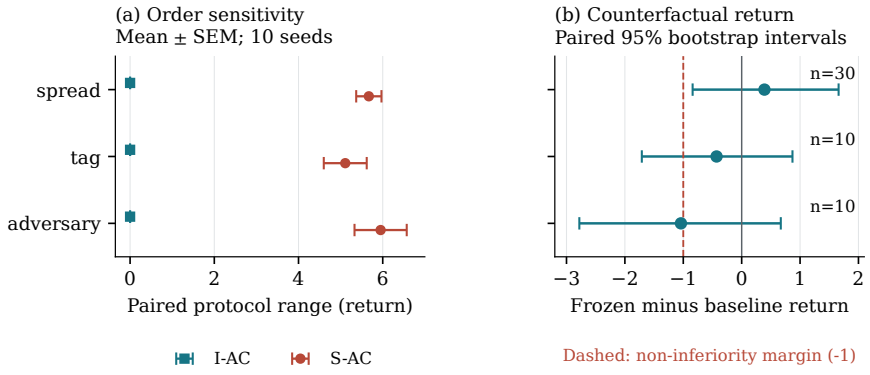


Figure 2: Two distinct estimands in the controlled MPE evidence. Panel (a) shows the existing paired protocol ranges for independent-critic (I-AC) and sequential shared-critic (S-AC) controls over ten training seeds. Panel (b) shows the existing frozen-counterfactual utility deltas with paired 95% bootstrap intervals; the dashed line is the registered non-inferiority margin. The figure does not pool cohorts or establish a return-improving schedule.

Table 4: Observed sample-mean ranking flips. In two of three environments, the best and worst fixed-order means straddle the I-AC uniform mean. Orders are selected on the same seed set; population ranking reversals have not been independently confirmed.

Environment	I-AC uniform	S-AC best	S-AC worst	Flip?
simple_adversary	−3.43	3.41	0.27	no
simple_spread	−28.02	−27.68	−28.99	yes
simple_tag	0.63	1.17	−0.39	yes

[15] Lanctot, M. et al. OpenSpiel: A framework for reinforcement learning in games. arXiv preprint arXiv:1908.09453, 2019.

[16] Geist, M., Scherrer, B., and Pietquin, O. A theory of regularized Markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, 97:2160–2169, 2019.

[17] Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*, pages 1861–1870, 2018.

[18] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017.

[19] Puterman, M. L. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 2014.

[20] Cesa-Bianchi, N. and Lugosi, G. *Prediction, Learning, and Games*. Cambridge University Press, 2006.

A Detector and Scheduling Details

Sensitivity detection and the boundary of signed selection. We branch every pair and full permutation from exactly the same model, Adam state, and training batch at five checkpoints; restoration after every probe is exact. The measured utility $\widehat{J}$ combines the frozen-advantage policy surrogate, entropy, and a negative critic squared-error term. It is not the environment return of a branch policy. The projection is $p = \nabla\widehat{J}(\theta_0)^\top(\theta_+ - \theta_-)$, computed from executed finite parameter differences, rather than an independent evaluation of $\eta^2[F_i, F_j]$. Same-sample agreement

checks this finite-difference projection, not an Adam version of Proposition 4. Independent rollout data give pair-sign accuracy only 0.50–0.60 and normalized regret 0.31–0.39 for this surrogate selector. The magnitude test targets a different estimand, fixed before collecting seeds 5–9: does mean absolute projection rank checkpoints by their mean absolute surrogate effects on 16 independent rollouts? The confirmatory mean within-seed Spearman correlations are 0.90, 0.74, and 0.70 on adversary, spread, and tag; all 15 seed–environment correlations are positive (two-sided sign test $p = 6.1 \times 10^{-5}$ under independent, equiprobable null signs; 50,000-draw block permutation $p = 2.0 \times 10^{-5}$). These inferential calculations assume independent seed–environment clusters. The discovery split independently gives 15/15 positive clusters and mean correlation 0.74. As a secondary allocation metric, the detector identifies the checkpoint with the largest measured surrogate effect among five in 11/15 held-out clusters (73.3% versus a 20% random baseline), and its top two recover 80% of the measured top two. In the frozen raw-norm follow-up on seeds 10–14, the parameter-commutator norm fails the adversary and spread gates. A same-cohort reanalysis averages absolute pair effects within each checkpoint and computes Spearman correlation over five checkpoints per seed and environment. Mean correlations for projection versus norm are $0.76/0.46$, $0.84/0.40$, and $0.94/0.88$ on adversary, spread, and tag, respectively. The mean paired difference is 0.267, with ten positive, four tied, and one negative difference across the 15 seed–environment pairs. This descriptive comparison uses existing records, not fresh confirmation or a pooled-cluster significance test. A preregistered transfer to HARL-base HAPPO speaker–listener also fails: mean projected Spearman is 0.38 and the block-permutation $p = 0.05094$, missing both frozen thresholds despite exact restores. Magnitude therefore predicts sensitivity of the named surrogate in the controlled MPE setting, but not environment return; these results do not establish a transferable selector or detector.

