The Reciprocity Gradient

Yue Lin[1,5,*] **Pascal Poupart**[2,4] **Shuhui Zhu**[2,4] **Dan Qiao**[1]
Wenhao Li[3] **Yuan Liu**[5] **Hongyuan Zha**[1] **Baoxiang Wang**[1,4,†]

[1]The Chinese University of Hong Kong, Shenzhen
[2]University of Waterloo [3]Tongji University
[4]Vector Institute [5]LIGHTSPEED

[*]Work done during an internship at LIGHTSPEED. [†]Corresponding author.

Abstract

Communication is fundamental to sustaining reciprocity and cooperation in strategic interactions. We identify and formulate the *influence attribution problem* as the central optimization difficulty inherent in such dynamics for a learning agent: any action or signal the agent emits reshapes the reputations of many third parties along combinatorially branching paths before feeding back into its own future rewards, forcing the agent to account for all of these indirect channels at once when choosing every action. To address this, we introduce the *reciprocity gradient*, which explicitly backpropagates reward gradients through private estimators of opponents' policies trained from public observations. The gradient flows through the reputation chain itself analytically, rather than being estimated from sampled returns. It jointly optimizes actions and evaluative signals without intrinsic rewards or reward shaping. Empirically, the method recovers near-optimal context-sensitive policies, while sample-based baselines collapse into constant-output policies.

1 Introduction

> Ye cannot live for yourselves; a thousand fibres connect you with your fellow-men, and along those fibres, as along sympathetic[1] threads, run your actions as causes, and return to you as effects.
>
> — Melvill (1855)

As artificial agents are increasingly deployed in socially interactive ecosystems, ranging from e-commerce platforms governed by complex seller-rating algorithms to digital gig economies (e.g., Uber, Airbnb, and Upwork) reliant on two-way peer evaluations, endowing them with the ability to navigate reputation mechanisms has become a critical challenge. In both natural societies and modern digital platforms, cooperation and trust are largely sustained through two primary channels: direct reciprocity, where repeated interactions allow partners to reward past behavior (Trivers, 1971), and indirect reciprocity, where behavior is regulated by third parties observing a public reputation (Nowak & Sigmund, 1998, 2005; Alexander, 2017). From an artificial intelligence perspective, a fundamental motivation of this work is understanding whether *a single learning agent* can seamlessly integrate into an established population operating under a complex, potentially opaque rating system. To maximize long-term utility, the agent must learn not only to act appropriately but also to manage its standing by decoding the community's evaluation criteria.

However, learning to optimize behavior in such reputation-mediated games exposes a fundamental structural difficulty, which we formalize as the **influence attribution problem**. In a reputation system, the causal chain linking an agent's current action to its future rewards is deeply indirect.

[1]Here, "sympathetic" denotes physical resonance between coupled systems (e.g., sympathetic strings in acoustics).

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

When agent i acts toward agent j, the immediate environment does not provide the full learning feedback. Instead, the feedback is routed through a third party's reputational assessment (gossip) of i, which subsequently shifts the behavioral policy of future partners interacting with i. Because this feedback is delayed, diffuse, and heavily dependent on the internal logic of other agents, sample-based estimators (score-function policy gradients and temporal-difference bootstrapping) struggle to attribute expected returns to specific past actions, since they do not exploit the analytical structure of the reputation chain. Consequently, agents trained with such estimators frequently collapse into trivial, constant-output policies (e.g., unconditional cooperation or mutual defection), failing to learn the context-dependent discrimination required for complex social norms.

To address this structural bottleneck, we introduce the **reciprocity gradient**, a gradient-based optimization framework that explicitly mathematically stitches this broken causal chain back together. Rather than treating the environment and other agents as a black box, the reciprocity gradient directly backpropagates the expected reward gradients through the multi-agent reputation dynamics. This formulation enables the agent to optimize not only its direct actions but also its evaluative signals (gossip), leveraging the latter as a deeply indirect channel to steer the future behavior of the population to its advantage. By tracing the influence of both the agent's actions and its evaluative signals through the opponents' assessment functions and back into future interactions, we provide the learning agent with an analytic, highly directed optimization feedback that explicitly maps how its behavior shapes its social standing and, ultimately, its utility. To execute this without the unrealistic assumption of oracle access to opponents' internal parameters (Foerster et al., 2018; Letcher et al., 2019; Willi et al., 2022), we integrate **differentiable opponent modeling** (Albrecht & Stone, 2018; Lin et al., 2023; Li et al., 2024). By fitting private, fully differentiable surrogate estimators of opponents' policies strictly from public observations, the agent reconstructs the causal computation graph internally and computes the reciprocity gradient without requiring any privileged access.

We frame the empirical investigation around the question of whether the reciprocity gradient operates as a competent *best-response learner*: against a given, established population, can a single learning agent compute a near-optimal, context-dependent strategy purely through gradient ascent? Our empirical evaluations affirmatively answer this question. On the joint setting under strictly observational access (private differentiable surrogates fitted from public observations only), the reciprocity gradient simultaneously recovers a context-dependent action rule and a context-dependent signal scheme, reaching 99% of the full-cooperation reference and matching the 99% obtained under an oracle-access reference. Strong continuous-control baselines on the same environment, DPG (Silver et al., 2014), DDPG (Lillicrap et al., 2015), TD3 (Fujimoto et al., 2018), and the reputation-shaped LR2 (Ren et al., 2025), all collapse to constant-output policies on every setting that requires a context-dependent policy.

Contributions.

- **(i) The Influence Attribution Problem** (Section 4.1). We identify and formalize the bottleneck that makes optimization in reputation-mediated games hard: every action or signal an individual learner emits alters the reputations of many third parties along combinatorially branching paths before affecting its own future rewards, and a rational learner has to simultaneously account for every such indirect feedback channel when selecting each current action.
- **(ii) The Reciprocity Gradient** (Sections 4, 5). We formulate a decentralized analytic gradient that solves influence attribution by backpropagating through multi-agent reputation dynamics, and remove the oracle assumption via private, differentiable opponent models fitted from public observations. Focusing on a single learning agent, experiments show that the method recovers highly discriminative action and signal policies and substantially outperforms model-free baselines.

2 Related Work

This section positions the reciprocity gradient against two lines of work: classical indirect-reciprocity theory and reinforcement-learning approaches to reputation-mediated cooperation. A more detailed comparison with differentiable inter-agent communication, opponent-shaping methods, training-time opponent access, and related literature is deferred to Appendix A.

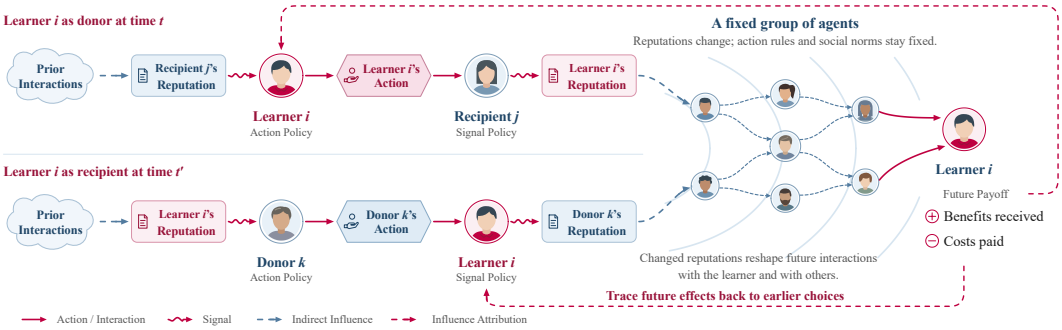


Figure 1: **The influence attribution problem.** As a donor, the learner's action shapes its own reputation through the recipient's evaluation; as a recipient, its signal shapes the donor's reputation. These reputation changes influence subsequent interactions throughout the population and ultimately affect the learner's future payoff. Influence attribution traces these effects back to the learner's earlier action and signal choices.

