THE GAP BETWEEN SHAPLEY-BASED CAUSAL ATTRIBUTION AND POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Counterfactual credit assignment can reveal which decisions matter in a trajectory, but good attribution does not guarantee a good policy update. We study this gap through Counterfactual Shapley credit. A later action can change Shapley credit assigned to an earlier decision, yet its return-to-go update never reads that earlier score. We show that even exact, credit-conserving Shapley attribution with correct within-state action rankings can induce a strict local decrease in the original return. This exposes an *update-visibility gap*: action-dependent credit can be present in the attribution but absent from the update of the action that changes it. We introduce RECAP, which preserves the original Shapley allocation while adding the omitted cross-time effect to the corresponding learning signal. For Counterfactual Shapley, we derive a tractable form of this effect and show when compensation yields a better local update, including when the original Shapley update is already beneficial. Across three tasks with distinct temporal dependencies, RECAP consistently improves over CSCA. Independent local diagnostics show that the predicted compensation gain tracks the observed return gain, and the advantage persists with larger attribution budgets, learned policy stochasticity, and noisy rewards.

1 INTRODUCTION

Long-horizon credit assignment often follows a common pattern: a trajectory-level outcome is transformed into per-step credit, and that credit is then used to train the policy (Pignatelli et al., 2024). Methods such as RUDDER (Arjona-Medina et al., 2019) and RRD (Ren et al., 2022) construct these signals through return decomposition or redistribution, while Counterfactual Shapley Credit Assignment (CSCA) (Li et al., 2026a) estimates each decision's contribution through counterfactual action replacement. These methods differ in how they assign credit. We ask a separate question that comes afterward: even when the credit assignment itself is reasonable, is it necessarily suitable as a stepwise policy-learning signal? In other words, there is an interface between credit assignment and policy optimization that deserves to be studied in its own right.

We focus on Counterfactual Shapley because it provides a particularly clean and stringent setting for this question. It has explicit counterfactual semantics, can be computed exactly in controlled constructions, and satisfies natural attribution properties such as credit conservation. Its retrospective structure also lets us trace how one action changes credit assigned to other time steps. We find that a later action can change the Shapley credit assigned to an earlier decision, while the later action's own return-to-go update never reads that earlier score. Shapley efficiency preserves total credit, but does not justify discarding cross-time terms in the policy gradient. We examine this distinction in relation to the gradient-equivalence claim of Li et al. (2026a). Even exact attribution that conserves credit relative to its reference and preserves action-value rankings within every reachable state can induce a strict local descent direction for the original expected return when used in a return-to-go update.

Analyzing this failure reveals a broader structure. For a class of retrospective credit signals that conserve return relative to an action-independent reference and are used through standard return-to-go updates, the missing information is precisely the action-dependent component of credit assigned to earlier steps. We call the resulting property update visibility: whether credit associated with an action is actually available to the update of that action. Guided by this principle, we introduce Retrospective Effect Compensation for Attribution-based Policy Learning (RECAP) for Counterfactual Shapley.

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.

RECAP preserves the original Shapley allocation and restores the cross-time effect omitted by the forward update. Under explicit structural conditions, we derive a directly computable form of this correction that requires no additional counterfactual sampling. We also characterize when the added effect improves the original task return, and show that RECAP can help even when the original Shapley update is already beneficial.

Across three tasks with different forms of temporal dependence, RECAP achieves higher mean final return than directly using Counterfactual Shapley credit for policy learning. Local analyses further show that the additional gain predicted by the compensation term matches the observed small-step return improvement, including cases in which the original Shapley update already improves the policy. The advantage also persists, within the tested settings, under larger attribution-estimation budgets, learnable policy stochasticity, and reward noise.

Our contributions are:

- We identify a gap between Counterfactual Shapley attribution and policy optimization. Even exact, credit-conserving, rank-preserving attribution can induce a harmful policy update.
- We derive update visibility from this failure and use it to develop RECAP. Update visibility captures whether action-dependent credit is available to the corresponding policy update; RECAP restores the missing effect for Counterfactual Shapley.
- We provide both training-level and mechanism-level evidence. Across three tasks, RECAP improves direct Shapley-based policy learning, while independent local analyses link the added effect to actual return gains.

2 Cross-Time Effects and Compensation

2.1 A Missing Effect in Counterfactual Shapley

Consider a deterministic two-step task with a fixed initial state, $A_1, A_2 \in \{0, 1\}$, $S_2 = A_1$, and $Y = 100A_1A_2$. The policy and reference policy choose either action with probability $1/2$ at every state.

Counterfactual Shapley Credit Assignment (CSCA) forms coalition values $v(S)$ by retaining subsets of factual actions and drawing the remaining actions from a reference policy. The Shapley rule averages each action's marginal contribution across coalition contexts. By efficiency, $\sum_t \phi_t = Y - J_{\text{ref}}$, where $Y = v(\{1, \ldots, T\})$ and $J_{\text{ref}} = v(\varnothing)$.

(a) Exact attribution and the changing marginal

$J_{\text{ref}} = 25, \quad m_1^{\emptyset} = 25$

(A_1, A_2)	$Y - J_{\text{ref}}$	$m_1^{\{2\}}$	ϕ_1	ϕ_2
$(1, 1)$	75	50	37.5	37.5
$(1, 0)$	-25	0	12.5	-37.5

$\phi_1 = \frac{1}{2}(m_1^{\emptyset} + m_1^{\{2\}})$.

(b) Environment and attribution

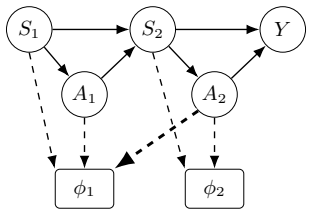


Figure 1: A later action changes earlier Shapley credit. (a) With A_1 fixed, changing A_2 changes $m_1^{\{2\}}$ from 0 to 50, and hence ϕ_1 from 12.5 to 37.5. (b) Solid arrows show environment dependencies; dashed arrows show retrospective attribution dependencies. The bold edge marks $A_2 \to \phi_1$.

Figure 1 shows the key dependence: changing A_2 changes the marginal contribution used to compute ϕ_1, even though (S_1, A_1) is fixed.

Using these scores in a return-to-go update gives

$$\widehat{U}_\phi = z_1(\phi_1 + \phi_2) + z_2\phi_2, \qquad z_t = \nabla_\theta \log \pi_\theta(A_t \mid H_t).$$

The A_1 update sees both scores, while the A_2 update sees only ϕ_2. Thus, the A_2-dependent part of ϕ_1 is absent from the A_2 update.

This is retrospective attribution, not backward environmental causation. The optimization issue is to use the A_2-dependent information without reallocating ϕ_1 away from A_1. We next characterize this

missing information beyond the example. The counterfactual rule and full calculation are given in Appendix A.1.

2.2 What Does a Stepwise Update Miss?

Section 2.1 suggests that past credit is not itself the problem. What matters is whether that credit still depends on the current action.

Consider retrospective scores $\psi_1, \ldots, \psi_T$ computed after a complete trajectory. At a fixed policy, let $z_t = \nabla_\theta \log \pi_\theta(A_t \mid H_t)$ and $P_t = \sum_{j<t} \psi_j$. Define

$$D_t = \mathbb{E}[P_t \mid H_t, A_t] - \mathbb{E}[P_t \mid H_t]. \tag{1}$$

The return-to-go update never uses P_t. The quantity D_t isolates the part of this past credit that still changes with the current action at the same history.

Proposition 1 (Update visibility)**.** *Under the usual score-function regularity conditions, the omitted past-credit contribution satisfies*

$$M_\psi := \mathbb{E}\left[\sum_{t=1}^{T} z_t P_t\right] = \mathbb{E}\left[\sum_{t=1}^{T} z_t D_t\right]. \tag{2}$$

The identity follows from $\mathbb{E}[z_t \mid H_t] = 0$. Any component of P_t determined by the pre-action history acts as a baseline and vanishes in expectation. Only the component that varies with A_t remains. A forward update does not miss all past credit; it misses the action-dependent part of that credit.

Ordinary rewards explain why this issue is easy to overlook. Rewards received before A_t are already determined when A_t is chosen, so their corresponding D_t is zero. Retrospective scores need not have this property: an earlier score can still depend on a later action, as ϕ_1 depends on A_2 in Section 2.1.

We call this property *update visibility*. When the credit rule also conserves return, the same missing contribution has a direct relation to the original objective.

Corollary 1 (Relation to the original return)**.** *Suppose $\sum_{j=1}^{T} \psi_j = Y - c$, where c is a fixed action-independent reference. If $h_\psi := \mathbb{E}[\sum_{t=1}^{T} z_t \sum_{j=t}^{T} \psi_j]$ denotes the expected return-to-go update, then*

$$\nabla J(\theta_0) = h_\psi + M_\psi = h_\psi + \mathbb{E}\left[\sum_{t=1}^{T} z_t D_t\right]. \tag{3}$$

A nonzero D_t identifies action-dependent information omitted by the forward update, but does not by itself imply a harmful policy update. We next return to Counterfactual Shapley and show that this visibility gap can reduce the original expected return even under strong attribution properties.

2.3 Strong Attribution Properties Are Not Enough

Section 2.2 identifies the action-dependent credit omitted by a forward update. A nonzero visibility gap does not by itself imply harm, since omissions across states may cancel. We now show that, for Counterfactual Shapley, they need not.

Theorem 1 (Strong attribution properties do not prevent harmful updates)**.** *For every finite horizon $T \geq 2$, there exist an MDP, a differentiable shared-parameter policy π_θ, and a parameter point θ_0 such that the exact Counterfactual Shapley scores computed with the frozen reference policy π_{θ_0} satisfy*

$$\sum_{t=1}^{T} \phi_t = Y - J_{\mathrm{ref}}, \qquad J_{\mathrm{ref}} = \mathbb{E}_{\pi_{\theta_0}}[Y],$$

and, at every reachable state, the conditional mean Shapley scores rank actions in the same order as $Q_t^{\pi_{\theta_0}}$.

Nevertheless, the expected return-to-go update $h_\phi = \mathbb{E}_{\pi_{\theta_0}}\left[\sum_{t=1}^{T} z_t \sum_{j=t}^{T} \phi_j\right]$ satisfies

$$J(\theta_0 + \eta h_\phi) < J(\theta_0)$$

for all sufficiently small $\eta > 0$, where $z_t = \nabla_\theta \log \pi_\theta(A_t \mid H_t)|_{\theta=\theta_0}$ and the attribution scores are held fixed during differentiation.

The failure does not require incorrect local attribution. It appears when shared policy parameters combine learning signals across states. If the visibility gap removes different amounts of action-dependent feedback in different states, it changes their relative influence on the same parameter update. The combined update can then reduce return even though the action ranking remains correct within every state.

Strong attribution properties do not guarantee a beneficial policy update. The construction and proof are given in Appendix A.2. We next use the coalition structure of Counterfactual Shapley to derive a computable form for the omitted effect.

3 RECAP: Cross-Time Effect Compensation

3.1 A Closed Form for the Missing Shapley Effect

Section 2.2 identified D_t as the action-dependent past credit omitted by a forward update. We now show that Counterfactual Shapley admits a closed form for D_t when its coalition values have a first-replacement structure.

Let

$$\bar{r}_t = \gamma^{t-1} r_t, \qquad W_t = \mathbb{E}_\pi\left[\sum\nolimits_{k=t}^{T} \bar{r}_k \,\middle|\, H_t\right], \qquad W_{T+1} = 0,$$

and define the value residual $\Delta_t = \bar{r}_t + W_{t+1} - W_t$.

For any nonempty replacement coalition Z, let $m = \min Z$. We say the coalition values satisfy *first-replacement sufficiency* if

$$\mathbb{E}_{\mathrm{cf}}[Y^Z \mid \tau] = \sum\nolimits_{k<m} \bar{r}_k + W_m. \tag{4}$$

Under this condition, the expected counterfactual return depends only on where the first replacement occurs, not on later members of Z.

Theorem 2 (Residual form of Counterfactual Shapley)**.** *If the coalition values satisfy first-replacement sufficiency, then the exact Counterfactual Shapley scores satisfy*

$$\phi_j = \sum_{k=j}^{T} \frac{\Delta_k}{k}, \qquad D_t = \frac{t-1}{t}\mathbb{E}_\pi[\Delta_t \mid H_t, A_t]. \tag{5}$$

The first identity follows from symmetry. The residual Δ_k is shared across the first k decision positions, so Shapley assigns each a contribution of Δ_k/k. For the current residual Δ_t, $t-1$ of these t shares lie before the current action. These are exactly the shares that can depend on A_t but are absent from its forward update, which yields the factor $(t-1)/t$.

First-replacement sufficiency is only needed for this Shapley-specific closed form. The update-visibility definition of D_t in Section 2.2 does not require it. Deterministic dynamics, an effective first replacement, and no later remerging with the factual branch give one sufficient set of conditions for Eq. (4); they are not required by the general definition.

The full coalition decomposition and sufficient-condition proof are given in the appendix. We next use the same suffix-value estimates to construct both the Shapley signal and its missing-effect compensation.

3.2 RECAP Updates

Theorem 2 lets us estimate the Shapley signal and its missing effect from the same value residuals. Fix the sampling and reference policy as $\pi_0 = \pi_{\theta_0}$. Using K suffix rollouts, let $\widehat{W}_t$ be an unbiased estimate of W_t and set $\widehat{\Delta}_t = \bar{r}_t + \widehat{W}_{t+1} - \widehat{W}_t$. We then compute $\widehat{\phi}_t = \frac{\widehat{\Delta}_t}{t} + \widehat{\phi}_{t+1}, \quad \widehat{D}_t = \frac{t-1}{t}\widehat{\Delta}_t$, with $\widehat{\phi}_{T+1} = 0$. Under unbiased suffix estimation, $\mathbb{E}[\widehat{D}_t \mid H_t, A_t] = D_t$.