Return-based scheduling remains unresolved. A separate scheduling experiment compares four methods on five seeds each at 10M interaction steps per cell (200M total). Spread is the only completed environment among the three planned. The critic-only gate's final return difference is $+0.529$ ($[-2.411, 3.470]$) against uniform sequential order and $+0.258$ ($[-1.502, 2.019]$) against a probability-0.5 random gate. These are approximate, unadjusted two-sided 95% Student-t intervals over five paired seeds. Neither establishes an advantage; both frozen primary metrics, including minimum-agent return, appear in Table 6. The random gate does not match the diagnostic gate's 14.17–42.40% joint-update rates (its available cells are 49.97–50.07%), and the diagnostic cells add 2.4M probe critic optimizer steps each. The cost/frequency fields are available for 5/5 diagnostic cells but only 2/5 random-gate cells; five of the 20 return summaries are missing. Thus the completed grid is not a matched-cost or matched-frequency causal comparison. Environment reset seeds also overlap across training seeds. The paired return intervals below are descriptive comparisons over the common completed seed rows, not an estimate of the diagnostic's isolated contribution. Appendix B separates this extension from the short-budget follow-ups and records their costs and remaining uncertainties.

B Follow-up Scheduling Experiments

These follow-ups are separate from the original controlled MPE and HARL audits. They use the controlled CPU actor–critic loop, not a full standard MAPPO implementation. They test whether measured order differences can guide return-based interventions; they are not evidence of a new optimizer.

Completed 10M spread extension. Uniform sequential order, joint critic updating, the critic-only diagnostic gate, and a probability-0.5 random gate each have five completed 10M cells. Only spread is complete (20 of the 60 planned cells). The diagnostic gate's joint fractions differ from the random control, violating the planned frequency matching. Each diagnostic cell also uses 2.4 million probe critic optimizer steps in addition to training updates. Five of the 20 summaries are missing; unavailable cost fields remain missing. The reset formula $100000s + e$, where e is the episode index, overlaps across seeds at 400000 episodes, so the reported seed intervals are not an unconditional independent-environment coverage guarantee. No policy checkpoints permit fresh evaluation of this grid. These limitations prevent a causal attribution of its return outcomes to the diagnostic alone.

Short-horizon utility prediction. Five discovery seeds train a fixed ridge predictor, followed by five distinct confirmation seeds with three snapshots each. Inputs are training stage and current return, with or without complete-probe policy TV. From each snapshot, three branches undergo five updates. Selection is offline from sequential parent states, with a fixed mode per branch, not a deployed scheduler. The augmented model fails the frozen advancement rule. The original screen saved no policies: confirmation parents and branches were reconstructed, their original returns and TV reproduced to 10^{-10}, and the policies saved. Re-evaluating these branches using two independent banks of 64 greedy episodes gives matching joint-minus-sequential signs at only 7/15 snapshots, with no pooled snapshot interval excluding zero. The augmented and baseline mean squared errors are 0.368 and 0.100; five-seed uncertainty does not establish population-level dominance. Discovery labels still use eight rollout evaluations, so this re-evaluation does not rule out training-label noise. It is not a second independent training confirmation. The randomized comparator is an expected mixture of branch outcomes at the discovery-frozen probability $4/15$, not a deployed closed-loop controller.