Indirect reciprocity in multi-agent cooperation. Nowak & Sigmund (1998, 2005); Nowak (2006) first established reputation-mediated indirect reciprocity as a fundamental evolutionary mechanism in large-scale population cooperation. Without the repeated interactions required for direct reciprocity, cooperation among strangers in one-shot encounters must instead be maintained through a shared reputation system sustained by third-party centralized assessments or peer monitoring (Milinski et al., 2002; Fehr & Fischbacher, 2004; Panchanathan & Boyd, 2004). Consequently, the study of stable assessment rules evolved from unstable first-order **Image Scoring** (Leimar & Hammerstein, 2001; Panchanathan & Boyd, 2003) to the robust **"Leading Eight"** third-order norms (Ohtsuki & Iwasa, 2004, 2006). Recent work further extends the classical setting to private and noisy assessment to capture the complexities of real-world social interactions (Hilbe et al., 2018; Fujimoto & Ohtsuki, 2023; Schmid et al., 2023). These classical analyses characterise the evolutionary stability of assessment rules: they ask whether a given social norm resists invasion under replicator dynamics or related selection pressures. We address a complementary question. Given a fixed population of opponents, can a single learning agent compute its best response purely from gradient ascent on its own return? We accordingly treat the leading-eight norms "Simple Standing" (L3) and "Stern Judging" (L6) (Ohtsuki & Iwasa 2006) as fixed test-bed opponents rather than as the object of stability analysis, and the resulting learner recovers highly discriminative action and signal policies.

Reputation systems with reinforcement learning. Leveraging the rich representational capacity of Reinforcement Learning (RL) in complex environments (Mnih et al., 2015; Leibo et al., 2017), recent studies have begun replacing classical rule-based strategies with learned reputation policies. In Anastassacos et al. (2021), the authors developed Q-learning algorithms with public reputation labels to enhance state information in the donation game. They found that cooperation could not naturally arise from scratch without seeding the population with fixed-strategy agents or introducing an introspective bonus. Vinitsky et al. (2023) utilized public sanctions based on reputation states as a shared signal to train classifiers for judging behaviors, effectively creating a top-down reputation mechanism in social dilemmas. Yaman et al. (2023) employed decentralized sanctioning matrices to redistribute rewards, disincentivizing antisocial roles while incentivizing prosocial ones that are not intrinsically profitable. Similarly, Ren et al. (2025) (LR2) rescaled environmental rewards by the agent's reputation to shape cooperative policies. However, such prior approaches typically rely on environmental customizations or heuristic reward designs, which bypass the core difficulty that reputation-mediated games impose on a learner: the causal chain from an action to future rewards is routed through the population, where the action is first assessed by third-party gossip and subsequently shifts the behavioural policies of future partners (we formalize this as the *influence attribution* problem in Section 4.1). Sample-based estimators (score-function and temporal-difference) do not exploit the analytical structure of this chain, and the resulting agents often collapse into trivial constant-output policies. To resolve this, we introduce the **reciprocity gradient**, which captures these influence links by backpropagating through differentiable surrogates of the opponents. Unlike prior work, our

learner discovers discriminative policies driven entirely by the analytic gradient of the unmodified donation-game reward, with no intrinsic bonus, reward shaping, or seed-strategy intervention.

3 Problem Setup

This section formalizes the continuous random-matching donation game and states the learning objective. The donation game isolates delayed reputation-mediated returns: a cooperating donor pays an immediate cost, while any compensating reputational benefit, if it occurs, arrives later through gossip signals, reputation updating, and subsequent actions toward the donor. Its simple benefit–cost structure and well-studied norms permit analytical best-response characterizations for benchmarking outcomes, making it a canonical testbed for indirect reciprocity. The donation game is well known to be a payoff-equivalent benefit–cost parametrization of the Prisoner's Dilemma.

At each timestep a matching rule pairs two agents into asymmetric roles, a *donor* and a *recipient*. The donor observes the recipient's reputation and chooses a continuous contribution level, paying a personal cost in exchange for a multiplied benefit to the recipient. The recipient then emits a continuous gossip signal that appends to the donor's reputation history and shapes how future partners assess them. We use directional superscripts throughout: $a_t^{i\to j}$ reads "donor i acts toward recipient j at step t," and $\sigma_t^{i\leftarrow j}$ reads "recipient j emits a signal about donor i at step t." Formally:

Definition 3.1 (Continuous Random-Matching Donation Game with Post-Gossip)**.** *A game $\mathcal{G} = \big(\mathcal{N},\, T,\, \mathcal{A},\, \Sigma,\, (\mathcal{O}_i)_{i\in\mathcal{N}},\, b,\, c\big)$ consists of a finite population $\mathcal{N}$ of N agents, horizon $T \in \mathbb{N}\cup\{\infty\}$, donor action space $\mathcal{A} = [0,1]$, signal space $\Sigma = [0,1]$, per-agent observation spaces $\mathcal{O}_i$, benefit multiplier $b \in (1,\infty)$, and donation cost $c \in (0,b)$. Each agent i has an action policy $\pi^i : \mathcal{O}_i \to \mathcal{A}$ and a gossip policy $\varphi^i : \mathcal{O}_i \to \Sigma$. At each step t, a matching rule ξ draws a donor i and a recipient $j \neq i$. The donor first chooses an action $a_t^{i\to j} = \pi^i(o_t^i) \in \mathcal{A}$; the recipient then emits a signal $\sigma_t^{i\leftarrow j} = \varphi^j(o_t^j) \in \Sigma$ about that action; instantaneous rewards $r_t^i = -c\, a_t^{i\to j}$ and $r_t^j = b\, a_t^{i\to j}$ accrue to the donor and recipient. Each emitted signal is appended to the donor's history, $\boldsymbol{\sigma}_t^i = \big(\sigma_{t_1}^{i\leftarrow k_1}, \ldots, \sigma_{t_m}^{i\leftarrow k_m}\big)$ with $t_1 < \cdots < t_m \le t$, and the reputation score is $s_t^i = f(\boldsymbol{\sigma}_t^i)$ for a fixed, differentiable aggregator f known to the learner (raw history in the main text; variants in Appendix F).*

The environment is defined by the following structural configurations:

Action and signal semantics. The continuous action space $\mathcal{A} = [0,1]$ represents the magnitude of cooperation, where 0 corresponds to absolute defection (zero donation) and 1 represents full cooperation (maximum donation). The continuous signal space $\Sigma = [0,1]$ operates as a pure communication channel devoid of pre-programmed semantics. While classical models of canonical social norms, such as the leading eight, explicitly hard-code the assumption that a signal of 1 denotes a "good" reputation, our learning agent possesses no such prior. Instead, the interpretation of any given signal is strictly emergent: the agent successfully recovers this exact semantic mapping purely from the analytic gradients backpropagated through the evaluation dynamics of its established opponents.

Orders of reputation and individualized observation. Each agent i maintains its own private and independent action policy π^i and signal policy φ^i. The observation space $\mathcal{O}_i$ encapsulates the public information required by these policies (e.g., the attached signals of the opponent and the agent itself, as well as relevant historical actions), and its exact composition is dictated by the underlying order of reputation. *First-order* reputation (image scoring) depends solely on the donor's current action, so the signal policy observes only the action: $\sigma_t^{i\leftarrow j} = \varphi^j(a_t^{i\to j})$; this is evolutionarily unstable (Leimar & Hammerstein, 2001; Panchanathan & Boyd, 2003). *Second-order* reputation additionally evaluates the recipient's standing to enable justified punishment, expanding the observation space so the signal policy evaluates both the donor's action and the recipient's aggregated reputation score: $\sigma_t^{i\leftarrow j} = \varphi^j\big(a_t^{i\to j},\, s_t^j\big)$, where $s_t^j = f(\boldsymbol{\sigma}_t^j)$ aggregates the recipient's prior signal history through a differentiable function f, e.g., a running mean $f_{\text{mean}} = \frac{1}{m}\sum_l \sigma_{t_l}^{j\leftarrow k_l}$ or an EMA $f_{\text{EMA}} = (1-\lambda)\sum_l \lambda^{m-l}\sigma_{t_l}^{j\leftarrow k_l}$ with decay $\lambda \in (0,1)$. We utilize continuous relaxations of two evolutionarily stable second-order rules from the classical "leading eight" (Ohtsuki & Iwasa, 2004, 2006): L3 (Simple Standing) and L6 (Stern Judging). Higher-order rules yield negligible additional cooperation (reviewed in Appendix B).