The direct attribution signal accumulates current and future Shapley scores, $\widehat{L}^{\phi}_t = \widehat{\phi}_t + \lambda \widehat{L}^{\phi}_{t+1}$, $\widehat{L}^{\phi}_{T+1} = 0$. RECAP adds the estimated effect on earlier credit:

$$\widehat{L}^{\mathrm{RECAP}}_t = \widehat{L}^{\phi}_t + \widehat{D}_t. \tag{6}$$

Algorithm 1 RECAP: one policy-gradient update

Require: Policy π_{θ_0}, batch size B, suffix budget K, trace parameter λ, step size η

Set $\pi_0 = \pi_{\theta_0}$ and sample B trajectories from π_0.
Estimate $\widehat{W}_{i,t}$ using K suffix rollouts under π_0.
Set $\widehat{W}_{i,T+1} = \widehat{\phi}_{i,T+1} = \widehat{L}^{\phi}_{i,T+1} = 0$.
for $t = T, \dots, 1$ **do**
    $\widehat{\Delta}_{i,t} \leftarrow \bar{r}_{i,t} + \widehat{W}_{i,t+1} - \widehat{W}_{i,t}$
    $\widehat{\phi}_{i,t} \leftarrow \widehat{\Delta}_{i,t}/t + \widehat{\phi}_{i,t+1}$
    $\widehat{L}^{\phi}_{i,t} \leftarrow \widehat{\phi}_{i,t} + \lambda \widehat{L}^{\phi}_{i,t+1}$
    $\widehat{D}_{i,t} \leftarrow \frac{t-1}{t}\widehat{\Delta}_{i,t}, \quad \widehat{L}^{\mathrm{RECAP}}_{i,t} \leftarrow \widehat{L}^{\phi}_{i,t} + \widehat{D}_{i,t}$
end for
$\widehat{U}_{\mathrm{RECAP}} \leftarrow \frac{1}{B}\sum_{i=1}^{B}\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta(A_{i,t} \mid H_{i,t})\big|_{\theta=\theta_0} \widehat{L}^{\mathrm{RECAP}}_{i,t}$
return $\theta_0 + \eta \widehat{U}_{\mathrm{RECAP}}$

Thus, RECAP keeps the original Shapley allocation and changes only how that attribution enters the policy update. In short, RECAP changes the learning signal, not the attribution.

For a batch of B factual trajectories, the resulting stochastic update is

$$\widehat{U}_{\mathrm{RECAP}} = \frac{1}{B}\sum_{i=1}^{B}\sum_{t=1}^{T} z_{i,t}\widehat{L}^{\mathrm{RECAP}}_{i,t}, \tag{7}$$

where $z_{i,t}$ is evaluated at θ_0 and all sampled labels are held fixed. Algorithm 1 summarizes the update. CSCA and RECAP reuse the same suffix estimates, so compensation requires no additional suffix rollouts at matched K.

Expected update. The compensation also restores the missing contribution identified in Section 2.2 at the level of the expected update.

Corollary 2 (Expected RECAP update)**.** *Under Theorem 2 and standard score-function regularity conditions, suppose* $\mathbb{E}_{\mathrm{aux}}[\widehat{W}_t \mid \tau] = W_t$. *Then, for any* $\lambda \in [0, 1]$ *and finite* $K \geq 1$,

$$\mathbb{E}[\widehat{U}_{\mathrm{RECAP}}] = h_\phi + M_\phi = \nabla J(\theta_0). \tag{8}$$

The first equality states that RECAP implements the update-visibility correction from Section 2.2; the second follows from Shapley credit conservation. Future residuals have zero conditional mean, so λ changes the realized attribution signal but not its expected score update. Finite K likewise affects estimation variance, not the mean. The full proof is given in the appendix.

Relation to reward-based learning. Equal expected updates do not make RECAP equivalent to reward-based learning. Under the conditions above, RECAP and a reward-based policy-gradient estimator have the same expectation at the sampling policy, but they use different trajectory-level signals and generally produce different finite-batch updates. Whether this difference leads to a better finite policy step is a separate question, which we study next.

3.3 When Does Compensation Help?

The expected-update result in Section 3.2 does not tell us whether compensation improves a finite stochastic step. We therefore compare CSCA and RECAP from the same policy θ_0. Write $\widehat{U}_{\mathrm{RECAP}} = \widehat{U}_\phi + \widehat{U}_D$, and define $g = \nabla J(\theta_0), \quad m_D = \mathbb{E}[\widehat{U}_D], \quad \kappa_D = g^\top m_D$. The quantity κ_D is the first-order return contribution added by compensation.

For a common step size η, let

$$\mathcal{G}(\eta) = \mathbb{E}\Big[J(\theta_0 + \eta\widehat{U}_{\mathrm{RECAP}}) - J(\theta_0 + \eta\widehat{U}_\phi)\Big].$$

Theorem 3 (Local gain from compensation)**.** *Suppose* J *has an* L*-Lipschitz gradient and* $\nu = \mathbb{E}\Big[\|\widehat{U}_{\mathrm{RECAP}}\|^2 + \|\widehat{U}_\phi\|^2\Big] < \infty$. *Then*

$$|\mathcal{G}(\eta) - \eta\kappa_D| \leq \frac{L\eta^2}{2}\nu. \tag{9}$$

Hence, if $\kappa_D > 0$, then $\mathcal{G}(\eta) > 0$ for all sufficiently small $\eta > 0$.

Thus, a positive first-order contribution from compensation yields a higher expected return after a sufficiently small step. This result does not require $\widehat{U}_{\mathrm{RECAP}}$ to be an unbiased gradient estimator; it compares the two stochastic updates directly. The proof follows from smoothness and is given in the appendix.

The benefit also does not require the original Shapley update to be harmful.

Corollary 3 (Improving an ascent update)**.** *Let $h_\phi = \mathbb{E}[\widehat{U}_\phi]$. Under Theorem 3, if $g^\top h_\phi > 0$ and $\kappa_D > 0$, then for all sufficiently small $\eta > 0$,*

$$\mathbb{E}[J(\theta_0 + \eta\widehat{U}_{\mathrm{RECAP}})] > \mathbb{E}[J(\theta_0 + \eta\widehat{U}_\phi)] > J(\theta_0). \tag{10}$$

RECAP can therefore strengthen an attribution update that already improves return; it is not limited to repairing harmful updates. Theorem 3 also gives two quantities that can be tested directly: the predicted first-order gain κ_D and the observed paired return difference $\mathcal{G}(\eta)$. Section 4.3 evaluates both.

4 Experiments

4.1 Experimental Setup

We evaluate on three controlled tasks with different forms of temporal dependence. In RoundTrip, early actions affect later movement through position and velocity. In LaggedLink, actions shape future decisions through delayed rate response and persistent queue backlog. In Information Allocation, measurement choices change the uncertainty carried into later stages. These tasks let us test the same question under distinct temporal structures: whether restoring omitted cross-time effects improves policy learning from retrospective credit. Full task definitions and task-specific justifications for applying the closed form in Section 3.1 are given in Appendix A.4.

Our primary comparison is among PPO, CSCA, and RECAP. PPO (Schulman et al., 2017) learns from the original rewards with a learned critic and GAE (Schulman et al., 2018). CSCA denotes direct policy learning from Counterfactual Shapley credit within our matched actor-training framework. RECAP starts from the same Shapley estimates and adds only the cross-time compensation derived in Section 3. CSCA and RECAP therefore isolate the effect of compensation: they share the factual data, Shapley estimator, suffix-sampling budget, and actor-training setup. PPO provides the reward-based reference. We also report QCA, HQCA (Mesnard et al., 2023), H-DICE (Velu et al., 2023), and RUDDER (Arjona-Medina et al., 2019) under the same actor-training framework.

Section 3 analyzes the learning signal under an ordinary stochastic policy-gradient step. In full training, we use the same CSCA and RECAP labels inside a standard multi-update PPO/Adam actor loop. Each round uses four actor updates while keeping the learning labels and old action log-probabilities fixed, so the attribution and compensation construction itself is unchanged. The core methods use matched actor architectures, initialization, exploration parameterization, optimizer, clipping threshold, and actor-update count. We train five seeds for 1,200 rounds and use the original-task return at the fixed training budget as the primary metric. Full hyperparameters and evaluation protocols are given in Appendix A.4.

4.2 Policy Learning with Counterfactual Credit

Figure 2 compares three ways to train the same policy class. PPO learns from the original rewards, CSCA learns directly from Counterfactual Shapley credit, and RECAP adds the cross-time compensation from Section 3 to the same Shapley signal. Direct attribution is not automatically better than reward-based learning: CSCA finishes below PPO in both RoundTrip and LaggedLink. RECAP changes this pattern. It improves over CSCA in all three tasks, with the gap emerging during training and persisting to the final checkpoint.

The CSCA–RECAP comparison isolates the effect of compensation. The two methods share the Counterfactual Shapley estimator, factual data, suffix-sampling budget, and actor-training setup; RECAP differs only by the added cross-time term. At the fixed training budget, RECAP improves the

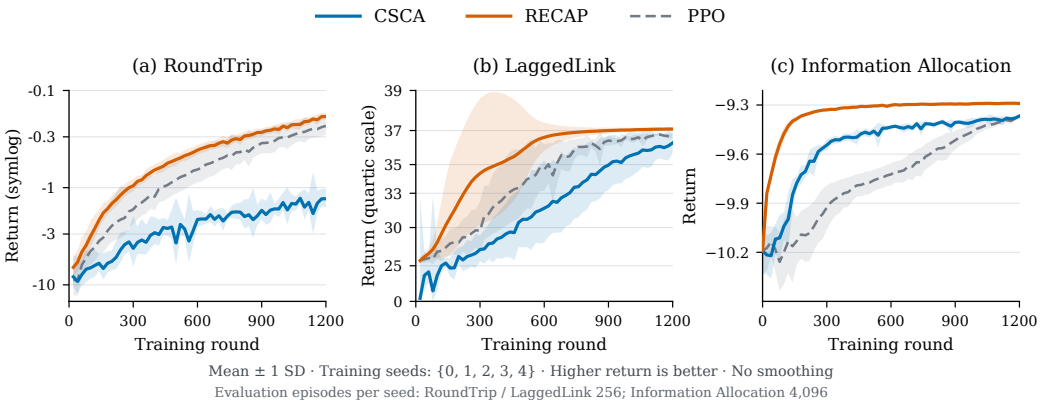


Figure 2: Learning curves for CSCA, RECAP, and PPO. Shading denotes one sample standard deviation across five training seeds. RoundTrip and LaggedLink use symmetric-logarithmic and quartic vertical scales, respectively; tick labels retain the original return units.

Table 1: Original-task return after 1,200 training rounds (mean ± sample standard deviation over training seeds $\{0, 1, 2, 3, 4\}$), evaluated using 4,096 episodes per seed. Higher is better; bold indicates the highest mean in each task.

Method	RoundTrip	LaggedLink	Information Allocation
PPO	-0.234 ± 0.065	36.693 ± 0.133	-9.375 ± 0.012
QCA	-0.217 ± 0.014	37.017 ± 0.082	-9.315 ± 0.009
HQCA	-0.226 ± 0.020	37.044 ± 0.076	-9.311 ± 0.011
H-DICE	-0.192 ± 0.008	36.174 ± 0.266	-9.835 ± 0.096
RUDDER	-1.089 ± 0.303	35.090 ± 2.661	-9.705 ± 0.143
CSCA	-1.306 ± 0.245	36.316 ± 0.588	-9.367 ± 0.010
RECAP	$\mathbf{-0.186 \pm 0.012}$	$\mathbf{37.084 \pm 0.100}$	$\mathbf{-9.291 \pm 0.003}$

mean final return over CSCA by 1.120 in RoundTrip, 0.769 in LaggedLink, and 0.077 in Information Allocation. These results show that how Shapley credit enters the policy update matters, not only how the credit itself is assigned.

PPO provides a complementary reference because it learns from the original rewards rather than retrospective attribution. Together, PPO, CSCA, and RECAP separate direct reward learning, direct Shapley learning, and compensated Shapley learning. CSCA does not consistently outperform PPO, whereas RECAP reaches a higher final mean than both in all three tasks. Table 1 places these results alongside QCA, HQCA, H-DICE, and RUDDER under the matched actor-training setup.

These full-training results establish the benefit of compensation, but do not yet show whether the gain follows the local effect predicted in Section 3. We test that prediction next.

4.3 Testing the Local-Gain Prediction

Section 3.3 predicts that, for a sufficiently small step, $\mathcal{G}(\eta) = \eta\kappa_D + O(\eta^2)$, where $\eta\kappa_D$ is the first-order return gain contributed by compensation. We test this prediction at CSCA and RECAP checkpoints from rounds 600 and 1,200 in all three tasks. From each frozen policy, we construct both updates from the same starting point. We estimate $\eta\kappa_D$ with an independent return-gradient reference and measure $\mathcal{G}(\eta)$ with separate paired evaluation rollouts.

Figure 3 compares the predicted and observed gains directly. Their means are positive in all 12 task–checkpoint groups, and their magnitudes broadly track each other. For example, at the RoundTrip CSCA/1,200 checkpoints, the predicted gain is 2.683×10^{-3}, compared with an observed RECAP–CSCA gain of 2.740×10^{-3}. Seed-level variation remains substantial at some checkpoints, so the

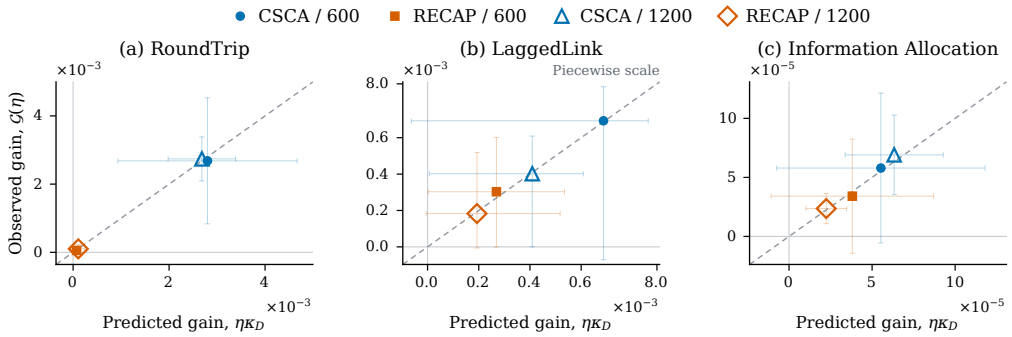


Figure 3: Predicted versus observed local gain from compensation. Each point represents one frozen-policy checkpoint group. The horizontal axis shows the first-order prediction $\eta\kappa_D$ from Section 3.3; the vertical axis shows the paired return difference $\mathcal{G}(\eta) = \Delta J_{\mathrm{RECAP}} - \Delta J_{\phi}$ (RECAP minus CSCA). The dashed diagonal marks $y = x$. Error bars show one sample standard deviation across five training seeds. In (b), both axes use a continuous piecewise-linear scale, with outer intervals compressed $25\times$ relative to $[0, 0.6] \times 10^{-3}$.

prediction is not expected to match every point exactly. The consistent pattern is that a positive independently estimated compensation contribution corresponds to a positive local return gain.