Matched-update-count intervention. Five new seeds train four arms for 500 episodes each: 250000 total training steps, plus 64000 evaluation steps. All actors in the delayed-ordered arm read the pre-update critic, after which the original ordered critic losses are applied. This matches the sequential arm's three critic updates per episode; it does not make heterogeneous critic updates commute. Separate controls take three mean-loss critic steps or one mean-loss step. At the frozen final endpoint, delayed-ordered minus sequential return is $+0.500$ ($[-1.872, 2.872]$, five paired seeds; 3/5 positive). It fails the advancement rule requiring mean gain at least one, a positive lower bound, and at least 4/5 positive seeds. The interval also permits meaningful gains, so the result does not establish ineffectiveness. Warmed fixed-batch checks verify identical critic endpoints and optimizer memory for sequential and delayed-ordered updates; these checks occurred after nine updates, rather than before training as planned. The intervention does not identify the cause of the earlier 10M gate outcomes.

A fresh evaluation of the same ten final policies uses two banks of 128 paired episodes per arm and seed (2560 episodes, no further training). Delayed-ordered minus sequential return is $+0.737$ ($[-1.169, 2.644]$, 95% paired t interval over the same five training seeds), with 3/5 positive seed differences in both banks. The advancement rule still fails. More evaluation reduces rollout uncertainty; it does not add independent training replications or establish ineffectiveness.

All-six-order checks at these five saved sequential states further separate actor agreement from complete-state invariance. Delayed-ordered updates have identical actors after one fixed-batch round at every state. At one state, however, the critic and its optimizer memory differ across orders; after a second round on the same batch, mean policy TV reaches 2.76×10^{-7}. Three mean-loss critic steps preserve exact complete-state equality at both rounds in all five states. These local checks expose delayed propagation of order effects, not a global commutation guarantee or a return benefit.

Artifact scope. The short-horizon re-evaluation contains 5760 new evaluation episodes and 45 saved branch policies; the delayed intervention contains 2560 evaluation episodes and 40 saved checkpoints, with 2560 additional episodes in the final-policy re-evaluation above. Models and optimizer states support policy re-evaluation, not exact training resumption without the missing environment and random states. These short-budget diagnostics are not additional 10M efficacy trials. Reconstruction additionally repeats parent training, five-update branches, the original eight-episode evaluations, and observation bank collection; the 5760 episodes count only the fresh re-evaluation. They motivate retaining sensitivity measurement while withholding a performance-improvement claim.

C Reproducibility Artifact

Finite-sample alternative and calibration. For a fixed protocol pair and window, suppose an independently justified bound $|X_{ste}| \le B$ holds almost surely, evaluation banks are conditionally independent with a common conditional mean, and training seeds are independent. The per-seed U-statistic $Z_s = |W|^{-1} \sum_t [D_{st}^2 - V_{st}]$ averages products of distinct bank differences, hence $Z_s \in [-B^2, B^2]$ and $\mathbb{E}Z_s = Q_W$. Hoeffding's inequality gives a two-sided interval $\bar{Z} \pm B^2 \sqrt{2\log(2/\alpha)/S}$ with coverage at least $1 - \alpha$, intersected with $[0, B^2]$. The same toler-

ance comparison yields a finite-sample classification, often unresolved with few seeds. The bound must follow from the task specification; an observed maximum is insufficient. Historical tables use the approximate t procedure, not this bounded alternative.

A diagnostic with 2,000 replications per condition illustrates the distinction. We use ten checkpoints and ten independent Gaussian-noise banks, with unit noise standard deviation. With probability 0.1 a seed has constant latent effect $\sqrt{Q_W/0.1}$; otherwise its latent effect is zero. At $Q_W = 1.1$ and tolerance one, nominal 95% t-interval coverage is 41.45%, 64.90%, and 89.85% for 5, 10, and 30 seeds; false-equivalence rates are 58.50%, 34.95%, and 11.60%. These synthetic results expose a small-sample failure mode and do not estimate error rates in the MARL tasks.

The artifact contains scripts, input manifests, aggregate CSVs, and source audits for the MPE grid, protocol quotient, synthetic certificate calibration, coupling and budget sweeps, frozen joint-critic counterfactual, commutator detector, failed signed selectors, and disclosed HARL-base checkpoint banks. The numerical values in Table 1 are generated from the canonical tail-window records. The supplementary manifest records the raw curves, source hashes, analysis parameters, and reconstruction procedure; the historical earlier tail audit is retained only as a prior snapshot. External HARL evidence records runtime identity, random-stream semantics, backend, independent bank seeds, complete trajectories, and separate agent-local HAPPO, native global-RNG endpoint, and native HATRPO certificates.