Matching regime. Both evaluated regimes ensure perfectly balanced role assignments. In *(1) direct-reciprocity allowed*, an episode concatenates a stochastic number of random round-robins over the $N(N-1)$ ordered pairs. Because pairs recur, both reputation and partner-history channels contribute to the gradient. In *(2) direct-reciprocity disallowed*, an episode comprises a single round-robin. Each pair meets exactly once, isolating reputation as the sole inter-agent channel (Nowak & Sigmund, 1998; Ohtsuki & Iwasa, 2006). We denote the policy-independent matching law as ξ and the role indicator as $\mathbb{1}(i = \mathsf{D}_t, j = \mathsf{R}_t)$. Setting (E) (Appendix J.6) details this matching ablation.

Reputation initialization. Initial reputations are drawn from three distributions: constant 0.5, uniform $U[0,1]$, or the empirical stationary distribution of the opponent norm (Appendix M).

Discrete variants. The continuous state and action spaces $\mathcal{A} = \Sigma = [0,1]$ can be discretized via the Gumbel-softmax relaxation (Jang et al., 2017; Maddison et al., 2017), albeit acting as a biased estimator.

The Objective Function. Every agent is self-interested and rational: it optimizes only its own expected cumulative reward $\mathbb{E}_\xi[\sum_t r_t^i]$, with no shared objective and no team return. Two channels feed this return: the cost paid as donor and the benefit received as recipient. The gossip policy φ does not contribute rewards directly, but it enters indirectly through the reputation scores conditioning every donor decision; φ shapes the present action distribution through the reputation-history channel. Our method optimizes this objective directly, with no auxiliary rewards, reward shaping, or intrinsic-motivation terms.

4 The Reciprocity Gradient

This section first identifies the influence attribution problem, then derives the reciprocity gradient to solve it, and finally removes its oracle assumption via differentiable opponent modeling.

4.1 The Influence Attribution Problem

The learning problem has a non-trivial multi-agent reputation-attribution structure: any action or signal i emits propagates through a combinatorially branching tree of downstream reputation updates, touching many third parties before re-entering i's own reward stream. Both policies bear on the future benefits i collects and the costs i pays through this coupled web (Appendix C traces concrete examples for π^i and φ^i). Enumerating these channels by hand is hopeless. We formalize this bottleneck as the *influence attribution* problem. Sample-based estimators (score-function and temporal-difference) treat this delayed feedback as opaque noise, recovering the chain only implicitly and typically collapsing into constant-output policies. To resolve this, our analytical alternative explicitly stitches the broken causal chain back together by treating the multi-agent society as an unrolled differentiable computational graph, allowing the learner to backpropagate reward gradients directly through its peers' evaluation logic (see Appendix C.3 for a minimal example). While conceptually elegant, realizing this without unrealistic oracle access to opponents' true parameters presents a severe algorithmic challenge, which we solve by dynamically fitting private, fully differentiable surrogate models purely from public observations.

4.2 Gradient Highway

Let θ^i denote the parameters of agent i's action policy π^i, and η^i the parameters of its signal policy φ^i. To isolate the structural identity and simplify the notation, we derive the following recursion under first-order gossip, where each policy depends on a single input: the action policy reads only the recipient's reputation, $a^{j\to i} = \pi^j(s^i)$, and the gossip policy reads only the donor's action, $\sigma^{j\leftarrow i} = \varphi^i(a^{j\to i})$. Higher-order assessment rules (Ohtsuki & Iwasa, 2004, 2006) extend φ^i with extra inputs (s^i, s^j, etc.); each adds one dependency edge to the same forward graph and is discussed later. Under this setting, the total derivative of the expected return commutes with the expectation and reduces to an expectation over per-agent policy derivatives:

$$\frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\left[\sum_{t=1}^{T} r_t^i\right] = \frac{1}{N(N-1)}\mathbb{E}_\xi\left[\sum_{t=1}^{T}\Big(-c\cdot\sum_{j\neq i}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^i(s_t^j) + b\cdot\sum_{j\neq i}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^j(s_t^i)\Big)\right]. \tag{1}$$

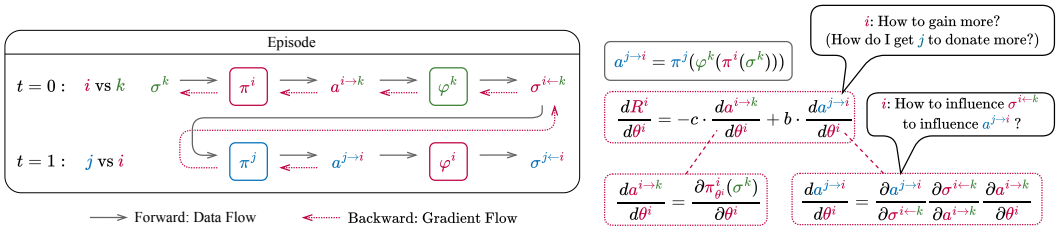


Figure 2: Reciprocity gradient: interaction flow and gradient path over two timesteps. Three-agent donation game with first-order gossip. Row heading "x vs y" means x is the donor and y is the recipient. Each row depicts one timestep t: the donor observes the recipient's reputation σ, applies its action policy π (boxed) to emit an action $a^{i\to j}$ (superscript read "from donor i to recipient j"), and the recipient applies its signal policy φ (boxed) to produce a new reputation entry $\sigma^{i\leftarrow j}$ (read "about agent i, produced by agent j").

The same identity holds with θ^i replaced by η^i. The expression inside $\mathbb{E}_\xi[\cdot]$ does not write ξ explicitly, but the matching sequence still enters through the reputation arguments s_t^j and s_t^i: ξ determines which ordered pair acts at each step $t' \le t$, which determines what gossip signal gets emitted and into which agent's reputation slot it is appended, so the value of every s_t^j along the rollout is a function of ξ. The total-derivative $\mathrm{d}/\mathrm{d}\theta^i$ on each term then captures the dependence of s_t^j on θ^i along this same realized matching sequence, and the rest of this section unpacks it into a closed-form recursion. Appendix E gives the full step-by-step derivation, including the conditions that let the derivative penetrate the expectation, the role-indicator simplification of the objective, and the extension to general-sum matrix games and continuous-deterministic games.

The objective gradients $\mathrm{d}\mathbb{E}_\xi[\sum_t r_t^i]/\mathrm{d}\theta^i$ and $\mathrm{d}\mathbb{E}_\xi[\sum_t r_t^i]/\mathrm{d}\eta^i$ reduce, after pushing the derivative inside the expectation, to two families of inner total derivatives, $\mathrm{d}\pi^i(s_t^j)/\mathrm{d}\theta^i$ and $\mathrm{d}\pi^j(s_t^i)/\mathrm{d}\theta^i$ for $j \neq i$, together with the analogous pair where θ^i is replaced by η^i. The chain rule decomposes each one as

$$\frac{\mathrm{d}}{\mathrm{d}\theta^i}\,\pi^i(s_t^j) = \underbrace{\frac{\partial\pi^i(s_t^j)}{\partial\theta^i}}_{\text{local}} + \underbrace{\frac{\partial\pi^i(s_t^j)}{\partial s_t^j}\cdot\frac{\mathrm{d}s_t^j}{\mathrm{d}\theta^i}}_{\text{transit through reputation}}. \tag{2}$$

For $\pi^j(s_t^i)$ the local term vanishes, since θ^i does not enter π^j at fixed reputation. The transit term has the same shape across all four cases: a local gradient of the policy output with respect to one reputation score, times the total derivative of that reputation score with respect to the parameter.

Explicit credit assignment through reputation. Standard policy-gradient estimators capture delayed reputational effects implicitly through sampled returns; RG differentiates these effects explicitly through the transit term in Equation 2. Treating all reputation scores as constants during backpropagation removes this term, setting $\mathrm{d}s_t^\ell/\mathrm{d}\theta^i = 0$, and analogously for η^i. The action policy retains only local donation-cost gradients, making $a^{i\to j} = 0$ for every recipient optimal for the remaining cost-only objective. Future reputational benefits remain in the forward return but provide no gradient. Equation 2 concerns the action-policy gradient; stopping the reputation chain has an even stronger effect on the signal-policy gradient. The signal policy loses every reward path: signals are costless, follow the current payoff, and affect future returns only through reputation-dependent actions. With separate signal-policy parameters and no auxiliary loss or regularization, its gradient is zero and its parameters remain at initialization.