The gain is not limited to cases where the CSCA update is harmful. In 11 of the 12 groups, CSCA already increases return on average, yet RECAP improves it further: $\Delta J_{\mathrm{RECAP}} > \Delta J_{\mathrm{CSCA}} > 0$. Compensation can therefore strengthen an attribution update that is already useful, as predicted by Corollary 3.

Additional equal-displacement and multi-update PPO/Adam probes show the same qualitative pattern in RoundTrip; full results are given in Appendix A.4.

4.4 Robustness Beyond the Main Setting

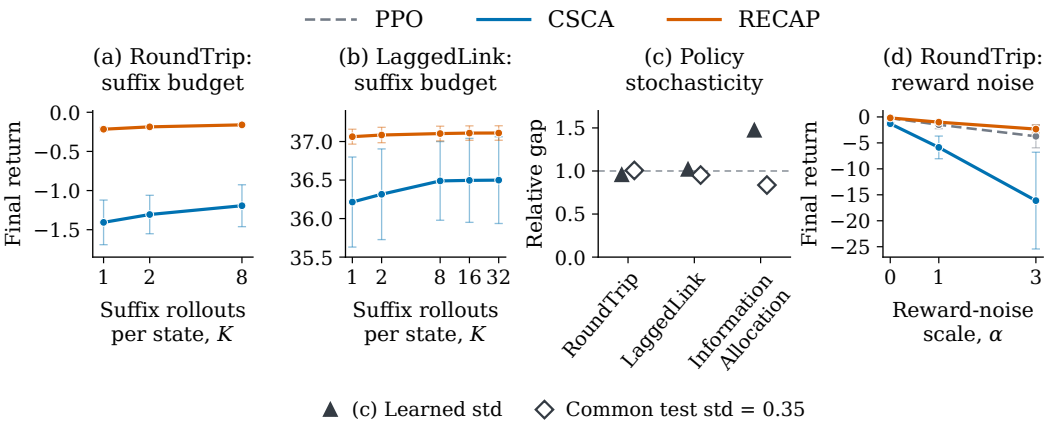


Figure 4: Robustness beyond the main setting. (a–b) Increasing the suffix-sampling budget does not remove the RECAP–CSCA gap. (c) The gain persists with learned policy stochasticity and under a common test standard deviation of 0.35. Points show mean RECAP–CSCA gaps divided by the corresponding main fixed-std mean gap, without error bars; the dashed line marks a relative gap of 1. (d) RECAP remains above CSCA under increasing reward noise. Error bars in (a), (b), and (d) denote one sample standard deviation across five training seeds.

We first increase the suffix budget K to test how the compensation gain changes with estimation precision. Figure 4(a–b) shows that both methods improve with more suffix samples, while the mean

final-return gap narrows. In RoundTrip, the RECAP–CSCA gap decreases from 1.190 at $K = 1$ to 1.033 at $K = 8$. In LaggedLink, it decreases from 0.769 at $K = 2$ to 0.612 at $K = 32$, with little change beyond $K = 8$. More accurate Shapley estimates therefore improve direct attribution learning, while compensation continues to provide an additional gain at every tested budget.

The effect also persists when policy stochasticity is learned rather than fixed. We replace the fixed standard deviation in the main experiments with trainable log-standard-deviation parameters and optimize them together with the policy mean. RECAP remains above CSCA in all three tasks. We then evaluate the same learned policies at a common test standard deviation of 0.35; the RECAP–CSCA gap remains positive. Figure 4(c) shows both comparisons, indicating that the gain is not tied to the fixed exploration scale used in the main setting.

We next perturb the reward signal itself. In RoundTrip, we add independent zero-mean reward noise during training and evaluate the resulting policies on the original task. RECAP remains above CSCA at every tested noise level and loses less return relative to its noise-free performance. At the largest tested noise scale, $\alpha = 3$, mean return drops by 2.155 for RECAP, compared with 3.493 for PPO and 14.812 for CSCA, each measured against its own zero-noise result.

At matched K, CSCA and RECAP use the same factual interactions and suffix rollouts; RECAP reuses the residual estimates already computed for the Shapley signal. At $K = 32$ in LaggedLink, mean CPU time over five seeds is 867.0 seconds for CSCA and 869.8 seconds for RECAP, a difference of about 0.3%. These timings cover the training loop, excluding monitoring and evaluation. Compensation thus adds no suffix sampling and has little measured computational overhead. Full sensitivity results and computation accounting are given in Appendix A.4.

5 Related Work

Prior work constructs informative per-step credit from delayed outcomes through temporal modeling (Hung et al., 2018; Raposo et al., 2021; Ferret et al., 2020), return redistribution (Gangwani et al., 2020; Zhang et al., 2023; Zhu et al., 2023; Xiao & Yu, 2025), and hindsight or counterfactual estimation (Harutyunyan et al., 2019; Rauber et al., 2019; Mesnard et al., 2021; Meulemans et al., 2023), including quantile and density-ratio methods (Mesnard et al., 2023; Velu et al., 2023). We study how retrospective credit enters policy updates.

Related credit constructions span multi-agent learning (Foerster et al., 2018; Wang et al., 2020; Li et al., 2021; Kapoor et al., 2024), language-model training (Cao et al., 2025; Li et al., 2026b; Khandoga et al., 2026; Deng et al., 2024; Kazemnejad et al., 2025; Meng & Chen, 2026; Chen et al., 2026; Yuan et al., 2025; Cheng et al., 2025), and language-model-assisted feedback (Qu et al., 2025; Chen et al., 2025). Our closest work is CSCA (Li et al., 2026a), whose Counterfactual Shapley attribution we retain. We ask a complementary question: are action-dependent changes in retrospective credit visible to the forward stepwise update of the action that changes them?

6 Conclusion

We study a missing link between retrospective credit assignment and policy optimization. Using Counterfactual Shapley as our main setting, we show that a later action can change credit assigned to an earlier decision while its own forward update never reads that change. Strong attribution properties do not prevent this failure: even exact, credit-conserving scores with correct within-state action rankings can still produce a harmful policy update. We capture this issue through *update visibility* and introduce RECAP, which preserves the original Shapley attribution while adding the omitted cross-time effect to the corresponding learning signal. Across full training and independent local diagnostics, compensation improves Counterfactual-Shapley-based policy learning, and the predicted local contribution tracks the observed return gain.

Our current results also have clear scope limits. RECAP's efficient residual form relies on structure specific to Counterfactual Shapley, and it remains open whether other retrospective credit rules admit similarly tractable visibility corrections. The current implementation also requires counterfactual suffix rollouts. RECAP adds no extra suffix sampling beyond CSCA, but access to such rollouts may itself limit where the method can be used. Extending update visibility to broader attribution rules and to settings without direct simulator access are natural next steps.

AI USE STATEMENT

We used generative AI tools to assist with literature search and summarization, as well as the drafting, translation, and editing of the manuscript. AI tools were also used to help inspect experimental results and improve the presentation of figures and technical content. We did not use generative AI as a substitute for running the methods reported in this paper, nor did we use AI to fabricate experimental results. All mathematical claims and proofs were checked by the authors. AI-assisted code was reviewed by the authors and tested through the experiments reported in this paper and controlled numerical checks. Statements and citations related to the literature were verified against the original sources. The authors reviewed all AI-assisted content and take full responsibility for the final text, claims, code, and experimental results.

REPRODUCIBILITY STATEMENT

We provide the details needed to reproduce both the theoretical and experimental results. Appendix A.1 gives the Counterfactual Shapley construction and the exact calculations used in our motivating example. Appendices A.2 and A.3 contain the full constructions and proofs for our theoretical results. Appendix A.4 specifies the three experimental tasks, training and evaluation protocols, baseline implementations, random seeds, local-update diagnostics, robustness experiments, and computational budgets. All reported comparisons use fixed experimental protocols and explicitly stated sampling and evaluation procedures. Core code needed to reproduce the experiments will be provided with the supplementary material.

REFERENCES

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/16105fb9cc614fc29e1bda00dab60d41-Paper.pdf.

Meng Cao, Shuyuan Zhang, Xiao-Wen Chang, and Doina Precup. Scar: Shapley credit assignment for more efficient rlhf, 2025. URL https://arxiv.org/abs/2505.20417.

Wen-Tse Chen, Jiayu Chen, Fahim Tajwar, Hao Zhu, Xintong Duan, Ruslan Salakhutdinov, and Jeff Schneider. Retrospective in-context learning for temporal credit assignment with large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 71973–71998. Curran Associates, Inc., 2025. doi: 10.52202/085713-2416. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/680be66a383c5e7bfce021c47aedbb9e-Paper-Conference.pdf.

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, 2026. URL https://arxiv.org/abs/2603.06859.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 131646–131671. Curran Associates, Inc., 2025. doi: 10.52202/085713-4382. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/be91eb86eb74efc055cff83e953f86ce-Paper-Conference.pdf.

Zhirui Deng, Zhicheng Dou, Yutao Zhu, Ji-Rong Wen, Ruibin Xiong, Mang Wang, and Weipeng Chen. From novice to expert: Llm agent policy optimization via step-wise reinforcement learning, 2024. URL https://arxiv.org/abs/2411.03817.

Johan Ferret, Raphael Marinier, Matthieu Geist, and Olivier Pietquin. Self-attentional credit assignment for transfer in reinforcement learning. In *Proceedings of the Twenty-Ninth International*

Joint Conference on Artificial Intelligence, IJCAI-PRICAI-2020, pp. 2655–2661. International Joint Conferences on Artificial Intelligence Organization, July 2020. doi: 10.24963/ijcai.2020/368. URL http://dx.doi.org/10.24963/ijcai.2020/368.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Tanmay Gangwani, Yuan Zhou, and Jian Peng. Learning guidance rewards with trajectory-space smoothing. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 822–832. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/0912d0f15f1394268c66639e39b26215-Paper.pdf.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Gheshlaghi Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Gregory Wayne, Satinder Singh, Doina Precup, and Remi Munos. Hindsight credit assignment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/195f15384c2a79cedf293e4a847ce85c-Paper.pdf.

Chia-Chun Hung, Timothy Lillicrap, Josh Abramson, Yan Wu, Mehdi Mirza, Federico Carnevale, Arun Ahuja, and Greg Wayne. Optimizing agent behavior over long time scales by transporting value, 2018. URL https://arxiv.org/abs/1810.06721.

Aditya Kapoor, Sushant Swamy, Kale ab Tessera, Mayank Baranwal, Mingfei Sun, Harshad Khadilkar, and Stefano V. Albrecht. Agent-temporal credit assignment for optimal policy preservation in sparse multi-agent reinforcement learning, 2024. URL https://arxiv.org/abs/2412.14779.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Mykola Khandoga, Rui Yuan, and Vinay Kumar Sankarapu. Beyond uniform credit: Causal credit assignment for policy optimization, 2026. URL https://arxiv.org/abs/2602.09331.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, pp. 934–942. ACM, August 2021. doi: 10.1145/3447548.3467420. URL http://dx.doi.org/10.1145/3447548.3467420.

Mingxuan Li, Kai-Zhan Lee, and Elias Bareinboim. Counterfactual shapley credit assignment. *Reinforcement Learning Journal*, 7, 2026a.

Yugu Li, Zehong Cao, Jianglin Qiao, and Siyi Hu. Ssvpo: Effective step-level credit assignment for rl training of language models. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 46667–46707, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/4da4dd613cc9e63932d643b9c6e8f86f-Paper-Conference.pdf.

Zibin Meng and Kani Chen. Craft: Counterfactual credit assignment from free sibling rollouts for self-distilled agentic reinforcement learning, 2026. URL https://arxiv.org/abs/2606.29476.

Thomas Mesnard, Theophane Weber, Fabio Viola, Shantanu Thakoor, Alaa Saade, Anna Harutyunyan, Will Dabney, Thomas S Stepleton, Nicolas Heess, Arthur Guez, Eric Moulines, Marcus Hutter, Lars Buesing, and Remi Munos. Counterfactual credit assignment in model-free reinforcement learning.

In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 7654–7664. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/mesnard21a.html.

Thomas Mesnard, Wenqi Chen, Alaa Saade, Yunhao Tang, Mark Rowland, Theophane Weber, Clare Lyle, Audrunas Gruslys, Michal Valko, Will Dabney, Georg Ostrovski, Eric Moulines, and Remi Munos. Quantile credit assignment. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 24517–24531. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/mesnard23a.html.

Alexander Meulemans, Simon Schug, Seijin Kobayashi, nathaniel daw, and Gregory Wayne. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 68685–68735. Curran Associates, Inc., 2023. doi: 10.52202/075280-3006. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/d8bd445c2abe1343cce0e14b361b2fb3-Paper-Conference.pdf.

Eduardo Pignatelli, Johan Ferret, Matthieu Geist, Thomas Mesnard, Hado van Hasselt, Olivier Pietquin, and Laura Toni. A survey of temporal credit assignment in deep reinforcement learning, 2024. URL https://arxiv.org/abs/2312.01072.

Yun Qu, Yuhang Jiang, Boyuan Wang, Yixiu Mao, Cheems Wang, Chang Liu, and Xiangyang Ji. Latent reward: Llm-empowered credit assignment in episodic reinforcement learning, 2025. URL https://arxiv.org/abs/2412.11120.

David Raposo, Sam Ritter, Adam Santoro, Greg Wayne, Theophane Weber, Matt Botvinick, Hado van Hasselt, and Francis Song. Synthetic returns for long-term credit assignment, 2021. URL https://arxiv.org/abs/2102.12425.

Paulo Rauber, Avinash Ummadisingu, Filipe Mutz, and Juergen Schmidhuber. Hindsight policy gradients, 2019. URL https://arxiv.org/abs/1711.06006.

Zhizhou Ren, Ruihan Guo, Yuan Zhou, and Jian Peng. Learning long-term reward redistribution via randomized return decomposition, 2022. URL https://arxiv.org/abs/2111.13485.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation, 2018. URL https://arxiv.org/abs/1506.02438.

Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker (eds.), *Contributions to the Theory of Games II*, volume 28 of *Annals of Mathematics Studies*, pp. 307–317. Princeton University Press, Princeton, NJ, 1953.