D HARL-Base Checkpoint-Bank Audit

Table 5 records the HARL-base checkpoint-bank audit used to separate isolated execution-order interventions from broader runtime protocol changes. RMS intervals are clustered by training seed and corrected using independent evaluation banks. Agent-local rows isolate execution order; native rows retain order-dependent global-RNG assignment and backend semantics and are therefore not pooled with agent-local rows.

Table 5: HARL-base checkpoint-bank audit. The Banks column is checkpoints / evaluation banks $\times$ episodes per bank. Displayed bounds are two-sided 95% seed-cluster t intervals transformed from Q_W to RMS; decisions use one-sided 95% bounds on Q_W at $\varepsilon = 1$. Agent-local rows isolate order; native rows audit broader runtime protocols.

Algorithm/window	Banks	Corrected RMS interval	Interpretation
HAPPO/spread local full	$100/10 \times 2$	$7.066\ [0, 10.816]$	isolated order
HAPPO/spread local tail	$10/10 \times 2$	$7.345\ [0, 12.195]$	large tail point estimate; certificate undetermined
HAPPO/spread local endpoint	$1/10 \times 2$	$5.274\ [0, 8.911]$	sign-changing
HAPPO/reference native tail	$10/10 \times 2$	$0.052\ [0, 0.159]$	equiv., $\varepsilon = 1$
HAPPO/reference native endpt.	$1/10 \times 2$	$0.055\ [0, 0.138]$	equiv., $\varepsilon = 1$
HAPPO/speaker native tail	$10/10 \times 2$	$0.059\ [0, 0.158]$	equiv., $\varepsilon = 1$
HAPPO/speaker native endpt.	$1/10 \times 2$	$0.162\ [0, 0.294]$	equiv., $\varepsilon = 1$
HATRPO/spread native full	$100/10 \times 2$	$1.684\ [0, 2.829]$	runtime audit

E RNG Isolation Control

Agent-local streams are persistent, serialized with restart state, restored around each named actor update, and tested not to advance the global RNG. A matched HAPPO speaker–listener prefix control uses seeds 1–3, four checkpoints through 400k steps, and ten paired banks. Relative to native global-RNG assignment, agent-local streams reduce corrected full-trajectory, trailing-two, and endpoint RMS by 50.4%, 44.5%, and 73.9%. A separate probe-on/probe-off control reports bit-identical bank CSVs and complete restart snapshots; the compact evidence package retains the bank CSVs and equality report, but not the large restart snapshots. These controls establish that global-RNG assignment is load-bearing and that the registered instrumentation is neutral; neither short control supplies a population certificate.

F Native Runtime-Protocol Boundary

The completed native-global-RNG HAPPO grid spans three base tasks, five paired seeds, and 10M steps per branch. Because loop reversal also reassigns global RNG draws and the backend is part of the run identity, these rows audit runtime-protocol sensitivity rather than pure execution order.

Task	Full trajectory	Tail 1M	Endpoint
simple spread	5.016 wide CI	5.838 wide CI	4.054 wide CI
simple reference	1.333 wide CI	0.052 equiv.	0.055 equiv.
speaker–listener	4.211 wide CI	0.059 equiv.	0.162 equiv.

All statuses use the registered $\varepsilon = 1$ certificate. The two two-agent tasks wash out by late training, whereas three-agent spread remains unresolved; task, role structure, agent count, backend, and RNG mapping are confounded, so this contrast is a boundary rather than a causal explanation.

Detector transfer boundary. On HARL-base HAPPO speaker–listener, the frozen utility-projected detector has mean Spearman 0.38, four of five positive seed correlations, block-permutation $p = 0.05094$, and top-1 recovery $1/5$. It fails the preregistered mean-correlation and p-value gates despite zero restore distance. Raw parameter norm is not a replacement claim because it was not registered and fails locally.

Supported claims. Within the canonical MPE inputs, I-AC has zero paired range, the simplified S-AC loop has 5.11–5.95 ranges, and two same-sample descriptive ranking flips rather than population ranking estimates. The quotient identifies structurally equivalent classes; the bank-corrected test classifies specified pairs under its coverage assumptions. The frozen counterfactual is exactly invariant, with return preservation unresolved in some settings. Finite-update projection magnitude ranks held-out sensitivity of the named surrogate, not environment return, in controlled MPE, while signed selection fails independent transfer. Disclosed agent-local HAPPO shows large, seed-heterogeneous isolated-order estimates; native HAPPO/HATRPO audit broader runtime protocols and cannot identify pure order or trust-region causes.