Gradient as influence. The inner derivatives $\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^i(s_t^j)$ and $\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^j(s_t^i)$, for all $j \neq i$, are non-local. A change in θ^i directly alters the immediate actions emitted by agent i. These modified outputs subsequently perturb the interacting partners' inputs, shifting their gossip signals about agent i, which perturbs every downstream reputation score and in turn every future action conditioned on those scores. The derivative chains through the population along these reputation channels. Figure 2 sketches the structure on a minimal two-timestep example; Appendix C.3 gives an explicit influence calculation for the same interaction order under L6 assessment. We use *influence* to mean the gradient

of a future action or signal involving i with respect to i's own parameters; computing it amounts to a single backward pass through the rolled-out interaction graph. This rationale also applies to η^i.

The parameter-to-reputation derivative $\mathrm{d}s^\ell/\mathrm{d}\theta^i$ factors through the reputation-to-reputation derivative $\mathrm{d}s^\ell/\mathrm{d}s^m$. The parameter θ^i controls i's emitted actions $a^{i\to k}$; each such action drives the recipient's gossip rule, producing a signal $\sigma^{i\leftarrow k}$ that gets appended to s^i. So θ^i enters the reputation graph at one node, s^i. The signal-policy parameter η^i controls the signals i emits about past donors $k \neq i$; each of those appends to a different s^k, so η^i enters at many nodes. Write $\mathcal{E}(\theta^i) = \{i\}$ and $\mathcal{E}(\eta^i) = \{k \in \mathcal{N} : k \neq i\}$ for the entry set of each parameter. Once θ^i has entered s^m, $m \in \mathcal{E}(\theta^i)$, its effect on any other reputation s^ℓ propagates through the reputation graph and is captured by $\mathrm{d}s^\ell/\mathrm{d}s^m$. Applying the chain rule across the entry set:

$$\frac{\mathrm{d}s^\ell}{\mathrm{d}\theta^i} = \sum_{m \in \mathcal{E}(\theta^i)} \underbrace{\frac{\mathrm{d}s^\ell}{\mathrm{d}s^m}}_{\text{onward propagation}} \cdot \underbrace{\frac{\partial s^m}{\partial \theta^i}}_{\text{entry channel}}. \tag{3}$$

The action-policy parameter θ^i enters s^i through one donor-side interaction in which i is the donor and some $j \neq i$ the recipient: i's action $a^{i\to j}$ depends on θ^i, the recipient emits a signal $\sigma^{i\leftarrow j}$ about i whose value depends on $a^{i\to j}$, and that signal appends to s^i. The signal-policy parameter η^i enters s^k, $k \neq i$, through one recipient-side interaction with k as donor and i as recipient: i's emitted signal $\sigma^{k\leftarrow i}$ depends on η^i, and the signal appends to s^k:

$$\frac{\partial s^i}{\partial \theta^i} = \frac{\partial s^i}{\partial \sigma^{i\leftarrow j}} \frac{\partial \sigma^{i\leftarrow j}}{\partial a^{i\to j}} \frac{\partial a^{i\to j}}{\partial \theta^i}, \qquad \frac{\partial s^k}{\partial \eta^i} = \frac{\partial s^k}{\partial \sigma^{k\leftarrow i}} \frac{\partial \sigma^{k\leftarrow i}}{\partial \eta^i}. \tag{4}$$

The first (action-policy) chain has three factors: aggregator slot, recipient's gossip rule, and donor's action policy. The second (signal-policy) chain has two factors, because the parameter η^i already lives in the gossip rule that produces $\sigma^{k\leftarrow i}$, so no donor-action factor intervenes. Together with the entry-set structure (θ^i at one node, η^i at $N-1$ nodes), this is the analytic origin of the asymmetric-LR protocol of Setting (A.3) (Appendix J.2) and of the gradient-norm gap that widens under indirect-only matching (Setting (E), Appendix J.6).

The reduction is now explicit: after one chain-rule step at the output and one entry step at the parameter, every reciprocity-gradient quantity is a linear combination of reputation-to-reputation total derivatives $\mathrm{d}s^\ell/\mathrm{d}s^m$.

The recursion as gradient highway. Fix distinct agents i and j. Their reputation scores are produced by the aggregator f acting on signal histories: $s^j = f(\boldsymbol{\sigma}^j)$, $s^i = f(\boldsymbol{\sigma}^i)$. An infinitesimal change in s^i affects s^j in two ways: directly, when the change reaches s^j in a single hop, and indirectly, when it first transits through some third agent's reputation s^k and then propagates onward. Writing this out:

$$\frac{\mathrm{d}s^j}{\mathrm{d}s^i} = \underbrace{\frac{\partial s^j}{\partial s^i}}_{\text{one hop } i \to j} + \sum_{k \notin \{i,j\}} \underbrace{\frac{\partial s^j}{\partial s^k}}_{\text{one hop } k \to j} \cdot \underbrace{\frac{\mathrm{d}s^k}{\mathrm{d}s^i}}_{\text{full influence } i \rightsquigarrow k}. \tag{5}$$

The partial $\partial s^j/\partial s^i$ collects every interaction in which the change in s^i reaches s^j without going through any other agent's reputation; we call this the *one-hop connection*. The summation collects all chains of length ≥ 2, factoring each into a final one-hop step from some s^k into s^j and a recursive total derivative for the prefix. Equation 5 is exact and self-similar: the unknown on the left and the unknowns inside the sum satisfy the same equation. Repeated substitution unrolls it into a sum over all paths $i \rightsquigarrow j$ in the reputation graph, with each path contributing the product of its one-hop partial derivatives, and a single backward pass through the rolled-out interaction graph evaluates this path-sum automatically without enumerating paths.

One-hop closure. We close the recursion by deriving the one-hop partial $\partial s^j/\partial s^i$ explicitly. A one-hop change from s^i to s^j proceeds through one interaction in which j is the donor and i is the recipient: j chooses an action $a^{j\to i}$ that depends on s^i, then i emits a signal $\sigma^{j\leftarrow i}$ about j which appends to s^j. Under first-order gossip the signal depends only on the donor's action, so a single chain of three partials suffices:

$$\frac{\partial s^j}{\partial s^i} = \frac{\partial s^j}{\partial \sigma^{j\leftarrow i}} \cdot \frac{\partial \sigma^{j\leftarrow i}}{\partial a^{j\to i}} \cdot \frac{\partial a^{j\to i}}{\partial s^i}. \tag{6}$$

The three factors are (i) how the reputation score s^j moves when the new signal $\sigma^{j\leftarrow i}$ is appended to its history, (ii) the donor's action's influence on the gossip $\sigma^{j\leftarrow i}$, and (iii) the recipient's reputation s^i's influence on the donor's action $a^{j\to i}$. Discretizing the action or signal space replaces the corresponding factor with its Gumbel-softmax relaxation. Equations 5 and 6 close the recursion under first-order gossip.

Extension to higher-order policies. First-order gossip reads only the donor's action; second-order additionally reads the recipient's standing s^i; third-order also reads the donor's standing s^j (Ohtsuki & Iwasa, 2004, 2006; Santos et al., 2021). Each higher order adds one more dependency edge of $\sigma^{j\leftarrow i}$ on a rollout variable in the same forward graph. Equation 5 is unchanged at every order; the one-hop partial $\partial s^j/\partial s^i$ collects one extra summand $\partial s^j/\partial\sigma^{j\leftarrow i}\cdot\partial\sigma^{j\leftarrow i}/\partial x\cdot \mathrm{d}x/\mathrm{d}s^i$ per added dependency x. As long as the gossip rule is differentiable, autograd propagates the gradient through every edge automatically.

Per-episode update. Operationally, the historical dependence noted above is captured by keeping the full episode in one autograd graph: per-step rewards and reputation updates are accumulated as live tensors throughout the rollout (no `detach()` between donor steps), and a single `torch.autograd.grad` call at the end of the episode collects the contribution of every past action to the present gradient. The policy update is therefore taken once per episode rather than once per step, and one backward pass internalizes the entire reciprocity chain that Equations 5–6 describe.

Empirical verification of gradient chains. A minimal three-agent demonstration (Appendix D) verifies the autograd computation concretely: a single `torch.autograd.grad` call on the agent's discounted return traverses all agent boundaries through the rolled-out interaction graph and matches a finite-difference estimate to float precision. The example certifies that no analytic surgery, custom backward hook, or score-function estimator is required to obtain the reciprocity gradient.