Akash Velu, Skanda Vaidyanath, and Dilip Arumugam. Hindsight-dice: Stable credit assignment for deep reinforcement learning, 2023. URL https://arxiv.org/abs/2307.11897.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7285–7292, April 2020. ISSN 2159-5399. doi: 10.1609/aaai.v34i05.6220. URL http://dx.doi.org/10.1609/aaai.v34i05.6220.

Minheng Xiao and Xian Yu. Reward redistribution via gaussian process likelihood estimation, 2025. URL https://arxiv.org/abs/2503.17409.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 73511–73525. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/yuan25c.html.

Yudi Zhang, Yali Du, Biwei Huang, Ziyan Wang, Jun Wang, Meng Fang, and Mykola Pechenizkiy. Interpretable reward redistribution in reinforcement learning: A causal approach. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 20208–20229. Curran Associates, Inc., 2023. doi: 10.52202/075280-0887. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/402e12102d6ec3ea3df40ce1b23d423a-Paper-Conference.pdf.

Tianchen Zhu, Yue Qiu, Haoyi Zhou, and Jianxin Li. Towards long-delayed sparsity: Learning a better transformer through reward redistribution. In Edith Elkind (ed.), *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pp. 4693–4701. International Joint Conferences on Artificial Intelligence Organization, 8 2023. doi: 10.24963/ijcai.2023/522. URL https://doi.org/10.24963/ijcai.2023/522. Main Track.

A Appendix

A.1 Counterfactual Shapley Credit Assignment

A.1.1 Counterfactual Shapley Scoring Rule

Counterfactual Shapley Credit Assignment (Li et al., 2026a) scores actions by replacing them in counterfactual trajectories. Fix a factual trajectory $\tau = (S_1, A_1, \ldots, S_T, A_T)$, its observed rewards, and its return Y, with any discounting already included. Let π be the policy that generated the trajectory and π_{ref} the reference policy used for replacements. Both policies are fixed while computing the scores.

Write $[T] = \{1, \ldots, T\}$ and let $Z \subseteq [T]$ contain the *replaced* decision positions. This is the original CSCA mask $\boldsymbol{z} = (\mathbf{1}\{t \in Z\})_{t=1}^{T}$: an entry of one requests a replacement. Starting from $S_1^Z = S_1$, generate actions by

$$A_t^Z \sim \begin{cases} \delta_{A_t}, & t \notin Z,\ S_t^Z = S_t, \\ \pi(\cdot \mid S_t^Z), & t \notin Z,\ S_t^Z \neq S_t, \\ \pi_{\mathrm{ref}}(\cdot \mid S_t^Z), & t \in Z. \end{cases} \tag{11}$$

Here δ_{A_t} puts all mass on the factual action. State matching compares the complete state at the same time step. A position outside Z reuses its factual action only when the states match; after divergence it samples from π instead. This check is repeated at every step, including after a later return to the factual state.

Under the CSCA counterfactual simulation rule, a transition or reward reuses its observed value when its full parent configuration matches the factual one. Otherwise, it is sampled from the corresponding counterfactual mechanism. In the deterministic example below, this means applying the transition and reward functions to the counterfactual state and action.

Let Y^Z be the return of this counterfactual trajectory. We use the conditional expected contribution

$$f_\tau(Z) = Y - \mathbb{E}_{\mathrm{cf}}[Y^Z \mid \tau]. \tag{12}$$

The factual outcome Y is fixed; the expectation averages the counterfactual randomness. The Shapley rule (Shapley, 1953) assigns

$$\phi_t = \sum_{Z \subseteq [T] \setminus \{t\}} \frac{|Z|!(T - |Z| - 1)!}{T!} \big[f_\tau(Z \cup \{t\}) - f_\tau(Z) \big]. \tag{13}$$

Each marginal measures the additional loss in expected outcome from replacing action A_t in a given replacement context. CSCA averages these effects with Shapley weights. With no replacements the factual trajectory is replayed, so $f_\tau(\varnothing) = 0$. Efficiency gives

$$\sum_{t=1}^{T} \phi_t = f_\tau([T]) - f_\tau(\varnothing) = Y - \mathbb{E}_{\mathrm{cf}}[Y^{[T]} \mid \tau]. \tag{14}$$

For long trajectories, the weighted average can be estimated by sampling coalitions; the two-step calculation below uses exact enumeration.

A.1.2 Exact Calculation in the Two-Step Example

Section 2.1 uses $A_1, A_2 \in \{0, 1\}$, $S_2 = A_1$, and $Y = 100 A_1 A_2$, with a fixed initial state. Both π and π_{ref} choose each action with probability $1/2$ at every relevant state. Success requires two ones. The fully replaced trajectory therefore succeeds with probability $1/4$, giving $J_{\mathrm{ref}} = 100 \times (1/4) = 25$.

To match Figure 1, we use the equivalent *retained-action* coalition $S = [T] \setminus Z$ and define

$$v(S) = \mathbb{E}_{\mathrm{cf}}[Y^{[T] \setminus S} \mid \tau], \qquad f_\tau(Z) = Y - v([T] \setminus Z).$$

Membership in S permits factual-action reuse subject to state matching. The figure's marginals are $m_t^S = v(S \cup \{t\}) - v(S)$ for $t \notin S$. Complementing the coalition contexts in Eq. equation 13 gives the same Shapley scores. For two steps,

$$\phi_1 = \tfrac{1}{2}(m_1^{\varnothing} + m_1^{\{2\}}), \qquad \phi_2 = \tfrac{1}{2}(m_2^{\varnothing} + m_2^{\{1\}}).$$

For both factual cases in Figure 1, $A_1 = 1$. The four coalition values follow directly from the simulation rule. With neither action retained, $v(\varnothing) = 25$. Retaining only A_1 fixes $S_2 = 1$ and samples A_2 uniformly, so $v(\{1\}) = 50$. Retaining only A_2 samples the first action uniformly. If $A_1^Z = 1$, then $S_2^Z = S_2 = 1$ and the factual A_2 is reused. If $A_1^Z = 0$, the states differ and A_2^Z is resampled; the return is zero regardless. Hence $v(\{2\}) = 50A_2$. Retaining both actions reproduces the factual outcome, $v(\{1, 2\}) = 100A_2$.

For $(A_1, A_2) = (1, 1)$, the values are $v(\varnothing) = 25$, $v(\{1\}) = 50$, $v(\{2\}) = 50$, and $v(\{1, 2\}) = 100$. Every marginal is therefore

$$m_1^{\varnothing} = v(\{1\}) - v(\varnothing) = 50 - 25 = 25, \quad m_1^{\{2\}} = v(\{1, 2\}) - v(\{2\}) = 100 - 50 = 50,$$
$$m_2^{\varnothing} = v(\{2\}) - v(\varnothing) = 50 - 25 = 25, \quad m_2^{\{1\}} = v(\{1, 2\}) - v(\{1\}) = 100 - 50 = 50.$$

Thus $\phi_1 = \frac{1}{2}(25+50) = 37.5$ and $\phi_2 = \frac{1}{2}(25+50) = 37.5$. Their sum is $75 = 100-25 = Y - J_{\text{ref}}$.

For $(A_1, A_2) = (1, 0)$, the values are $v(\varnothing) = 25$, $v(\{1\}) = 50$, $v(\{2\}) = 0$, and $v(\{1, 2\}) = 0$. The marginals become

$$m_1^{\varnothing} = v(\{1\}) - v(\varnothing) = 50 - 25 = 25, \quad m_1^{\{2\}} = v(\{1, 2\}) - v(\{2\}) = 0 - 0 = 0,$$
$$m_2^{\varnothing} = v(\{2\}) - v(\varnothing) = 0 - 25 = -25, \quad m_2^{\{1\}} = v(\{1, 2\}) - v(\{1\}) = 0 - 50 = -50.$$

Thus $\phi_1 = \frac{1}{2}(25 + 0) = 12.5$ and $\phi_2 = \frac{1}{2}(-25 - 50) = -37.5$. Their sum is $-25 = 0 - 25 = Y - J_{\text{ref}}$.

A.1.3 The Cross-Time Attribution Dependency

Hold (S_1, A_1) fixed with $A_1 = 1$. Changing A_2 from 1 to 0 leaves $m_1^{\varnothing} = 25$ unchanged but changes $m_1^{\{2\}}$ from 50 to 0, so ϕ_1 falls from 37.5 to 12.5:

$$A_2 \longrightarrow m_1^{\{2\}} \longrightarrow \phi_1.$$

This is a dependency created by retrospective attribution, not a backward causal effect in the environment.

A.2 Proofs for Update Visibility and Optimization Failure

A.2.1 Notation and Update Decomposition

All expectations below are evaluated at π_{θ_0}. For retrospective scores $\psi_1, \ldots, \psi_T$, retain the notation

$$z_t = \nabla_\theta \log \pi_\theta(A_t \mid H_t)|_{\theta=\theta_0}, \qquad P_t = \sum_{j<t} \psi_j, \qquad D_t = \mathbb{E}[P_t \mid H_t, A_t] - \mathbb{E}[P_t \mid H_t].$$

The scores are held fixed in the policy update. Define

$$h_\psi = \mathbb{E}\left[\sum_{t=1}^{T} z_t \sum_{j=t}^{T} \psi_j\right], \qquad M_\psi = \mathbb{E}\left[\sum_{t=1}^{T} z_t P_t\right].$$

We assume the displayed expectations exist and the policy score is well defined under the usual score-function regularity conditions. In particular, $\mathbb{E}[z_t \mid H_t] = 0$. Splitting the total credit before each decision gives

$$h_\psi + M_\psi = \mathbb{E}\left[\sum_{t=1}^{T} z_t \sum_{j=1}^{T} \psi_j\right].$$

This decomposition does not require credit conservation.

A.2.2 Proof of the Update-Visibility Identity

To prove Proposition 1, fix t. The score z_t is determined by (H_t, A_t), so the tower property gives

$$\begin{aligned}\mathbb{E}[z_t P_t] &= \mathbb{E}[z_t \mathbb{E}[P_t \mid H_t, A_t]] \\ &= \mathbb{E}[z_t D_t] + \mathbb{E}[z_t \mathbb{E}[P_t \mid H_t]]\,.\end{aligned}$$

The second term vanishes by conditioning on H_t:

$$\mathbb{E}[z_t \mathbb{E}[P_t \mid H_t]] = \mathbb{E}[\mathbb{E}[z_t \mid H_t]\mathbb{E}[P_t \mid H_t]] = 0.$$

Thus $\mathbb{E}[z_t P_t] = \mathbb{E}[z_t D_t]$. Summing over decisions yields

$$M_\psi = \mathbb{E}\left[\sum_{t=1}^{T} z_t D_t\right].$$

If P_t is already determined by H_t, both conditional expectations in D_t equal P_t, and $D_t = 0$. Ordinary past rewards $r_1, \ldots, r_{t-1}$ are included in the pre-action history and have this property. Retrospectively assigned past credit can still depend on later actions, so it need not have zero D_t.

A.2.3 Gradient Relation under Credit Conservation

For Corollary 1, suppose $\sum_{j=1}^{T} \psi_j = Y - c$, where c is the fixed action-independent reference in the corollary. Its score contribution is zero: $\mathbb{E}[\sum_t z_t c] = 0$. The likelihood-ratio identity gives

$$\nabla J(\theta_0) = \mathbb{E}\left[\sum_{t=1}^{T} z_t Y\right] = \mathbb{E}\left[\sum_{t=1}^{T} z_t (Y - c)\right].$$

For every t, conservation implies $Y - c = P_t + \sum_{j=t}^{T} \psi_j$. Substituting this split and applying Proposition 1 gives

$$\nabla J(\theta_0) = h_\psi + M_\psi = h_\psi + \mathbb{E}\left[\sum_{t=1}^{T} z_t D_t\right].$$

Conservation therefore connects the previously identified omitted credit contribution to the original return gradient.

A.2.4 Construction for Theorem 1

We now give the construction used in Theorem 1. Start with horizon $T = 2$. The initial state is x or y, each with probability $1/2$, and both actions are binary. After the first action, the state records the initial context and that action: $x \to x_{A_1}$ and $y \to y_{A_1}$. The first reward is zero. The second reward is the undiscounted return

$$Y_x = 64 A_1 A_2 - 16, \qquad Y_y = 60(1 - A_1) - 30.$$

In context x, the return is 48 for $(A_1, A_2) = (1, 1)$ and -16 otherwise. In context y, it is 30 when $A_1 = 0$ and -30 when $A_1 = 1$, independently of A_2.

Every reachable decision state uses the same scalar parameter:

$$\pi_\theta(A_t = 1 \mid s) = q_\theta = \sigma(\theta), \qquad \theta_0 = 0, \qquad \pi_{\mathrm{ref}} = \pi_{\theta_0}.$$

Here $\sigma(\theta) = (1 + e^{-\theta})^{-1}$. At θ_0, both actions have probability $1/2$ at every state and $z_t = A_t - 1/2$. The context-specific reference returns are

$$J_x^{\mathrm{ref}} = 64(1/2)^2 - 16 = 0, \qquad J_y^{\mathrm{ref}} = 60(1 - 1/2) - 30 = 0.$$

Hence $J_{\mathrm{ref}} = \frac{1}{2} J_x^{\mathrm{ref}} + \frac{1}{2} J_y^{\mathrm{ref}} = 0$. The fully replaced counterfactual also has conditional mean zero in each context, so trajectory-wise efficiency gives $\phi_1 + \phi_2 = Y - J_{\mathrm{ref}} = Y$.

We use the retained-action coalition S and value $v(S)$ from Appendix A.1; the replaced positions are $Z = \{1, 2\} \setminus S$. All coalition values below condition on the factual context and actions. In context x, $v(\varnothing) = 0$, $v(\{1\}) = 32 A_1 - 16$, and $v(\{1, 2\}) = Y_x$.

For $v(\{2\})$, the first action is resampled. With factual $(1,0)$, a replacement value of 1 preserves state x_1 and reuses $A_2 = 0$, giving -16. A replacement value of 0 reaches x_0 and requires resampling A_2; either action still gives -16. Thus $v(\{2\}) = -16$. With factual $(1,1)$, these branches give 48 and -16, so $v(\{2\}) = \frac{1}{2}(48) + \frac{1}{2}(-16) = 16$. For either factual trajectory with $A_1 = 0$, the replacement value 0 preserves x_0 and gives -16. The replacement value 1 reaches x_1 and resamples A_2, giving expected return 16. Hence $v(\{2\}) = \frac{1}{2}(-16) + \frac{1}{2}(16) = 0$.