Selector boundary. Balanced cycles, disjoint-seed selection, and role-first schedules fail paired confirmation. A privileged six-branch planner with a selective fallback rule gains $+1.74$ over uniform order on five held-out spread seeds (60% selection coverage), at roughly $16\times$ the continuation budget. This is an oracle boundary, not a matched-budget algorithm.

G Evidence Map and Additional Results

Controlled learners and estimands. I-AC and S-AC denote the simplified independent-critic and sequential shared-critic actor–critic controls defined in the main text. The archived control names are retained only in the supplementary artifact; they do not identify standard PPO implementations. The supplementary evidence index maps each result to its input cohort, analysis procedure, and statistical unit. A machine-readable manifest records file hashes and separates raw inputs, derived summaries, source snapshots, and unavailable historical state.

Table 1 and Table 4 use the canonical tail-window manifest. It identifies 460 curves and their summaries. The separate descriptive trajectory comparison uses a paired local-tail summary: its primary local tail contains five checkpoints and its observable is the absolute paired gap in curve means. It is not an independent-bank, noise-corrected Q_W certificate. For the banked HARL comparisons, the independent unit is the training seed; each checkpoint has ten independently seeded banks of two episodes per branch.

Frozen utility comparison. Table 2 uses paired tail-10 utility differences and 95% percentile-bootstrap intervals, with 100,000 seed resamples. The spread cohort has 30 seeds; tag and adversary have ten each. Its source curves and per-seed rows are separate from the canonical descriptive grid. Non-inferiority requires the interval lower endpoint to exceed -1. Neither failed non-inferiority nor a null superiority interval establishes equivalence. The source intervention changes critic update count as well as which critic state actors read.

Formal spread comparison and resource audit. The four methods below have five completed cells each at 10M training steps, for 200M training steps in total. The original plan contains three environments and 60 cells; only the 20 spread cells are complete. Both frozen primary metrics are shown. Methods use the simplified S-AC loop, not standard MAPPO.

Table 6: Completed 10M spread subset. Differences are method minus baseline; higher is better. Bounds are unadjusted two-sided 95% Student-t intervals over five paired training seeds. Overlapping training-reset seeds limit an unconditional independent-environment coverage interpretation.

Comparison	Final mean return	Final minimum-agent return
Joint − uniform	$+0.640\ [-3.239, 4.519]$	$+0.588\ [-3.008, 4.184]$
Diagnostic − uniform	$+0.529\ [-2.411, 3.470]$	$+0.499\ [-2.384, 3.383]$
Random − uniform	$+0.271\ [-2.783, 3.325]$	$+0.189\ [-2.689, 3.068]$
Diagnostic − random	$+0.258\ [-1.502, 2.019]$	$+0.310\ [-1.302, 1.922]$

The resource audit below uses the per-cell fields in the frozen resource summary; blank entries are reported as unavailable rather than imputed. "Training critic steps" and "probe critic steps" count optimizer steps per 10M-step cell, while "joint fraction" is the fraction of training episodes receiving a joint update. These are implementation counters, not wall-clock cost or FLOPs.

Table 7: Available 10M spread resource fields. Ranges are over cells with a recorded field; n is the number of recorded cells out of five. The table is an audit of comparability, not a matched-cost experiment.

Protocol	Joint fraction	Training critic steps	Probe critic steps
Uniform	0 ($n = 3$)	1,200,000 ($n = 3$)	0 ($n = 3$)
Joint	1 ($n = 5$)	400,000 ($n = 5$)	0 ($n = 5$)
Diagnostic gate	0.1417–0.4240 ($n = 5$)	860,830–1,086,632 ($n = 5$)	2,400,000 ($n = 5$)
Random gate	0.4997–0.5007 ($n = 2$)	799,422–800,226 ($n = 2$)	2,400,000 ($n = 2$)

The diagnostic gate therefore has neither matched intervention frequency nor matched measured optimizer-step cost against the random gate, and the random gate's missing three cost rows prevent a complete resource comparison. The return comparisons remain paired by training seed where both endpoint summaries exist: each reported difference is method minus comparator, with $n = 5$ for the endpoint table, and its unadjusted two-sided Student-t interval is computed over those five paired seed differences. This pairing reduces seed-level variation but does not repair the frequency, cost, reset-seed overlap, or missing-summary limitations. The final curves do not provide policy checkpoints or historical source identity for every retried run. The available code therefore supports inspection but does not establish the exact source used by every historical execution.