4.3 Differentiable Opponent Modeling

Backpropagating through Equation 5 requires that other agents' policies be differentiable functions of their inputs, and the learner typically does not see their parameters either. We therefore fit private, differentiable surrogates from public observations, a standard instantiation of opponent modeling (Albrecht & Stone, 2018). Agent i fits each surrogate by mean-squared regression onto publicly observed interactions involving j, without access to other agents' policy parameters or internal state: $\mathcal{L}_{\hat\pi_i^j}=\mathbb{E}_{\mathcal{B}_i^j}[(\hat\pi_i^j-a^j)^2]$ and $\mathcal{L}_{\hat\varphi_i^j}=\mathbb{E}_{\mathcal{B}_i^j}[(\hat\varphi_i^j-\sigma^j)^2]$, where $\mathcal{B}_i^j$ is i's buffer of public observations of j and a^j,σ^j are its emitted actions and signals in the respective records. The complete pseudocode is given as Algorithm 1 in Appendix F.

Estimator parameterizations. Agent i approximates the action and gossip policies of every other agent $j\neq i$ with private, differentiable estimators $\hat\pi_i^j$ and $\hat\varphi_i^j$; concrete input choices are tied to the assessment order being modeled and are deferred to Appendix F. Two parameterizations of this estimator set are admissible. The *per-opponent* variant fits one pair $(\hat\pi_i^j,\hat\varphi_i^j)$ independently for every $j\neq i$, with parameter count $O(N)$ in the population size. The *parameter-shared* variant replaces all $N-1$ pairs with a single pair $(\hat\pi(\cdot;z_j),\hat\varphi(\cdot;z_j))$ shared across opponents and indexed by a per-opponent embedding $z_j\in\mathbb{R}^d$, bringing the parameter count to $O(1)$ in N and trading per-opponent specialization for parameter efficiency at scale (Setting (D), Appendix J.5).

Update schedule. At each outer iteration i collects a batch of real episodes, takes K_{est} MSE steps on the surrogate losses above, and *freezes* the surrogates for the inner loop. The frozen surrogates then act as a stationary virtual environment in which the learner performs N_{train} policy steps: every opponent function is replaced by its surrogate, fresh matching sequences and initial reputations are sampled for the virtual graph, and the per-episode reciprocity-gradient update of the previous subsection is taken on this virtual rollout. The virtual rollouts use the same policy-independent matching law as the real environment (Section 3); substituting surrogates affects the policy-dependent part of the computation. Surrogates and policy therefore alternate in epochs rather than co-evolving inside a single autograd graph, which keeps the modeled opponent functions fixed within each inner loop. Appendix F provides the complete pseudocode (Algorithm 1) and surveys the remaining engineering choices (modeling buffer, update schedule, virtual-reward optimism).

5 Experiments

The primary objective of our empirical investigation is to evaluate the reciprocity gradient as a best-response learner and to demonstrate its capacity to resolve the influence attribution problem. To avoid evaluation bias and address potential concerns regarding environment selection, we sweep over a wide range of opponent populations and learner-side configurations. Two canonical second-order leading-eight rules from the indirect-reciprocity literature, L3 (*Simple Standing*) and L6 (*Stern Judging*) (Ohtsuki & Iwasa, 2004, 2006), anchor the canonical ends of the sweep. We also include two opponent pools, ProudCoop+AllDefector and HybridCoop+AllDefector (full definitions in Appendix I), in which the analytical best response is state-dependent rather than constant-output. Across all of these we further sweep two learner-side knobs: the reputation initialization mode, and the choice of aggregator f (Section 3) that maps the signal history into a scalar reputation score. We structure the evaluation by a taxonomy of these settings based on the topological complexity of their analytical optimum.

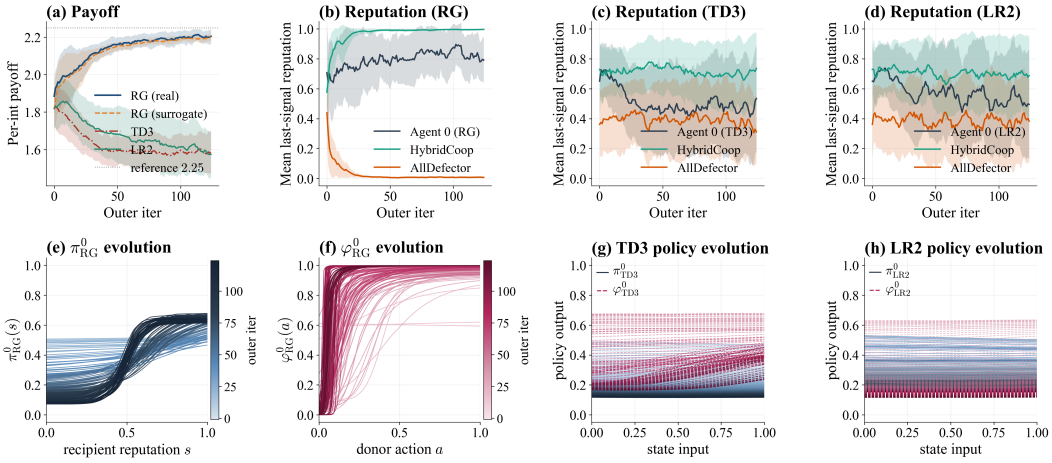


Figure 3: Joint optimization against HybridCoop+AllD. RG reaches $\sim 99\%$ of reference; TD3/LR2 collapse to flat policies. **Top:** payoff and mean per-agent reputation (bold = cross-seed mean, bands = ± 1 std). **Bottom:** representative-seed network outputs across training iters (light$\rightarrow$dark = early$\rightarrow$late). Single-pathway ablations: Figure 8.

Degenerate vs. discriminative regimes. Most classical reputation populations admit a constant-output analytical optimum (e.g., unconditional cooperators or defectors, or leading-eight opponents whose stationary reputation distributions make the optimal response state-independent). All evaluated methods reach the oracle-access reference on such regimes; we use them only as sanity checks (Appendix I). The interesting cases are populations that require *context-dependent* policies, where flat-output learners underperform. We isolate three: **action-policy optimization** (Agent 0 trains π^0 against two L6 opponents with a hard-coded honest signal channel), **signal-policy optimization** (Agent 0 trains φ^0 against ProudCoop+AllDefector with L3 assessment, where ProudCoop operates on a pride-driven mechanism: it cooperates generously when its own reputation is high, and withdraws cooperation when a low reputation signals that its prior efforts went unpraised), and **joint optimization** (Agent 0 trains both π^0 and φ^0 against HybridCoop+AllDefector; HybridCoop's cooperation probability scales jointly with its own and its partner's reputations, establishing a compounding positive dependency in which the learner must cultivate its own standing through a discriminative π^0 *and* inflate HybridCoop's standing through a discriminative φ^0 to elicit cooperation). Full opponent definitions are in Appendix I.

Across all three discriminative regimes, model-free methods consistently collapse to constant-output policies, while the reciprocity gradient recovers the analytic optima. Figure 3 visualizes the joint setting under observational access: neither TD3 nor LR2 learns a context-dependent action policy, both settling on near-zero cooperation across the entire reputation domain. This is the locally safe attractor that avoids the donor cost at the expense of future-reciprocity payoff.

Mechanistic ablations and scalability. Beyond the joint setting in Figure 3, four sweeps stress-test the analytic gradient. *(i) Indirect-only matching.* Severing direct reciprocity drops the joint payoff from 99% of reference to 79% (-20 pp) and collapses φ^0 to a constant output, while π^0 remains discriminative; the φ^0-gradient norm shrinks $5\times$ on the same transition, exactly the consequence the analysis of Section 4 predicts when the within-episode partner-history loop into φ^0 is severed. *(ii) Population-size scaling.* Across $N \in \{5, 10, 15, 20, 25, 30\}$ a parameter-shared estimator with $O(1)$ parameters in N keeps action-vs-L6 in the 93–95% band, and the joint $p{=}0.5$ column tracks the $N{=}3$ anchor of 2.225 across the entire six-fold scaling. *(iii) Pool composition.* Per-interaction payoff rises monotonically with p at every N, reaching 3.59 at $p{=}0.8, N{=}30$ as the full-cooperation reference lifts with the cooperator fraction. *(iv) Discrete-action games.* On the symmetric Prisoner's Dilemma with a Gumbel-sigmoid relaxation at the action layer, the same π^0/φ^0 networks learn a discriminative policy against both a ProudCoop+AllDefector pool and a pair of L6 opponents (per-round payoff 2.47 in the former; in the latter the agent additionally exploits good-standing partners for the temptation payoff $T{=}5$), confirming the framework extends beyond the continuous donation game without algorithmic surgery. Full numerical results, including the indirect-only baseline comparison, the complete (N, p) grid, and the discrete-action PD extension, are in Appendix K and Appendix L.