Applying the two-player Shapley formulas,

$$\phi_1 = \tfrac{1}{2}\big[v(\{1\}) - v(\varnothing) + v(\{1,2\}) - v(\{2\})\big],$$
$$\phi_2 = \tfrac{1}{2}\big[v(\{2\}) - v(\varnothing) + v(\{1,2\}) - v(\{1\})\big],$$

gives the exact values in context x:

(A_1, A_2)	$v(\varnothing)$	$v(\{1\})$	$v(\{2\})$	$v(\{1,2\})$	ϕ_1	ϕ_2
$(0,0)$	0	-16	0	-16	-16	0
$(0,1)$	0	-16	0	-16	-16	0
$(1,0)$	0	16	-16	-16	8	-24
$(1,1)$	0	16	16	48	24	24

In context y, $v(\varnothing) = v(\{2\}) = 0$ and $v(\{1\}) = v(\{1,2\}) = Y_y$. Thus A_2 is a dummy player and

$$(\phi_1, \phi_2) = \begin{cases} (30, 0), & A_1 = 0, \\ (-30, 0), & A_1 = 1, \end{cases}$$

independently of the factual value of A_2.

A.2.5 Verification and Extension to Arbitrary Horizons

Credit conservation. For the four context-x trajectories, the sums are $-16 + 0 = -16$ for both $(0,0)$ and $(0,1)$, $8 - 24 = -16$ for $(1,0)$, and $24 + 24 = 48$ for $(1,1)$. In context y, they are $30 + 0 = 30$ and $-30 + 0 = -30$. Every sum equals its factual return, so $\phi_1 + \phi_2 = Y - J_{\text{ref}}$ on every trajectory.

Within-state action ordering. All action values in this paragraph are under π_{θ_0}. At the first decision in context x,

$$Q_1(x,0) = -16, \qquad Q_1(x,1) = \tfrac{1}{2}(-16) + \tfrac{1}{2}(48) = 16,$$
$$\mathbb{E}[\phi_1 \mid S_1 = x, A_1 = 0] = -16, \quad \mathbb{E}[\phi_1 \mid S_1 = x, A_1 = 1] = \tfrac{1}{2}(8) + \tfrac{1}{2}(24) = 16.$$

Both prefer action 1. In context y, $Q_1(y,0) = 30$ and $Q_1(y,1) = -30$; the corresponding conditional Shapley means are also 30 and -30. Both prefer action 0.

At the second decision, each state fixes the first action, so the table entries for ϕ_2 are also its conditional means. At x_0, $Q_2(x_0,0) = Q_2(x_0,1) = -16$ and both Shapley scores are zero. At x_1, the action values are $(-16, 48)$ and the Shapley scores are $(-24, 24)$, both ordered by actions $(0,1)$; both prefer action 1. At y_0, the action values are $(30, 30)$, and at y_1 they are $(-30, -30)$; both states have Shapley scores $(0,0)$. Thus the conditional mean scores preserve the action ordering at every reachable state, including all ties.

True return gradient. Writing $q = \sigma(\theta)$, the context returns and their average are

$$J_x(q) = 64q^2 - 16, \qquad J_y(q) = 30 - 60q, \qquad J(q) = 32q^2 - 30q + 7.$$

Since $dq/d\theta = q(1-q)$, at $q = 1/2$ we obtain

$$g_x = 128q\, q(1-q) = 16, \qquad g_y = -60q(1-q) = -15, \qquad g = \tfrac{1}{2}g_x + \tfrac{1}{2}g_y = \tfrac{1}{2}.$$

The initial objective is $J(\theta_0) = 0$.

Direct Shapley update. With the exact scores held fixed, the trajectory update is

$$\widehat{U}_\phi = z_1(\phi_1 + \phi_2) + z_2\phi_2.$$

In context x, its values are

(A_1, A_2)	(ϕ_1, ϕ_2)	$\widehat{U}_\phi$
$(0,0)$	$(-16, 0)$	8
$(0,1)$	$(-16, 0)$	8
$(1,0)$	$(8, -24)$	4
$(1,1)$	$(24, 24)$	36

Each trajectory has conditional probability $1/4$, giving $h_\phi^x = \frac{1}{4}(8+8+4+36) = 14$. In context y, $\phi_2 = 0$. If $A_1 = 0$ then $\widehat{U}_\phi = (-1/2)30 = -15$; if $A_1 = 1$ then $\widehat{U}_\phi = (1/2)(-30) = -15$. Hence $h_\phi^y = -15 = g_y$.

The two contexts have correct local rankings, but the omitted cross-time contribution reduces the positive update from x from 16 to 14, while the contribution from y stays at -15. The shared parameter combines these terms, reversing the sign:

$$g = \tfrac{1}{2}(16) + \tfrac{1}{2}(-15) = \tfrac{1}{2}, \qquad h_\phi = \tfrac{1}{2}(14) + \tfrac{1}{2}(-15) = -\tfrac{1}{2}, \qquad g^\top h_\phi = -\tfrac{1}{4} < 0.$$

Strict local decrease. The updated parameter is $\theta' = \theta_0 + \eta h_\phi = -\eta/2$. Let $q' = \sigma(-\eta/2)$. The exact objective factors as

$$J(q) = (2q-1)(16q-7).$$

For $0 < \eta < 2\log(9/7)$, the logistic formula gives $7/16 < q' < 1/2$. Consequently $2q' - 1 < 0$ and $16q' - 7 > 0$, so

$$J(\theta_0 + \eta h_\phi) = (2q'-1)(16q'-7) < 0 = J(\theta_0).$$

This proves strict decrease throughout an explicit positive neighborhood of step sizes.

Arbitrary finite horizons. For any $T > 2$, enter an absorbing state after the second reward and continue sampling actions until step T. Each added step gives zero reward and stays in that state independently of its action. The same policy $\pi_\theta(1 \mid s) = \sigma(\theta)$ is used there. No added action can affect the return. For every retained coalition $S \subseteq [T]$, its value is the two-step value of $S \cap \{1,2\}$. Thus each added player has zero marginal contribution. In a uniformly random permutation of all T players, the relative order of players 1 and 2 is still uniform. Dummy players do not change either marginal, so averaging over permutations leaves ϕ_1 and ϕ_2 unchanged and gives $\phi_3 = \cdots = \phi_T = 0$.

At every added state, $Q_t(s,0) = Q_t(s,1) = 0$, matching the zero Shapley scores. The original six states retain their action values and conditional score orderings. The objective is unchanged, and all added terms in the forward update are zero. Therefore $\nabla J(\theta_0) = 1/2$, $h_\phi = -1/2$, and $\nabla J(\theta_0)^\top h_\phi = -1/4$ for every finite $T \geq 2$. The same step-size range proves strict decrease and completes the proof of Theorem 1.

A.3 Additional Theory for RECAP

From branch separation to the Shapley representation. Fix the sampling policy $\pi = \pi_{\theta_0}$ and use the same policy for reference actions and counterfactual continuation. Throughout, $Z \subseteq [T]$ denotes *replacement* positions, and $f_\tau(Z) = Y - \mathbb{E}_{\mathrm{cf}}[Y^Z \mid \tau]$ is the game in Appendix A.1. Retain the main-text definitions

$$\bar{r}_t = \gamma^{t-1} r_t, \qquad W_t = \mathbb{E}_\pi\left[\sum_{k=t}^{T} \bar{r}_k \,\middle|\, H_t\right], \qquad W_{T+1} = 0, \qquad \Delta_t = \bar{r}_t + W_{t+1} - W_t.$$

For each nonempty Z, let $m = \min Z$. First-replacement sufficiency is the property

$$\mathbb{E}_{\mathrm{cf}}[Y^Z \mid \tau] = \sum_{k<m} \bar{r}_k + W_m.$$

The expected counterfactual return then depends on the coalition only through the time of its first replacement.

One sufficient set of conditions under the state-matching rule of Appendix A.1 is as follows. Transitions and rewards are deterministic given the state and action. For every factual trajectory and

nonempty replacement coalition, the first selected replacement differs from the factual action almost surely, and the resulting branch never returns to the factual state at the same later time:

$$\Pr(A_m^Z \neq A_m \mid \tau) = 1, \qquad \Pr(S_k^Z \neq S_k \text{ for all } k > m \mid \tau) = 1.$$

Before m, no replacement has occurred. Starting from the same initial state, the state-matching rule reuses each factual action; deterministic dynamics and rewards preserve the entire prefix. Induction gives

$$S_k^Z = S_k, \quad A_k^Z = A_k, \quad \bar{r}_k^Z = \bar{r}_k \quad (k < m), \qquad H_m^Z = H_m.$$

At m, the rule draws $A_m^Z \sim \pi(\cdot \mid H_m)$ using fresh policy randomness. The effective-replacement condition ensures that this action differs from A_m. For every $k > m$, the states differ. If $k \in Z$, the action is drawn from the reference policy π; if $k \notin Z$, the state mismatch also requires sampling from π, rather than replaying A_k. Thus later coalition membership does not change the action law. The whole suffix has the distribution of a fresh rollout from H_m under π, so

$$\mathbb{E}_{\mathrm{cf}}\left[\sum_{k=m}^{T} \bar{r}_k^Z \,\middle|\, \tau\right] = W_m.$$

Adding the factual prefix proves first-replacement sufficiency. These conditions are sufficient, not necessary. In particular, the effective-replacement condition can fail for a discrete policy: a replacement draw can equal the factual action with positive probability. Theorem 2 assumes first-replacement sufficiency directly, not this stronger route to it. The general definition of D_t in Section 2.2 requires neither condition.

Under first-replacement sufficiency, telescoping yields, for $m = \min Z$,

$$f_\tau(Z) = \sum_{k=m}^{T} \bar{r}_k - W_m = \sum_{k=m}^{T} (\bar{r}_k + W_{k+1} - W_k) = \sum_{k=m}^{T} \Delta_k.$$

Since $m \leq k$ exactly when $Z \cap [k] \neq \varnothing$, and $f_\tau(\varnothing) = 0$, this is equivalently

$$f_\tau(Z) = \sum_{k=1}^{T} \Delta_k g_k(Z), \qquad g_k(Z) = \mathbf{1}\{Z \cap [k] \neq \varnothing\}.$$

In the elementary game g_k, players $1, \ldots, k$ are symmetric and the remaining players are dummy players. A player $j \leq k$ contributes one precisely when it is the first among these k players in a uniformly random permutation, an event of probability $1/k$. Thus $\phi_j(g_k) = 1/k$ for $j \leq k$ and zero otherwise. Shapley linearity proves the first identity in Theorem 2:

$$\phi_j = \sum_{k=1}^{T} \Delta_k \phi_j(g_k) = \sum_{k=j}^{T} \frac{\Delta_k}{k}.$$

Compensation and the expected RECAP update. The residual Δ_k contributes to $\min(k, t-1)$ of the scores in $P_t = \sum_{j<t} \phi_j$. There are k such shares when $k < t$ and $t-1$ when $k \geq t$, giving

$$P_t = \sum_{k<t} \Delta_k + \sum_{k=t}^{T} \frac{t-1}{k} \Delta_k.$$

The value recursion implies the martingale-difference identity

$$\mathbb{E}_\pi[\Delta_k \mid H_k] = \mathbb{E}_\pi[\bar{r}_k + W_{k+1} \mid H_k] - W_k = 0.$$

We apply the conditional contrast defining D_t to three groups of terms. For $k < t$, Δ_k is already determined by H_t and cancels. For $k > t$, $\sigma(H_t, A_t) \subseteq \sigma(H_k)$, so the tower property gives

$$\mathbb{E}_\pi[\Delta_k \mid H_t, A_t] = \mathbb{E}_\pi[\mathbb{E}_\pi[\Delta_k \mid H_k] \,|\, H_t, A_t] = 0, \qquad \mathbb{E}_\pi[\Delta_k \mid H_t] = 0.$$

Only $k = t$ remains, with coefficient $(t-1)/t$ and $\mathbb{E}_\pi[\Delta_t \mid H_t] = 0$. Therefore

$$D_t = \mathbb{E}_\pi[P_t \mid H_t, A_t] - \mathbb{E}_\pi[P_t \mid H_t] = \frac{t-1}{t} \mathbb{E}_\pi[\Delta_t \mid H_t, A_t],$$

which proves the second identity in Theorem 2.

To prove Corollary 2, first use exact values. Expanding the trace recursion and interchanging finite sums gives, for every $\lambda \in [0, 1]$,

$$L_t^{\phi}(\lambda) = \sum_{j=t}^{T} \lambda^{j-t} \phi_j = \sum_{k=t}^{T} \frac{\Delta_k}{k} \sum_{j=t}^{k} \lambda^{j-t}.$$

The coefficient of Δ_t is $1/t$; all future residuals have zero conditional mean given (H_t, A_t). Hence

$$\mathbb{E}_{\pi}[L_t^{\phi}(\lambda) \mid H_t, A_t] = \frac{1}{t} \mathbb{E}_{\pi}[\Delta_t \mid H_t, A_t].$$

The realized signal depends on λ, but this conditional mean does not. Adding D_t restores the remaining $(t-1)/t$ share:

$$\mathbb{E}_{\pi}[L_t^{\phi}(\lambda) + D_t \mid H_t, A_t] = \mathbb{E}_{\pi}[\Delta_t \mid H_t, A_t].$$

With exact suffix values, Algorithm 1 uses the realized residual $(t-1)\Delta_t/t$ as its compensation signal. This need not equal D_t on each trajectory; its conditional mean given (H_t, A_t) is D_t, so it gives the same conditional identity.