Separate exploratory HARL cohort. Table 8 reports completed agent-local HAPPO spread seeds 6–10 separately from seeds 1–5. There are 100 checkpoints per branch, ten two-episode banks per checkpoint, and 20,000 returns in each five-seed cohort. Both cohorts were analyzed after the original registration and are reported separately as exploratory evidence, without retrospective pooling or claims of confirmation.

Table 8: Agent-local HAPPO spread, exploratory seeds 6–10. Displayed bounds are two-sided 95% seed-cluster t intervals on Q_W, transformed to RMS. Decisions compare one-sided 95% bounds on Q_W with $\varepsilon^2 = 1$; thus a displayed two-sided interval can include zero while that rule returns dependent. Coverage remains distribution-dependent with five seeds.

Window	Corrected RMS [interval]	One-sided decision
Full trajectory (100 checkpoints)	$9.693\ [1.161, 13.658]$	dependent
Tail (10 checkpoints)	$9.460\ [0, 13.805]$	dependent
Endpoint	$9.742\ [0, 14.879]$	undetermined

The full and tail one-sided lower bounds on Q_W are 22.846 and 11.862; the endpoint lower bound is negative. The resulting classifications apply to this cohort under the stated coverage assumptions,

not to all HAPPO runs. Native-global-RNG rows remain separate because reversal also changes random-stream assignment.

These decisions are pointwise in the registered window. A sensitivity analysis allocating $0.05/3$ to each of the full, tail, and endpoint windows makes all three decisions undetermined in both five-seed cohorts; it does not repair the distributional assumptions of the t bounds. In seeds 6–10, the pointwise full and tail dependence decisions also become undetermined after removing seed 6, 7, or 9, but remain dependent after removing seed 8 or 10. These overlapping leave-one-seed-out analyses quantify sensitivity, not independent replications or grounds for discarding a seed.

Detector controls. The magnitude detector predicts effects of a frozen surrogate utility, not environment return. A retrospective comparison fits environment-specific training-stage effects on discovery data, with or without the projection magnitude. On the previously inspected held-out clusters, adding magnitude reduces mean squared log-transformed prediction error in 15/15 clusters. This stage-control analysis is exploratory and is not a new confirmation or evidence of return-based scheduling. Its data, predictions, and fitting script are included under the stage-baseline entry in the evidence index.

Artifact limits. The offline verification script recomputes the key tables without creating environments or invoking training. Saved short-budget policies support re-evaluation, but their model and optimizer state alone is insufficient for exact continuation without environment and RNG state. The external-dependency release retains HARL source hashes but excludes the modified implementation and intervention patch. A vanilla upstream checkout is insufficient for a full rerun; the historical extensions are additionally required. Source hashes do not establish the exact implementation used by every historical execution. Missing cost summaries, formal-grid policies, and historical execution identity remain unavailable.

H Smooth Local Analysis and Ranking Constructions

The local expansion applies to smooth, small steps in parameter space. It does not cover arbitrary complete-state updates: at fixed Adam momentum coefficients, memory and counters need not approach the identity as the learning rate vanishes, and clipping is not globally smooth. Let each block update be written as

$$U_i(\theta) = \theta + \eta F_i(\theta),$$

where F_i is the update vector field and η is the step size. For two blocks i, j, the difference between the two possible compositions is

$$U_j \circ U_i(\theta) - U_i \circ U_j(\theta) = \eta^2 \left(\nabla F_j(\theta) F_i(\theta) - \nabla F_i(\theta) F_j(\theta)\right) + O(\eta^3).$$

The second-order term is the Lie bracket $[F_i, F_j](\theta)$.

Proposition 3 (First non-commuting term)**.** *Assume F_i, F_j are twice continuously differentiable with bounded first and second derivatives on a neighborhood containing the update segments, and take $0 \le \eta \le \eta_0$ small enough that those segments remain in that neighborhood. Then*

$$\|U_j \circ U_i(\theta) - U_i \circ U_j(\theta)\| \le \eta^2 \|[F_i, F_j](\theta)\| + C\eta^3$$

for a local constant C.