Reputation-history gradient control. To test the reputation pathway in Equation 2, we detach assessments as they enter the public history. Across ten paired seeds with matched initializations and random tapes, mean payoff falls by 0.721 in signal-only and 0.590 in joint; signal parameters remain at initialization in every stopped run (Appendix R.1). This truncates RG's pathwise gradient; score-function policy gradients retain return-weighted log-probability updates and need not have zero signal gradients.

Additional opponent-model baselines. LOLA-OM-DDPG and local-credit LIAM/NAM adaptations attain at most 0.639 in signal-only and 1.584 in joint, versus RG's 1.090 and 2.178 (Appendix R.2). These methods add opponent models or recurrent representations, but do not backpropagate reward gradients through the reputation chain as RG does. Their lower payoffs show that these particular additions do not reproduce RG's performance in either setting.

6 Limitations, Applicability, and Future Work

Appendices P and Q provide a fuller discussion of limitations, applicability, and future work. Our experiments evaluate a single learner's best response to fixed, non-learning opponents, and do not establish convergence or equilibrium properties when multiple agents learn simultaneously. The setting assumes public assessments, persistent reputation histories, and a fixed, known, differentiable aggregator: reputation scores evolve, whereas the aggregation rule does not. Private assessments and unknown or changing aggregation remain untested. The effects of shorter reputation histories on gradient propagation and opponent-model data coverage also remain open. The opponent models cover the leading-eight class; more expressive history-dependent opponents would require higher-capacity surrogates and further evaluation. Population-size scaling is evaluated up to 30 agents and does not establish performance for broader mixtures of heterogeneous agent types.

A next step is to study simultaneously learning agents and identify conditions for stable joint dynamics. If successful, this could motivate simulations inspired by e-commerce and digital gig work, where different rating rules induce different strategic responses. With well-specified and validated platform models, such simulations could eventually support counterfactual comparisons and human decision-making through an equilibrium-analysis sandbox. This would also require an explicit equilibrium concept and methods for identifying and validating candidate equilibria; stable learning alone is insufficient. The present work neither validates a real platform nor provides an equilibrium solver.

7 Conclusion

We formulated the *influence attribution problem* and the reciprocity gradient: analytic backpropagation through multi-agent reputation dynamics, paired with differentiable opponent modeling that drops the oracle-access assumption. Empirically, it recovers context-dependent policies under observational surrogates, opens substantial margins over policy-gradient and reputation-shaped baselines, and scales with population size.

Acknowledgments and Disclosure of Funding

Baoxiang Wang and Yue Lin are partially supported by the National Natural Science Foundation of China (NSFC; Grant No. 72394361) and the Shenzhen Science and Technology Program (Grant Nos. JCYJ20250604141218024 and JCYJ20250604141032005). Hongyuan Zha is supported in part by the Shenzhen Stability Science Program 2023 and the NSFC (Grant No. 72495131). Wenhao Li is supported by the NSFC (Grant No. 62406270) and the Shanghai Rising-Star Program of the Science and Technology Commission of Shanghai Municipality (Grant No. 24YF2748800). Pascal Poupart also acknowledges funding from the Canada CIFAR AI Chair program, the Natural Sciences and Engineering Research Council of Canada and a grant from IITP & MSIT of Korea (No. RS2024-00457882, AI Research Hub Project).

References

Stefano V. Albrecht and Peter Stone. Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, 258:66–95, 2018. doi: 10.1016/j.artint.2018.01.002.

Richard Alexander. *The biology of moral systems*. Routledge, 2017.

Nicolas Anastassacos, Julian García, Stephen Hailes, and Mirco Musolesi. Cooperation and reputation dynamics with reinforcement learning. *arXiv preprint arXiv:2102.07523*, 2021.

Dietmar Fehr and Matthias Sutter. Gossip and the efficiency of interactions. *Games and Economic Behavior*, 113:448–460, 2019.

Ernst Fehr and Urs Fischbacher. Third-party punishment and social norms. *Evolution and human behavior*, 25(2):63–87, 2004.

J Foerster, R Chen, M Al-Shedivat, S Whiteson, P Abbeel, and I Mordatch. Learning with opponent-learning awareness. *Autonomous Agents and Multi-Agent Systems (AAMAS 2018)*, 2018.

Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

Scott Fujimoto, Herke Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In *International conference on machine learning*, pp. 1587–1596. PMLR, 2018.

Yuma Fujimoto and Hisashi Ohtsuki. Evolutionary stability of cooperation in indirect reciprocity under noisy and private assessment. *Proceedings of the National Academy of Sciences*, 120(20): e2300544120, 2023.

Christian Hilbe, Laura Schmid, Josef Tkadlec, Krishnendu Chatterjee, and Martin A Nowak. Indirect reciprocity with private, noisy, and incomplete information. *Proceedings of the national academy of sciences*, 115(48):12241–12246, 2018.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Claudia Keser. Experimental games for the design of reputation management systems. *IBM Systems Journal*, 42(3):498–506, 2003.

Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. *arXiv preprint arXiv:1702.03037*, 2017.

Olof Leimar and Peter Hammerstein. Evolution of cooperation through indirect reciprocity. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 268(1468):745–753, 2001.

A Letcher, J Foerster, D Balduzzi, T Rocktaschel, and S Whiteson. Stable opponent shaping in differentiable games. In *2019 International Conference on Learning Representations*. OpenReview, 2019.

Zhiyuan Li, Wenshuai Zhao, Lijun Wu, and Joni Pajarinen. Backpropagation through agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13718–13726, 2024.

Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015. URL https://arxiv.org/abs/1509.02971.

Yue Lin, Wenhao Li, Hongyuan Zha, and Baoxiang Wang. Information design in multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 36:25584–25597, 2023.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in neural information processing systems*, 30, 2017.

Christopher Lu, Timon Willi, Christian A Schroeder De Witt, and Jakob Foerster. Model-free opponent shaping. In *International Conference on Machine Learning*, pp. 14398–14411. PMLR, 2022.

C Maddison, A Mnih, and Y Teh. The concrete distribution: A continuous relaxation of discrete random variables. In *Proceedings of the international conference on learning Representations*. International Conference on Learning Representations, 2017.

Naoki Masuda and Mitsuhiro Nakamura. Coevolution of trustful buyers and cooperative sellers in the trust game. 2012.

Henry Melvill. Partaking in other men's sins. 1855.

Manfred Milinski, Dirk Semmann, and Hans-Jürgen Krambeck. Reputation helps solve the 'tragedy of the commons'. *Nature*, 415(6870):424–426, 2002.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Martin A Nowak. Five rules for the evolution of cooperation. *science*, 314(5805):1560–1563, 2006.

Martin A Nowak and Karl Sigmund. Evolution of indirect reciprocity by image scoring. *Nature*, 393 (6685):573–577, 1998.

Martin A Nowak and Karl Sigmund. Evolution of indirect reciprocity. *Nature*, 437(7063):1291–1298, 2005.

Hisashi Ohtsuki and Yoh Iwasa. How should we define goodness?—reputation dynamics in indirect reciprocity. *Journal of theoretical biology*, 231(1):107–120, 2004.

Hisashi Ohtsuki and Yoh Iwasa. The leading eight: social norms that can maintain cooperation by indirect reciprocity. *Journal of theoretical biology*, 239(4):435–444, 2006.

Karthik Panchanathan and Robert Boyd. A tale of two defectors: the importance of standing for evolution of indirect reciprocity. *Journal of theoretical biology*, 224(1):115–126, 2003.

Karthik Panchanathan and Robert Boyd. Indirect reciprocity can stabilize cooperation without the second-order free rider problem. *Nature*, 432(7016):499–502, 2004.