Now let $\widehat{W}_t$ use any finite $K \geq 1$ suffix rollouts with $\mathbb{E}_{\text{aux}}[\widehat{W}_t \mid \tau] = W_t$, and set $\widehat{W}_{T+1} = 0$. Linearity gives

$$\mathbb{E}_{\text{aux}}[\widehat{\Delta}_t \mid \tau] = \Delta_t, \qquad \mathbb{E}_{\text{aux}}[\widehat{\phi}_t \mid \tau] = \phi_t, \qquad \mathbb{E}_{\text{aux}}[\widehat{L}_t^{\phi} \mid \tau] = L_t^{\phi}(\lambda).$$

The compensation recursion is also linear, so

$$\mathbb{E}_{\text{aux}}[\widehat{D}_t \mid \tau] = \frac{t-1}{t} \Delta_t, \qquad \mathbb{E}[\widehat{D}_t \mid H_t, A_t] = D_t.$$

No independence between $\widehat{W}_t$ and $\widehat{W}_{t+1}$ is needed for these expectation identities. Applying the tower property once more yields

$$\mathbb{E}[\widehat{L}_t^{\text{RECAP}} \mid H_t, A_t] = \left(\frac{1}{t} + \frac{t-1}{t} \right) \mathbb{E}_{\pi}[\Delta_t \mid H_t, A_t] = \mathbb{E}_{\pi}[\Delta_t \mid H_t, A_t].$$

Finite K preserves this mean; it can change the estimator's variance.

All scores z_t are evaluated at θ_0, and sampled targets are held fixed. The batch average in Eq. equation 7 therefore has expectation

$$\mathbb{E}[\widehat{U}_{\text{RECAP}}] = \mathbb{E}_{\pi} \left[\sum_{t=1}^{T} z_t \, \mathbb{E}_{\pi}[\Delta_t \mid H_t, A_t] \right].$$

The value definition and the tower property give

$$\mathbb{E}_{\pi}[\Delta_t \mid H_t, A_t] = \mathbb{E}_{\pi} \left[\sum_{k=t}^{T} \bar{r}_k \,\middle|\, H_t, A_t \right] - W_t.$$

Since W_t and the rewards before t are determined by H_t, their expected score contributions vanish. Under the usual score-function regularity conditions,

$$\mathbb{E}[\widehat{U}_{\text{RECAP}}] = \mathbb{E}_{\pi} \left[\sum_{t=1}^{T} z_t \sum_{k=t}^{T} \bar{r}_k \right] = \mathbb{E}_{\pi} \left[\sum_{t=1}^{T} z_t Y \right] = \nabla J(\theta_0).$$

To connect this to Section 2.2, the conditional identity for $L_t^{\phi}(\lambda)$ also holds at $\lambda = 1$. Thus $\mathbb{E}[\widehat{U}_{\phi}] = h_{\phi}$ for every $\lambda \in [0, 1]$, where h_{ϕ} is the $\lambda = 1$ return-to-go expectation defined there. Proposition 1 and the compensation identity give $\mathbb{E}[\widehat{U}_D] = \mathbb{E}_{\pi}[\sum_t z_t D_t] = M_{\phi}$. Consequently

$$\mathbb{E}[\widehat{U}_{\text{RECAP}}] = h_{\phi} + M_{\phi} = \nabla J(\theta_0).$$

Indeed, summing the Shapley representation gives $\sum_j \phi_j = \sum_k \Delta_k = Y - W_1$. The initial-history reference W_1 is determined before every action and has zero expected score contribution, including when the initial state is random. This is the credit-conservation argument from Section 2.2. These equalities concern expectations at the sampling policy; they do not identify RECAP's trajectory-level labels or finite-batch updates with those of reward-based learning.

Finite-step improvement. We now prove Theorem 3 without assuming that RECAP is unbiased. Write $\widehat{U}_{\mathrm{RECAP}} = \widehat{U}_\phi + \widehat{U}_D$, $g = \nabla J(\theta_0)$, $m_D = \mathbb{E}[\widehat{U}_D]$, and $\kappa_D = g^\top m_D$. The paired expected gain is

$$\mathcal{G}(\eta) = \mathbb{E}\Big[J(\theta_0 + \eta\widehat{U}_{\mathrm{RECAP}}) - J(\theta_0 + \eta\widehat{U}_\phi)\Big].$$

For any vector u, an L-Lipschitz gradient bounds the remainder

$$R_\eta(u) = J(\theta_0 + \eta u) - J(\theta_0) - \eta g^\top u, \qquad |R_\eta(u)| \le \frac{L\eta^2}{2}\|u\|^2.$$

Apply this separately to the two random updates. Their difference is

$$\mathcal{G}(\eta) - \eta\kappa_D = \mathbb{E}[R_\eta(\widehat{U}_{\mathrm{RECAP}}) - R_\eta(\widehat{U}_\phi)].$$

Taking absolute values and using the triangle inequality gives

$$|\mathcal{G}(\eta) - \eta\kappa_D| \le \frac{L\eta^2}{2}\mathbb{E}[\|\widehat{U}_{\mathrm{RECAP}}\|^2 + \|\widehat{U}_\phi\|^2] = \frac{L\eta^2}{2}\nu.$$

The finite second-moment assumption makes both remainders integrable. If $\kappa_D > 0$ and $L\nu > 0$, then

$$\mathcal{G}(\eta) \ge \eta\kappa_D - \frac{L\eta^2}{2}\nu > 0 \qquad \text{for } 0 < \eta < \frac{2\kappa_D}{L\nu}.$$

If $L\nu = 0$, the remainder bound vanishes and $\mathcal{G}(\eta) = \eta\kappa_D > 0$ for every $\eta > 0$.

For Corollary 3, also assume $g^\top h_\phi > 0$, where $h_\phi = \mathbb{E}[\widehat{U}_\phi]$. The same smoothness bound gives

$$\mathbb{E}[J(\theta_0 + \eta\widehat{U}_\phi)] - J(\theta_0) \ge \eta g^\top h_\phi - \frac{L\eta^2}{2}\mathbb{E}[\|\widehat{U}_\phi\|^2].$$

For $L > 0$, both this lower bound and the paired gain are positive whenever

$$0 < \eta < \min\left\{\frac{2\kappa_D}{L\nu}, \frac{2g^\top h_\phi}{L\mathbb{E}[\|\widehat{U}_\phi\|^2]}\right\}.$$

Both denominators are positive under the stated strict projection conditions. If $L = 0$, both gains are exactly their positive linear terms. Thus, in either case, all sufficiently small positive steps satisfy

$$\mathbb{E}[J(\theta_0 + \eta\widehat{U}_{\mathrm{RECAP}})] > \mathbb{E}[J(\theta_0 + \eta\widehat{U}_\phi)] > J(\theta_0).$$

Compensation can therefore strengthen an attribution update that already improves return.

A.4 Experimental Details

Tasks, training, and baseline implementations. The task definitions below use zero-based control indices $k = 0, \dots, T-1$; the main text uses $t = k + 1$. In every task, $J = \mathbb{E}[\sum_{k=0}^{T-1}\gamma^k r_k]$ with $\gamma = 0.99$. Episodes terminate at the fixed horizon, with no early termination.

RoundTrip. The state consists of two joint angles q, angular velocities $\dot{q}$, and the control index. The environment uses MuJoCo 3.12.0 with two planar rigid links of length 0.1 m and mass 0.1 kg each. Each center of mass lies 0.05 m along its link, with diagonal inertia $(4.5 \times 10^{-7}, 1/12000, 1/12000)$ kg m^2. Both joints rotate about the perpendicular axis, with damping 0.002, zero friction loss, zero gravity, and disabled contacts. The two-dimensional action $a_k \in [-1, 1]^2$ applies joint torques $0.005a_k$, held for eight RK4 integration steps of 0.005 s each. The horizon is $T = 50$, starting deterministically from $q_0 = (0, \pi/2)$ and $\dot{q}_0 = 0$. The tip position is

$$p(q) = 0.1\big(\cos q_1 + \cos(q_1 + q_2),\ \sin q_1 + \sin(q_1 + q_2)\big).$$

The targets are $p_A = p(0.2, \pi/2)$ and $p_B = p(-0.2, \pi/2)$; $p_k^\star = p_A$ for $k < 20$ and p_B thereafter. The 15-dimensional observation concatenates

$$o_k = \big(\cos q_k, \sin q_k, \dot{q}_k/5, p_A/0.2, p_B/0.2, (p(q_k) - p_k^\star)/0.2,$$
$$(50 - k)/50, \max(20 - k, 0)/50, \mathbf{1}\{k \ge 20\}\big).$$

Rewards are evaluated after the control step. Write $d_k = \|p(q_{k+1}) - p_k^\star\|$ and $v_k = \|\dot{p}(q_{k+1}, \dot{q}_{k+1})\|$. The reward is

$$r_k = -0.04\frac{d_k^2}{0.1^2} - 0.04(0.01)\|a_k\|^2 - \mathbf{1}\{k = 19\}\frac{d_k^2}{0.02^2} - \mathbf{1}\{k = 49\}\left(\frac{d_k^2}{0.01^2} + \frac{0.5v_k^2}{0.1^2}\right).$$

The simulator is deterministic; policy actions supply the sampled randomness in the main task. The tanh action map respects the motor limits without additional action clipping. Observations use only the fixed transformations above, and rewards are neither clipped nor normalized.

LaggedLink. The state is (k, q_k, x_k), comprising time, queue backlog, and realized sending rate. The horizon is $T = 60$, with $q_0 = 0$ and $x_0 = 8$. Capacity c_k follows the deterministic sequence $(10, 6, 12, 5, 10, 6)$, each value held for ten steps. A scalar action $a_k \in [-1, 1]$ sets target rate $v_k = 8(1 + a_k)$. The dynamics and delivered amount d_k are

$$x_{k+1} = 0.75x_k + 0.25v_k, \qquad d_k = \min(q_k + x_{k+1}, c_k), \qquad q_{k+1} = q_k + x_{k+1} - d_k.$$

The reward and six-dimensional observation are

$$r_k = \frac{d_k}{10} - 0.25\left(\frac{q_{k+1}}{10}\right)^2 - 0.02\left(\frac{x_{k+1} - x_k}{10}\right)^2 - \mathbf{1}\{k = 59\}\left(\frac{q_{k+1}}{10}\right)^2,$$
$$o_k = \big(q_k/20, x_k/16, c_k/16, c_{\min(k+1,59)}/16, (60 - k)/60, (10 - (k \bmod 10))/10\big).$$

The next rate depends on the previous realized rate, and undelivered traffic remains in the queue. No environment noise is sampled; the initial state and capacity schedule are fixed.

Information Allocation. The state is (k, p, P_k), where $p \in \{0, 1, 2, 3\}$ is a public initial profile and P_k is a symmetric 2×2 covariance matrix. The horizon is 48, and $P_0 = 0.6I$ in every profile. Define

$$F = \begin{pmatrix} 0.94 & 0.10 \\ 0.03 & 0.92 \end{pmatrix}, \quad (\operatorname{diag} Q_p)_{p=0}^3 = \big((0.01, 0.04), (0.04, 0.01), (0.015, 0.03), (0.03, 0.015)\big).$$

For $a_k \in [-1, 1]^2$, total effort is $b_k = (1 + a_{k,1})/2$ and allocation fraction is $f_k = (1 + a_{k,2})/2$, giving $(e_{k,1}, e_{k,2}) = (b_k f_k, b_k(1 - f_k))$. The covariance update and reward are

$$P_k^- = FP_kF^\top + Q_p, \qquad P_{k+1} = \big((P_k^-)^{-1} + \operatorname{diag}(e_{k,1}, e_{k,2})\big)^{-1},$$
$$r_k = -w_{p,s}(P_{k+1})_{11} - (1 - w_{p,s})(P_{k+1})_{22} - 0.18b_k^2, \qquad s = \lfloor k/8 \rfloor.$$

The six-phase priority schedules, with one row per profile, are

$$(w_{p,s}) = \begin{pmatrix} .8 & .2 & .7 & .3 & .8 & .2 \\ .2 & .8 & .3 & .7 & .2 & .8 \\ .2 & .3 & .8 & .7 & .2 & .8 \\ .8 & .7 & .2 & .3 & .8 & .2 \end{pmatrix}.$$

Writing $(x, y, z) = ((P_k)_{11}, (P_k)_{12}, (P_k)_{22})$, the observation contains $x/(1+x)$, $z/(1+z)$, $y/\sqrt{xz}$, both entries of $\operatorname{diag} Q_p/0.04$, the current priority pair, the next phase's priority pair (clamped to the last phase), $(48 - k)/48$, $(8 - (k \bmod 8))/8$, and a four-way one-hot profile indicator, for 15 entries in total. Covariances and actions receive no clipping or jitter. Process and measurement uncertainty enter analytically through the covariance update; no measurement outcomes or process-noise trajectories are sampled. Profiles have equal weight. Each training batch contains four episodes per profile in random order; each evaluation bank contains 1,024 per profile. Evaluation averages returns within each profile and then averages the four profile means.

Applicability of the closed form. In the main tasks, transitions and rewards are deterministic functions of state and action; Information Allocation's profile is part of the initial state. The reference and continuation policies are the same frozen sampling policy. In exact arithmetic, its tanh-Gaussian action distribution has a density, so a fresh replacement differs from the factual action almost surely. Branch separation follows from the task dynamics.

In LaggedLink, $x_{k+1} = 0.75x_k + 2(1 + a_k)$ is strictly increasing in a_k. From any current state, matching a specified factual next rate requires at most one action and therefore has probability zero. Queue emptying alone does not merge the complete state. In Information Allocation, a specified next

covariance uniquely determines the measurement increments through $\mathrm{diag}(e_1, e_2) = P_{k+1}^{-1} - (P_k^-)^{-1}$. Since $b > 0$, these increments uniquely determine $b = e_1 + e_2$ and $f = e_1/b$, hence the action. Matching the full covariance likewise has probability zero.

For RoundTrip, the contact-free RK4 transition is real analytic in joint positions, velocities, and interior actions. Fix a suffix start and a later decision time. The squared distance between the factual physical state and a fresh suffix state is analytic in their joint action sequence. It is not identically zero: along the zero-torque rest trajectory, perturbing only the last suffix action changes the next velocity.[1] A nonzero real-analytic function has a zero set of Lebesgue measure zero. The joint action law has a density, so exact state matching has probability zero, also conditionally for almost every factual trajectory.

A finite union over suffix starts and decision times preserves these zero-probability conclusions. Thus, after the first replacement, the state-matching rule samples every remaining action from the frozen policy, irrespective of later coalition membership. This gives Eq. (4) by the argument in Appendix A.3. The implementations separately monitor exact state matches and tanh saturation in finite precision.