Proof. Taylor's theorem and the bounded derivatives give $F_j(\theta + \eta F_i(\theta)) = F_j(\theta) + \eta \nabla F_j(\theta) F_i(\theta) + O(\eta^2)$, and the analogous expansion with i, j exchanged, uniformly for the stated step-size range. Multiplying by η and subtracting cancels the first-order terms. The remaining difference is $\eta^2 [F_i, F_j](\theta) + O(\eta^3)$, and the triangle inequality gives the bound with a local constant C. □

Proposition 4 (Utility-projected local ordering)**.** *If J is twice continuously differentiable and $w_{ij}(\theta) = \nabla J(\theta)^\top [F_i, F_j](\theta)$, then*

$$J(U_j U_i(\theta)) - J(U_i U_j(\theta)) = \eta^2 w_{ij}(\theta) + O(\eta^3),$$

$$J(U_\pi(\theta)) = C(\theta, \eta) + \frac{\eta^2}{2} \sum_{a<b} w_{i_a i_b}(\theta) + O(\eta^3).$$

Here $\pi = (i_1, \ldots, i_m)$ and C is protocol-independent; maximizing the signed pair sum is therefore locally optimal to second order.

Proof. Taylor expansion of J around θ shows that reversing an adjacent pair cancels the common first-order and symmetric second-order terms, leaving $\eta^2 \nabla J(\theta)^\top [F_i, F_j](\theta) + O(\eta^3)$. Summing the pair contributions over a fixed-length permutation yields the displayed pair sum; all permutation-independent terms are collected in C. □

The weights depend on state, data, and utility and may form directed cycles, so this is a weighted linear-ordering problem, not a universal scalar priority.

The same expansion bounds the error in predicting effect magnitude without requiring a stable sign.

Proposition 5 (Commutator-magnitude ranking)**.** *At checkpoint k, let the paired local utility effect satisfy $\Delta_k = \eta^2 w_k + r_k$ with $|r_k| \le C\eta^3$. Then*

$$\big||\Delta_k| - \eta^2 |w_k|\big| \le C\eta^3.$$

Hence $\eta^2(|w_a| - |w_b|) > 2C\eta^3$ implies $|\Delta_a| > |\Delta_b|$.

Proof. The reverse triangle inequality gives $\big||\Delta_k| - \eta^2|w_k|\big| \le |r_k| \le C\eta^3$. Applying this bound to checkpoints a, b gives the ranking condition. □

Magnitude ranking distinguishes large from small local order effects. It neither identifies the better order nor certifies a long-horizon return bound.

Affine-system check. We also test Proposition 4 on smooth four-block affine systems using all 24 permutations and 500 fresh systems in each of an exchangeable and a shared-bias regime. At $\eta = 0.1$ and one update sweep, pairwise sign accuracy is 98.7% and 98.5%, permutation-level Spearman correlation is 0.992 in both regimes, and normalized selected-order regret is 0.0024 and 0.0011. Pair preferences contain directed three-cycles in 61–64% of systems, directly ruling out a general scalar priority rule despite the existence of a locally optimal permutation. The approximation has the predicted boundary: at $\eta = 0.75$, rank correlation falls from about 0.70 after one sweep to about 0.25 after four sweeps.

H.1 Ranking Separation

Proposition 6 (Protocol-dependent rankings can be ill-specified)**.** *Let algorithm A be evaluated by J_A and algorithm B have two protocols π_1, π_2 with*

$$J(\theta_B^{\pi_1}) > J_A > J(\theta_B^{\pi_2}).$$

Then the qualitative ranking between A and B is not determined by the algorithm names alone; it is determined by the protocol choice.

Proof. Under protocol π_1, B ranks above A; under π_2, B ranks below A. Both runs use the same named algorithm B, so any comparison that omits π is under-specified. □

Theorem 2 (Constructive ranking separation)**.** *There exists a two-block finite-budget training system with identical initialization, identical update blocks, identical budget, and identical evaluation functional such that changing only the protocol reverses the ranking against a fixed baseline.*

Proof. Let $\theta = (x, y) = (0, 0)$, $U_1(x, y) = (x + 1, y)$, $U_2(x, y) = (x, y + 2x - 1)$, and $J(x, y) = y$. Then $J(U_2 U_1 \theta) = 1$ whereas $J(U_1 U_2 \theta) = -1$. Against the fixed baseline $J_A = 0$, changing only the order reverses the ranking while leaving the initialization, update blocks, budget, and evaluation functional fixed. □