Georgios Papoudakis, Filippos Christianos, and Stefano Albrecht. Agent modelling under partial observability for deep reinforcement learning. *Advances in Neural Information Processing Systems*, 34:19210–19222, 2021.

Tianyu Ren, Xuan Yao, Yang Li, and Xiao-Jun Zeng. Bottom-up reputation promotes cooperation with multi-agent reinforcement learning. *arXiv preprint arXiv:2502.01971*, 2025.

Fernando P Santos, Jorge M Pacheco, and Francisco C Santos. The complexity of human cooperation under indirect reciprocity. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 376(1838), 2021.

Laura Schmid, Farbod Ekbatani, Christian Hilbe, and Krishnendu Chatterjee. Quantitative assessment can stabilize indirect reciprocity under imperfect information. *Nature Communications*, 14(1): 2086, 2023.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International conference on machine learning*, pp. 387–395. Pmlr, 2014.

Ralf D Sommerfeld, Hans-Jürgen Krambeck, Dirk Semmann, and Manfred Milinski. Gossip as an alternative for direct observation in games of indirect reciprocity. *Proceedings of the national academy of sciences*, 104(44):17435–17440, 2007.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

Robert L Trivers. The evolution of reciprocal altruism. *The Quarterly review of biology*, 46(1):35–57, 1971.

Eugene Vinitsky, Raphael Köster, John P Agapiou, Edgar A Duéñez-Guzmán, Alexander S Vezhnevets, and Joel Z Leibo. A learning agent that acquires social norms from public sanctions in decentralized multi-agent settings. *Collective Intelligence*, 2(2):26339137231162025, 2023.

Timon Willi, Alistair Hp Letcher, Johannes Treutlein, and Jakob Foerster. Cola: consistent learning with opponent-learning awareness. In *International Conference on Machine Learning*, pp. 23804–23831. PMLR, 2022.

Anil Yaman, Joel Z Leibo, Giovanni Iacca, and Sang Wan Lee. The emergence of division of labour through decentralized social sanctioning. *Proceedings of the Royal Society B: Biological Sciences*, 290(2009), 2023.

Shuhui Zhu, Yue Lin, Shriya Kaistha, Wenhao Li, Baoxiang Wang, Hongyuan Zha, Gillian K Hadfield, and Pascal Poupart. Talk, judge, cooperate: Gossip-driven indirect reciprocity in self-interested llm agents. *arXiv preprint arXiv:2602.07777*, 2026.

A Extended Related Work

This appendix expands Section 2 with more detailed discussion of learning-based and analytic approaches to reputation-mediated cooperation. We emphasize the assumptions each line of work places on agents, in order to make explicit where our heuristic-free formulation differs.

Differentiable inter-agent communication. Foerster et al. (2016) established the precedent of differentiable inter-agent communication (DIAL), demonstrating that continuous messages passed during training create a usable gradient channel. However, DIAL is fundamentally restricted to fully cooperative settings optimizing a shared team return. Lin et al. (2023) extended this to general-sum games by formalizing the *signaling gradient*, allowing a sender to optimize its individual utility by differentiating through a receiver's response. Our *reciprocity gradient* belongs to this family but introduces a critical structural departure: instead of a single-hop (sender-to-receiver) link, we backpropagate through a *multi-hop reputation chain*. An agent optimizes its behavior to shape a second agent's assessment (gossip), strategically steering how a third agent will treat it in future interactions, thereby capturing the native computational structure of indirect reciprocity. Finally, unlike DIAL's simultaneous training, our differentiable opponent modeling relies strictly on public observations, completely avoiding the need for oracle access to others' parameters.

Connection to opponent shaping. Opponent shaping studies how an agent can influence other agents' learning behavior. Learning with Opponent-Learning Awareness (LOLA) (Foerster et al., 2018) anticipates an opponent's learning step and accounts for how the learner's policy changes that update, thereby shaping the opponent's subsequent learning. The reciprocity gradient addresses reputation-mediated credit assignment within interactions: it differentiates how the learner's actions and signals affect later behavior and returns while holding opponents' policy parameters fixed. These mechanisms are conceptually orthogonal: RG supplies the base best-response gradient, while LOLA adds a correction for the opponent's learning response.

To make this distinction explicit, let $J^i(\theta^i, \theta^j)$ denote agent i's expected return in two-agent notation, and let α_{L} and α_{O} denote the learner and opponent step sizes, respectively. LOLA anticipates

$$\Delta\theta^j = \alpha_{\mathrm{O}} \nabla_{\theta^j} J^j(\theta^i, \theta^j)$$

and uses the first-order look-ahead update

$$f^i_{\mathrm{LOLA}} = \underbrace{\alpha_{\mathrm{L}} \nabla_{\theta^i} J^i}_{\text{base best-response term}} + \underbrace{\alpha_{\mathrm{L}} \alpha_{\mathrm{O}} \left(\nabla_{\theta^i} \nabla_{\theta^j}^{\top} J^j \right) \nabla_{\theta^j} J^i}_{\text{LOLA opponent-shaping term}}.$$

In LOLA's policy-gradient implementation, the first term is estimated by a vanilla score-function policy gradient. RG replaces this base estimator by differentiating through reputation-mediated interactions, motivating our comparisons with standard policy-gradient methods. The second term accounts for how a change in the learner's policy changes the opponent's next parameter update and, through that update, the learner's return. In our current experiments, the opponent populations are fixed and non-learning ($\alpha_{\mathrm{O}} = 0$), so the second term is zero and canonical LOLA adds nothing beyond its chosen base estimator. RG and LOLA can in principle be composed by using RG for the base best-response term and adding an opponent-shaping correction; we leave such compositions with learning opponents to future work.

Access to opponents. While centralized training with decentralized execution (CTDE) (Lowe et al., 2017) permits training-time access to opponents' parameters, this assumption is justifiable only in fully cooperative settings. In general-sum games, opponents possess independent objectives, rendering their strategies inherently private. To enforce strict decentralization without privileged channels, our method adopts differentiable opponent modeling (Albrecht & Stone, 2018). The learning agent fits private surrogate models strictly from publicly observable interactions. Consequently, the reciprocity gradient propagates analytically through these surrogates, completely eliminating the need for shared parameters, joint gradients, or any centralized training pipeline.

Reinforcement learning for reputation emergence. Two recent studies apply reinforcement learning directly to indirect-reciprocity games. Anastassacos et al. (2021) train RL agents in a donation game with public reputation labels and decompose the problem into two coupled coordination

tasks: agents must learn both how to interpret existing reputations and which social norm to assign reputations with. Empirically, cooperation fails to emerge from scratch in their base setting; they propose two remedies: (i) *seed agents*, a fixed-strategy minority that steers the rest of the population toward a good equilibrium, and (ii) *introspective rewards*, an intrinsic payoff proportional to how well an agent's own strategy performs against a copy of itself. This is a strong negative signal for end-to-end reputation learning without auxiliary scaffolds. Ren et al. (2025) propose LR2, a bottom-up scheme in which the training reward shapes the environmental payoff by the agent's own reputation. The published code[2] ships two branches and uses the additive one as default (`repu_reward_coef`$=4$):

$$r^i = \begin{cases} r^i_{\text{env}} + \kappa\, P^i_t & \text{if } \kappa \geq 1 \quad \text{(default, } \kappa{=}4\text{)}, \\ \kappa\, r^i_{\text{env}} + (1{-}\kappa)\, P^i_t\, r^i_{\text{env}} & \text{if } \kappa < 1 \quad \text{(matches Eq. 5 of the paper)}, \end{cases}$$

where $P^i_t = \alpha P^i_{t-1} + (1{-}\alpha)\,\text{mean}(\text{neighbour assessments})$ is an EMA reputation with $\alpha{=}0.5$ in the repo. The paper's exposition foregrounds the multiplicative branch, but the experimental default is the additive one; we follow the repo default in our reproduction. LR2 reports cooperation in matrix-game social dilemmas. The scaffold, however, hard-codes precisely the claim the paper intends to demonstrate, namely that higher reputation is beneficial, by wiring reputation into the reward. Our reciprocity gradient differs from both methods in a sharp way: the learner's reward is the unmodified donation-game payoff, the gossip signal carries no semantics at initialization, and no intrinsic reward, seed agent, or reputation-shaped reward is used. Cooperation, discriminative action rules, and informative signal schemes all arise from analytic gradient propagation through reputation alone.