Shared training and evaluation. All methods use seeds $\{0, 1, 2, 3, 4\}$, $1{,}200$ training rounds, and 16 fresh episodes per round. The actor has two Tanh hidden layers of width 32, with input/output sizes $15/2$, $6/1$, and $15/2$ for RoundTrip, LaggedLink, and Information Allocation. It samples $u \sim \mathcal{N}(\mu_\theta(o), 0.35^2 I)$ and applies $a = \tanh u$. Actor computations use float64. RoundTrip's initial Linear weights are generated in float32 and converted to float64; the other two tasks initialize in float64. Within each task and seed, all methods clone the same initial actor and start independent empty Adam states. Adam uses learning rate 3×10^{-4}, $\beta = (0.9, 0.999)$, $\epsilon = 10^{-8}$, and zero weight decay. Every round performs four full-batch clipped PPO actor updates with clipping threshold 0.2, summing over time and averaging over episodes. Learning labels and old log probabilities remain fixed through these updates. The common actor loop adds no general advantage normalization, entropy bonus, gradient clipping, or learning-rate schedule. Gaussian log probabilities are evaluated at the stored latent actions; the tanh Jacobian cancels in policy ratios.

Factual action-noise streams are paired across methods within each seed, with separate streams for auxiliary fitting and evaluation. CSCA and RECAP also pair suffix innovations and use the same estimator and sampling budget. During full training, each method rolls out its own current policy, so realized trajectories and suffix values can differ once the policies diverge. In the local comparisons below, both signals are computed from exactly the same factual batch and suffix estimates at a common frozen policy. The actor architecture and PPO optimization settings are matched; auxiliary models and simulation costs depend on the learning signal.

The primary metric is stochastic-policy return at round $1{,}200$, using $4{,}096$ evaluation episodes per seed and paired evaluation randomness across methods. Tables report means and sample standard deviations over the five training seeds. Method contrasts are paired within seed; descriptive 95% intervals use the five seed-level differences. Learning curves use checkpoints every 20 rounds: RoundTrip and LaggedLink use 256 monitoring episodes, while Information Allocation uses its $4{,}096$-episode bank. The fixed late average uses rounds $1120, 1140, 1160, 1180, 1200$. The endpoint is not selected from the learning curve.

PPO. The critic has the same two width-32 Tanh hidden layers as the actor, with a scalar output and separate float64 initialization using seed $10000 + \mathrm{seed}$. It predicts the remaining return in absolute discounted units. For frozen predictions W_k^{old}, set $\delta_k = \gamma^k r_k + W_{k+1}^{\mathrm{old}} - W_k^{\mathrm{old}}$, with terminal value zero, and compute $A_k = \delta_k + 0.95 A_{k+1}$ backwards. PPO uses A_k as its actor label. After the four actor updates, the critic takes four full-batch Adam updates with the same optimizer hyperparameters, minimizing $\frac{1}{2}\,\mathrm{mean}[(W_k - (W_k^{\mathrm{old}} + A_k))^2]$ against fixed targets. Discounting is already included in $\gamma^k r_k$; the trace does not multiply by γ again. PPO uses no counterfactual suffix simulation.

CSCA and RECAP. Both use $K = 2$ independent suffix rollouts from every factual pre-action state under the frozen sampling/reference policy. The residual and Shapley recursions are those in Algorithm 1. CSCA uses the zero-critic attribution trace $\widehat{L}_t^{\phi} = \widehat{\phi}_t + 0.95 \widehat{L}_{t+1}^{\phi}$, with credit discount

[1] At rest, the mass matrix is $M_0 = \frac{1}{3000}\left(\begin{smallmatrix} 5 & 1 \\ 1 & 1 \end{smallmatrix}\right)$. The linearized eight-substep velocity response has eigenvalues $2.5[1 - R(-0.005\beta)^8] > 0$, where $\beta = 4.5 \pm 1.5\sqrt{5}$ and $R(x) = 1 + x + x^2/2 + x^3/6 + x^4/24$ is the RK4 stability polynomial.

one. RECAP adds $(t-1)\widehat{\Delta}_t/t$ from the same estimates, without another suffix call. Neither method fits a critic or compensation network. These are Counterfactual Shapley learning signals inside the common PPO actor loop, not the complete training pipeline of the original CSCA implementation.

QCA and HQCA. These are two variants of the same quantile-credit method, adapted to continuous actions. Let $G_k = \sum_{j=k}^{T-1} \gamma^{j-k} r_j$ and $q_\xi(o, a, \tau)$ denote interpolated action-conditional return quantiles. QCA estimates the rank τ_k of factual G_k by a piecewise-linear inverse of its quantiles, clamped at the endpoints. HQCA predicts a distribution over quantile levels from the observed suffix and uses its mean level instead. Both use the actor label

$$\gamma^k \left(G_k - \frac{1}{32} \sum_{m=1}^{32} q_\xi(o_k, \widetilde{a}_{k,m}, \tau_k) \right), \qquad \widetilde{a}_{k,m} \sim \pi_{\mathrm{old}}(\cdot \mid o_k).$$

The 32 candidate actions are model queries, not simulator rollouts. There is no further GAE pass or PPO critic. The quantile model sums a state-value MLP with two width-128 ReLU layers and an action-conditioned residual MLP with three width-128 ReLU layers. Its first quantile residual is unconstrained; subsequent increments use softplus. QCA uses five levels and HQCA ten, at $(i - 1/2)/m$. Joint fitting uses 0.1 times the value MSE plus 0.1 times the quantile Huber loss (threshold one), summed over quantiles and averaged over transitions. Each round uses ten epochs and four minibatches per epoch, of 200, 240, or 192 transitions for the three tasks. RMSprop uses learning rate 5×10^{-4}, smoothing 0.99, $\epsilon = 10^{-8}$, zero momentum and weight decay, and no centering. HQCA then fits a backward LSTM with 128 units and a four-layer width-128 ReLU head, using the same optimizer, ten epochs, four episodes per minibatch, and soft-target cross-entropy weighted by 0.01. Inputs include the current and later factual observations, actions, and raw rewards; recurrent state resets between episodes. The quantile inverse supplies the two-point soft targets. The common actor then takes four updates with the resulting labels fixed.

H-DICE. Our adapter uses the source hindsight-action model, return density model, and DualDICE module in latent-action coordinates. Its actor label is

$$\gamma^k \big(1 - d(o_k, u_k, G_k)\, p_{\mathrm{ret}}(G_k \mid o_k)\big) \frac{G_k - \overline{G}}{s_G + 10^{-7}},$$

where $\overline{G}$ and s_G are the batch mean and sample standard deviation. The two Gaussian models use negative log-likelihood; DualDICE uses a sigmoid output with scale $c = 1$ and minimizes the empirical objective $\frac{1}{2}\mathbb{E}_h[d^2] - \mathbb{E}_{\pi,\psi}[d]$, with ψ uniform over the factual batch's minimum and maximum returns. The hindsight expectation uses sampled hindsight actions and factual returns; the other uses factual actions and sampled ψ returns. All three models have two width-128 ReLU layers, float64 parameters, Adam learning rate 3×10^{-4}, minibatch size 256, and gradient-norm limit 10 for their network weights. Each is fitted for ten epochs per round, giving 40, 40, and 30 updates per model in RoundTrip, LaggedLink, and Information Allocation. Layer weights are refreshed each round while Adam history and scalar log standard deviations are retained, as in the source modules. The return model standardizes targets with batch population standard deviation plus 10^{-6} and returns density in those standardized coordinates. The adapter preserves that density scale and the source's omission of the uniform-density factor. It retains internal return standardization but omits the source PPO pipeline's final general advantage normalization. It uses no separate value critic, GAE, or simulator suffixes; its fixed labels enter the common four-update PPO actor loop.

RUDDER. The return predictor uses the tutorial's 16-unit LSTM, with float32 observation/action inputs and a scalar prediction p_k at each prefix. With $p_{-1} = 0$ and $Y = \sum_k \gamma^k r_k$, redistributed credit is

$$\widetilde{r}_k = p_k - p_{k-1} + (Y - p_{T-1})/T, \qquad \sum_k \widetilde{r}_k = Y.$$

The predictor minimizes terminal MSE plus one half of the mean prefix MSE against Y, using Adam with learning rate 10^{-3} and weight decay 10^{-5}. It receives 10,000 updates per seed, each with eight episodes drawn uniformly with replacement from its own accumulated factual data. After n collected episodes, its cumulative update count is $\lfloor 10000n/19200 \rfloor$. The first training round uses original rewards; subsequent rounds use the frozen predictor's redistributed credits. A separate width-32 Tanh critic supplies GAE with $\lambda = 0.95$ and credit discount one, following PPO's four-update critic protocol above. The order is actor, critic, then predictor fitting. The replay buffer trains only the

predictor; the actor always uses the current factual batch. No auxiliary simulator rollouts or separate pretraining episodes are used.

The external methods retain their quantile, hindsight-ratio, or redistribution constructions while replacing the native actor trainer with the matched PPO interface. Auxiliary architectures, loss weights, and fitting schedules above are the values frozen in the experiment configurations and implemented in their adapters; they are not a claim of identical auxiliary computation.

Local-update diagnostics. Table 2 covers CSCA and RECAP policies at rounds 600 and $1{,}200$ in each task, with training seeds $\{0, 1, 2, 3, 4\}$: 60 frozen policy points in total. The RECAP/600 points extend the original diagnostic matrix using saved training checkpoints and the same sampling protocol. Each point uses an independent calibration batch of 64 episodes, followed by eight independent diagnostic repeats. Each repeat draws a factual batch of 16 episodes and $K = 2$ suffix rollouts per pre-action state to construct $\widehat{U}_\phi$, $\widehat{U}_D$, and $\widehat{U}_{\mathrm{RECAP}}$. The simulator, attribution labels, and sum-over-time, mean-over-episodes gradient reduction match the training setup.

Two separate factual batches produce reward-score gradient estimates $\widehat{g}_1$ and $\widehat{g}_2$. Each batch contains 256 episodes in RoundTrip and LaggedLink, and 128 in Information Allocation. Their return baselines depend only on the independent calibration batch, absolute time, and the public initial profile where applicable. With $\widehat{g} = (\widehat{g}_1 + \widehat{g}_2)/2$, we estimate the original update's contribution and the added contribution using $\widehat{g}^\top \widehat{U}_\phi$ and $\widehat{g}^\top \widehat{U}_D$, respectively. The reference batches are independent of the attribution batch and evaluation episodes.

The calibration batch is split into four blocks of 16 episodes. Let $\widehat{U}^{(0)}_{m,b}$ be the update from calibration block b. We set

$$S_{\theta_0} = \left[\frac{1}{8} \sum_{b=1}^{4} \left(\|\widehat{U}^{(0)}_{\phi,b}\|^2 + \|\widehat{U}^{(0)}_{\mathrm{RECAP},b}\|^2 \right) \right]^{1/2}, \qquad \eta = \rho / S_{\theta_0}.$$

The learning rate is shared by the CSCA and RECAP branches at each policy point. We use $\rho = 10^{-4}$ for the main comparison and $\rho = 10^{-3}$ for step-size sensitivity. Each repeat evaluates the starting policy and both updated branches using 256 independent episodes in RoundTrip and LaggedLink, or 128 in Information Allocation. Candidates share evaluation randomness within a repeat; both step sizes share the same update directions and evaluation episodes. Information Allocation uses a balanced, equally weighted mixture of its four initial profiles.

We compute return changes and RECAP–CSCA differences within each repeat, average the eight repeats within each training seed, and then report the mean and sample standard deviation of the five seed means.

Table 2: Detailed local-update results underlying Figure 3. Each branch takes one ordinary gradient step with the same learning rate. ΔJ is measured relative to the starting policy; the last two columns give the observed RECAP–CSCA return difference and its first-order prediction $\eta\kappa_D$. Entries are means $\pm$ sample standard deviations across five seeds, each averaged over eight repeats. The primary scale is $\rho = 10^{-4}$; task-specific return units appear in the first column.

Task (unit)	Starting policy	ΔJ_ϕ	$\Delta J_{\mathrm{RECAP}}$	RECAP − CSCA	
				Observed	Predicted
RoundTrip (10^{-3})	CSCA / 600	0.220 ± 0.332	2.901 ± 2.075	2.682 ± 1.846	2.800 ± 1.868
	RECAP / 600	-0.011 ± 0.018	0.046 ± 0.032	0.057 ± 0.038	0.072 ± 0.025
	CSCA / 1200	0.130 ± 0.180	2.870 ± 0.726	2.740 ± 0.645	2.683 ± 0.704
	RECAP / 1200	0.008 ± 0.017	0.109 ± 0.138	0.101 ± 0.121	0.108 ± 0.135
LaggedLink (10^{-3})	CSCA / 600	0.065 ± 0.084	2.989 ± 4.673	2.924 ± 4.686	2.781 ± 4.369
	RECAP / 600	0.026 ± 0.047	0.329 ± 0.371	0.303 ± 0.328	0.269 ± 0.266
	CSCA / 1200	0.024 ± 0.061	0.425 ± 0.478	0.402 ± 0.425	0.409 ± 0.402
	RECAP / 1200	0.051 ± 0.082	0.234 ± 0.416	0.183 ± 0.336	0.193 ± 0.325
Information Allocation (10^{-5})	CSCA / 600	0.575 ± 0.481	6.368 ± 6.793	5.792 ± 6.347	5.528 ± 6.271
	RECAP / 600	0.657 ± 0.474	4.066 ± 5.214	3.409 ± 4.835	3.812 ± 4.893
	CSCA / 1200	0.284 ± 0.865	7.193 ± 3.965	6.908 ± 3.368	6.329 ± 2.953
	RECAP / 1200	0.237 ± 0.264	2.597 ± 1.262	2.361 ± 1.280	2.241 ± 1.229

Table 3 reports both step sizes and the residual between the observed local gain $\mathcal{G}(\eta) = \Delta J_{\mathrm{RECAP}} - \Delta J_\phi$ and the predicted local gain $\eta\kappa_D$ in original return units, with the displayed task-specific scaling.

Table 3: Step-size sensitivity of the local return difference and prediction residual. Both scales use the same sampled update directions and evaluation episodes within each repeat. Entries are means $\pm$ sample standard deviations of five seed means; units match Table 2.

Task (unit)	Starting policy	RECAP − CSCA: observed		Observed − predicted	
		$\rho = 10^{-4}$	$\rho = 10^{-3}$	$\rho = 10^{-4}$	$\rho = 10^{-3}$
RoundTrip (10^{-3})	CSCA / 600	2.682 ± 1.846	25.608 ± 17.889	-0.119 ± 0.065	-2.396 ± 0.972
	RECAP / 600	0.057 ± 0.038	0.295 ± 0.367	-0.015 ± 0.019	-0.421 ± 0.207
	CSCA / 1200	2.740 ± 0.645	26.124 ± 6.683	0.057 ± 0.146	-0.705 ± 1.826
	RECAP / 1200	0.101 ± 0.121	0.845 ± 1.163	-0.007 ± 0.024	-0.232 ± 0.272
LaggedLink (10^{-3})	CSCA / 600	2.924 ± 4.686	28.689 ± 46.069	0.142 ± 0.321	0.874 ± 2.420
	RECAP / 600	0.303 ± 0.328	2.344 ± 2.414	0.034 ± 0.069	-0.346 ± 0.400
	CSCA / 1200	0.402 ± 0.425	3.871 ± 4.221	-0.007 ± 0.033	-0.214 ± 0.350
	RECAP / 1200	0.183 ± 0.336	1.436 ± 3.226	-0.010 ± 0.030	-0.495 ± 0.432
Information Allocation (10^{-5})	CSCA / 600	5.792 ± 6.347	56.556 ± 62.620	0.264 ± 0.806	1.272 ± 7.684
	RECAP / 600	3.409 ± 4.835	32.025 ± 47.385	-0.403 ± 0.627	-6.092 ± 5.707
	CSCA / 1200	6.908 ± 3.368	67.051 ± 33.054	0.579 ± 0.685	3.756 ± 6.178
	RECAP / 1200	2.361 ± 1.280	22.404 ± 12.811	0.120 ± 0.499	-0.003 ± 4.878

Equal-length direction probes. The RoundTrip direction experiment covers initialization, CSCA/600, CSCA/1,200, and RECAP/1,200, using all five training seeds and eight factual batches per point. For each sampled update, let $d_m = \widehat{U}_m / \|\widehat{U}_m\|$. Using 64 independent paired evaluation episodes, we estimate the centered response

$$S_m(\epsilon) = \frac{J(\theta_0 + \epsilon d_m) - J(\theta_0 - \epsilon d_m)}{2\epsilon}.$$

Table 4 reports $K = 2$ and $\epsilon = 10^{-4}$, averaging the response of each normalized stochastic update. Normalizing the mean update is a separate diagnostic. The broader saved experiment also covers $K \in \{1, 2, 4, 8, 32\}$ and $\epsilon = 10^{-3}$. These probes hold Euclidean parameter displacement fixed and measure directional return response.

Four-step PPO/Adam probes. For the same four RoundTrip snapshot classes, each of eight factual batches per seed defines two branches with identical starting actor parameters and Adam states.

Table 4: RoundTrip direction probes with equal Euclidean displacement length, $K = 2$ and $\epsilon = 10^{-4}$. S is the centered finite-difference return response per unit displacement along each normalized stochastic update. Entries are means ± sample standard deviations across five seed means.

Starting policy	S_ϕ	S_{RECAP}	RECAP − CSCA
Initialization	343.596 ± 210.646	467.508 ± 261.858	123.912 ± 63.891
CSCA / 600	8.041 ± 7.028	24.960 ± 17.277	16.919 ± 12.955
CSCA / 1200	1.848 ± 6.886	18.314 ± 5.924	16.467 ± 8.091
RECAP / 1200	0.283 ± 0.554	1.267 ± 1.382	0.984 ± 1.088

CSCA and RECAP use their respective $K = 2$ labels, which remain fixed together with the old action log-probabilities through four clipped PPO actor updates. The Adam learning rate is 3×10^{-4} and the clipping threshold is 0.2. After each update, 64 independent episodes evaluate each branch with shared evaluation randomness. Table 5 gives the return trajectory over all four updates, relative to the starting policy. Parameter displacements and clipping fractions are recorded alongside these returns in the diagnostic outputs.

Table 5: RoundTrip PPO/Adam probes from identical actor parameters and optimizer histories. Return changes are relative to the common starting policy, in original return units. Each branch performs four clipped actor updates with fixed $K = 2$ learning signals and old log-probabilities. Entries are means ± sample standard deviations across five seed means.

Starting policy	Update	ΔJ_ϕ	$\Delta J_{\mathrm{RECAP}}$	RECAP − CSCA
Initialization	1	1.94970 ± 1.35331	2.64246 ± 1.66954	0.69276 ± 0.38888
	2	3.72115 ± 2.80557	5.03671 ± 3.49998	1.31556 ± 0.82457
	3	5.30375 ± 4.33907	7.17140 ± 5.48087	1.86765 ± 1.31970
	4	6.63331 ± 5.84789	9.00861 ± 7.55280	2.37529 ± 1.89690
CSCA / 600	1	0.00337 ± 0.02374	0.04613 ± 0.05048	0.04275 ± 0.03682
	2	0.00608 ± 0.04329	0.10811 ± 0.10870	0.10203 ± 0.08633
	3	0.00786 ± 0.05953	0.14943 ± 0.14955	0.14158 ± 0.12093
	4	0.00799 ± 0.07326	0.13629 ± 0.15481	0.12830 ± 0.12122
CSCA / 1200	1	0.01844 ± 0.01927	0.05573 ± 0.03590	0.03729 ± 0.02135
	2	0.03307 ± 0.03601	0.12191 ± 0.07880	0.08884 ± 0.05488
	3	0.04425 ± 0.05056	0.17185 ± 0.11907	0.12760 ± 0.09159
	4	0.05211 ± 0.06298	0.18638 ± 0.14925	0.13426 ± 0.12398
RECAP / 1200	1	0.00004 ± 0.00107	0.00045 ± 0.00116	0.00041 ± 0.00043
	2	0.00002 ± 0.00202	0.00116 ± 0.00233	0.00114 ± 0.00118
	3	-0.00004 ± 0.00285	0.00202 ± 0.00349	0.00206 ± 0.00212
	4	-0.00014 ± 0.00360	0.00287 ± 0.00455	0.00301 ± 0.00308

Robustness and computational details. These experiments use the shared training and evaluation protocol above. Each changes its stated factor while holding the remaining configuration fixed. Method differences and changes from a reference setting are formed within each seed before aggregation.

Suffix budgets. Table 6 gives the full-training results underlying Figure 4(a–b). Both attribution methods use the same number K of independent suffixes at every pre-action state, with fixed latent standard deviation 0.35. The $K = 2$ entries reuse the main experiment's completed runs. RoundTrip's $K = 1, 8$ and LaggedLink's $K = 1, 8, 16, 32$ entries are additional full training runs. RoundTrip's separate frozen-policy probes also include $K = 32$; the full-training comparison here uses $K \in \{1, 2, 8\}$.

Table 6: Full-training suffix-budget results at round $1{,}200$. Entries are mean $\pm$ sample SD over five seeds. The last column uses paired seed-level differences.

Task	K	CSCA	RECAP	RECAP−CSCA
RoundTrip	1	-1.406 ± 0.285	-0.216 ± 0.022	1.190 ± 0.270
	2	-1.306 ± 0.245	-0.186 ± 0.012	1.120 ± 0.237
	8	-1.194 ± 0.268	-0.161 ± 0.016	1.033 ± 0.263
LaggedLink	1	36.215 ± 0.584	37.063 ± 0.097	0.848 ± 0.502
	2	36.316 ± 0.588	37.084 ± 0.100	0.769 ± 0.493
	8	36.489 ± 0.510	37.103 ± 0.095	0.615 ± 0.421
	16	36.496 ± 0.545	37.110 ± 0.094	0.614 ± 0.457
	32	36.499 ± 0.563	37.111 ± 0.094	0.612 ± 0.475

Learning and evaluating the policy standard deviation. We introduce one global trainable log-standard-deviation parameter per latent action dimension, initialized at $\log(0.35)$. The policy mean and log-standard-deviation parameters are optimized together using each method's learning signal and the same Adam configuration. All methods share the initial policy within each seed, and CSCA and RECAP retain $K = 2$. Suffix sampling uses the frozen policy, including its current standard deviation, at each training round.

Table 7 compares three evaluations: the original policies trained and evaluated with fixed standard deviation; the new policies trained and evaluated with learned standard deviations; and the same new policies evaluated with standard deviation temporarily set to 0.35 in every dimension. The third evaluation preserves the learned mean network and changes only test-time stochasticity. All three tasks retain a positive mean RECAP–CSCA difference under this common test scale. For Information Allocation, the difference is 0.112664 with learned scales and 0.064055 at the common scale, compared with 0.076545 for the original fixed-scale policies. The common-scale comparison evaluates the learned policy means at a shared test-time standard deviation.

Table 7: Training / evaluation standard deviations. Fixed denotes $\sigma = 0.35$ in every latent action dimension. Learned / fixed evaluates the same learned-standard-deviation model with σ temporarily set to 0.35. All entries are endpoint means $\pm$ sample SD; differences are paired by training seed.

Train / test	PPO	CSCA	RECAP	RECAP−CSCA
RoundTrip				
Fixed / fixed	-0.234 ± 0.065	-1.306 ± 0.245	-0.186 ± 0.012	1.120 ± 0.237
Learned / learned	-0.220 ± 0.062	-1.240 ± 0.221	-0.171 ± 0.011	1.069 ± 0.213
Learned / fixed	-0.251 ± 0.066	-1.330 ± 0.230	-0.203 ± 0.013	1.127 ± 0.224
LaggedLink				
Fixed / fixed	36.693 ± 0.133	36.316 ± 0.588	37.084 ± 0.100	0.769 ± 0.493
Learned / learned	36.729 ± 0.132	36.371 ± 0.588	37.154 ± 0.102	0.783 ± 0.489
Learned / fixed	36.565 ± 0.106	36.313 ± 0.590	37.046 ± 0.109	0.733 ± 0.487
Information Allocation				
Fixed / fixed	-9.375 ± 0.012	-9.367 ± 0.010	-9.291 ± 0.003	0.077 ± 0.009
Learned / learned	-9.318 ± 0.013	-9.247 ± 0.004	-9.134 ± 0.002	0.113 ± 0.004
Learned / fixed	-9.379 ± 0.012	-9.352 ± 0.005	-9.288 ± 0.003	0.064 ± 0.007

Reward-noise construction. The RoundTrip noise experiment uses fixed policy standard deviation 0.35 and $K = 2$ for both attribution methods. At time $t \in \{0, \dots, 49\}$, the observed undiscounted reward is

$$r_t^{\mathrm{obs}} = r_t + \sigma_r \epsilon_t, \qquad \epsilon_t \sim \mathcal{N}(0,1), \qquad \sigma_r = \frac{\alpha S_{\mathrm{ref}}}{\sqrt{\sum_{t=0}^{49} \gamma^{2t}}}.$$

The added noise is independent across reward draws and does not enter the dynamics or policy observations. We calibrate $S_{\mathrm{ref}} = 16.209441$ once from the square root of the mean within-seed return variance, using 256 independent episodes from each of the five initial policies. For $\alpha = 0, 1, 3$, the per-step noise standard deviations are 0, 2.871845, and 8.615536, respectively. Thus the standard deviation of the added discounted-return noise is αS_{ref}.

Training uses noisy factual rewards and noisy suffix returns; evaluation uses the original noise-free task rewards with stochastic policy actions. Independence and zero mean ensure that both reward processes have the same expected return for a fixed policy. The zero-noise entries reuse the original runs. Table 9 reports the change relative to each method's own zero-noise endpoint, complementing the absolute returns in Table 8.

Table 8: Detailed robustness results underlying Figure 4. Final original-task return (mean ± sample SD over five seeds; 4,096 evaluation episodes per seed). Top: evaluation with each policy's learned standard deviation. Bottom: reward noise of scale α during training, evaluated on the original rewards.

Task / noise level	PPO	CSCA	RECAP
Learned policy standard deviation			
RoundTrip	-0.220 ± 0.062	-1.240 ± 0.221	$\mathbf{-0.171 \pm 0.011}$
LaggedLink	36.729 ± 0.132	36.371 ± 0.588	$\mathbf{37.154 \pm 0.102}$
Information Allocation	-9.318 ± 0.013	-9.247 ± 0.004	$\mathbf{-9.134 \pm 0.002}$
Reward noise in RoundTrip (fixed policy standard deviation)			
$\alpha = 0$	-0.234 ± 0.065	-1.306 ± 0.245	$\mathbf{-0.186 \pm 0.012}$
$\alpha = 1$	-1.503 ± 0.763	-5.869 ± 2.181	$\mathbf{-1.005 \pm 0.244}$
$\alpha = 3$	-3.727 ± 2.220	-16.118 ± 9.317	$\mathbf{-2.342 \pm 0.753}$

Table 9: Return changes relative to each method's own zero-noise result: $J_m(\alpha) - J_m(0)$. Negative values denote degradation. Entries are mean ± sample SD of paired differences over five training seeds.

α	PPO	CSCA	RECAP
1	-1.268 ± 0.817	-4.563 ± 2.238	-0.819 ± 0.234
3	-3.493 ± 2.274	-14.812 ± 9.478	-2.155 ± 0.745

Simulator and computation budgets. For horizon T, one run uses $1{,}200 \times 16 \times T$ factual control steps. The suffix budget per attribution method is

$$1{,}200 \times 16 \times K \sum_{t=1}^{T} (T - t + 1) = 1{,}200 \times 16 \times K \frac{T(T+1)}{2}.$$

CSCA and RECAP have identical counts because compensation reuses the sampled residuals. PPO uses the same factual budget, learns a value function, and makes no auxiliary suffix calls. All methods submit 4,800 actor updates per run; PPO also submits 4,800 critic updates. Table 10 gives per-run budgets and recorded training-round CPU times. For Information Allocation ($T = 48$), the main $K = 2$ setting uses 0.9216 million factual steps and 45.1584 million suffix steps per attribution run. Control-step counts measure simulation work; numerical batching and task dynamics determine computation time. Historical $K = 2$ budgets describe the original reused runs.

Table 10: Per-run training cost. Step counts (millions) are identical for CSCA and RECAP. CPU seconds average recorded training-round time over five seeds, excluding monitoring, evaluation, and whole-process overhead.

Task	K	Factual	Suffix	CSCA CPU (s)	RECAP CPU (s)
RoundTrip	1	0.960	24.480	443.6	445.6
	2	0.960	48.960	866.7	840.3
	8	0.960	195.840	3290.7	3275.5
LaggedLink	1	1.152	35.136	51.0	50.9
	2	1.152	70.272	59.9	59.9
	8	1.152	281.088	232.3	232.6
	16	1.152	562.176	434.6	437.3
	32	1.152	1124.352	867.0	869.8