Analytic and evolutionary models of indirect reciprocity. The canonical analyses of indirect reciprocity treat strategies as fixed lookup tables and study their evolution under replicator dynamics or related dynamics (Nowak & Sigmund, 1998, 2005; Ohtsuki & Iwasa, 2004, 2006). Santos et al. (2021) provides a recent synthesis of this line, emphasizing the cognitive demands of reputation-based cooperation and the role of social-norm complexity. Masuda & Nakamura (2012) analyze a two-role trust game in which buyers choose whether to trust and sellers choose whether to cooperate, with sellers carrying a reputation score that buyers update at no cost. They show under replicator dynamics that zero-cost evaluation can still sustain an evolutionarily stable cooperative equilibrium coexisting with an asocial equilibrium, a density-dependent non-matrix structure. Hilbe et al. (2018) relax the classical public-assessment assumption and analyze indirect reciprocity under private, noisy, and incomplete information, identifying regimes where the leading-eight rules lose stability once observations diverge across agents. Our formulation is complementary rather than competing: instead of enumerating discrete strategies and testing stability, we keep actions and signals continuous and differentiable and treat the leading-eight rules (L3, L6) as hard-coded *opponents* against which a learning agent performs best response.

LLM-driven gossip and indirect reciprocity. A complementary line of work replaces hand-crafted strategies with pretrained large language models that play repeated donation games and emit free-form gossip. Zhu et al. (2026) place self-interested LLM agents in a population that talks, judges, and cooperates, and find that natural-language gossip can sustain cooperation when each agent reads the messages of others before deciding how to act. The cooperative behavior in that setting rides on social-reasoning capacity already encoded in the pretrained model, so the experiments isolate what gossip buys *given* a competent reasoner rather than what a tabula-rasa learner can recover from payoffs alone. Our setting is the strict inversion: the action and signal networks are randomly initialized, the gossip channel carries a single scalar with no a priori semantics, and the only training signal is the donation-game payoff. The two formulations therefore answer different questions: Zhu et al. (2026) ask whether linguistic gossip can coordinate already-capable agents, while we ask whether reputation-mediated cooperation can be *discovered* from gradients alone. The two perspectives are compatible, and a learning agent like ours could in principle be dropped into the LLM-population setting as a non-linguistic baseline.

Experimental reputation systems and gossip. A behavioral-economics literature studies how human participants use reputation and gossip in laboratory trust or donation games. Sommerfeld et al. (2007) show that gossip can substitute for direct observation in indirect-reciprocity games:

[2] https://github.com/itstyren/LR2.

participants condition behavior on third-party reports even when those reports are noisy. Fehr & Sutter (2019) find that allowing a third-party observer to send a free-form gossip message about a trustee raises trust and trustworthiness relative to a no-gossip baseline, though a substantial share of the effect is attributable to the mere presence of an observer rather than to message content. These experiments motivate the structural role that gossip plays in our setting (one agent rates another after interacting, and that rating is read by a future partner) but do not prescribe a learning rule. We inherit the same interaction schema and ask what a gradient-based learner can recover inside it without being told that gossip is valuable. Earlier reputation-system experiments such as Keser (2003) likewise establish that reputation mechanisms raise efficiency in repeated trust games; this line informs the setting rather than the algorithmic formulation.

B Background: Indirect Reciprocity, Gossip, and the Leading Eight

This appendix restates the evolutionary-game-theory background we assume in the main text. It is not new material; it is included for readers less familiar with the indirect-reciprocity literature, and to fix the exact notions of "first-order," "second-order," and "leading-eight" that we rely on.

Direct and indirect reciprocity. Two mechanisms sustain cooperation among self-interested agents. In *direct reciprocity*, two agents interact repeatedly and condition their behavior on their shared history; an agent cooperates because the same partner may reciprocate later (Trivers, 1971). In *indirect reciprocity*, agents rarely meet the same partner twice, so cooperation propagates through reputation: *third parties* reward an agent for cooperative behavior they observed earlier (Alexander, 2017; Nowak & Sigmund, 1998, 2005). Indirect reciprocity therefore requires a mechanism for transmitting information about past behavior across the population.

Gossip and reputation. The transmission mechanism is *gossip*: after observing a donor's action, the recipient (or a bystander) emits an evaluative signal about the donor. This signal joins the donor's *reputation*, a public record that future partners consult when deciding whether to cooperate. Two intertwined decisions form the strategic core of indirect reciprocity: how to *act* given a partner's reputation (the action policy), and how to *assess* a donor's behavior (the gossip policy). Both decisions simultaneously determine the population's reputational dynamics.

First-order reputation (image scoring). Under first-order assessment, the gossip signal depends only on the donor's action: cooperation produces a good signal, defection a bad one. This *image scoring* rule (Nowak & Sigmund, 1998) is evolutionarily unstable. It cannot distinguish justified defection (refusing to help a known defector) from unjustified defection (refusing to help a cooperator), leaving it vulnerable to invasion by unconditional cooperators and subsequent exploitation by defectors (Leimar & Hammerstein, 2001; Panchanathan & Boyd, 2003).

Second-order reputation and the leading eight. Second-order assessment lets the gossip signal condition on both the donor's action *and* the recipient's reputation, enabling *justified punishment*: a donor who defects against a bad-standing recipient retains a good reputation. Ohtsuki & Iwasa (2004, 2006) enumerated the second-order rules that are simultaneously stable, self-consistent, and resistant to invasion by defectors, and identified exactly eight. These *leading eight* norms represent the minimal sufficient complexity for robust indirect reciprocity in classical models.

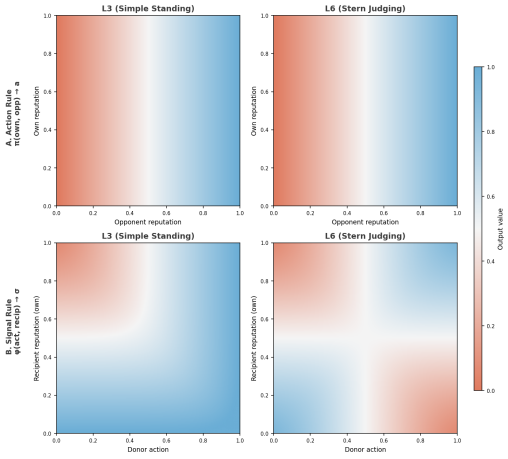
L3 (Simple Standing) and L6 (Stern Judging). The experiments in the main text use two specific leading-eight rules as test-bed opponents (Figure 4 for the discrete tables and the continuous tanh-relaxations). A second-order assessment is determined by the pair (a, s_r), where $a \in \{C, D\}$ is the donor's action and $s_r \in \{\text{Good}, \text{Bad}\}$ is the recipient's standing. The two rules differ in a single setting:

	L3 (Simple Standing)		**L6 (Stern Judging)**	
$s_r \backslash a$	C	D	C	D
Good (G)	G	B	G	B
Bad (B)	G	G	B	G

L3 is *forgiving*: every action toward a bad-standing recipient earns a good signal, so justified defection against a known defector is rewarded. L6 (*Stern Judging*) (Ohtsuki & Iwasa, 2004, 2006) is *strict*: cooperating toward a bad-standing recipient itself attracts a bad signal, in addition to defecting against a good-standing recipient. The L6 entry $(C, \text{Bad}) \to B$ is a quadrant flip, in which the cooperation assessment changes sign with the recipient's standing, and any sufficiently expressive opponent surrogate has to recover this from observations. The two rules anchor opposite ends of the rigor–leniency spectrum within the leading eight. We use L6 in the action-network setting (Setting (A.1), Setting (B.1)) because its quadrant structure carves a rectangular decision boundary into reputation space and so admits a sharp discriminative best response. We use L3 in the signal-network setting (Setting (A.2), Setting (B.2)) because its forgivingness preserves the reciprocity-gradient channel through the learner's justified-defection move, which a strict norm would destroy. The continuous, differentiable relaxations of these discrete tables that we use as opponents in practice are derived in Appendix N.



(a) Discrete tables. **Top:** action rule $\pi(s_{\text{own}}, s_{\text{opp}}) \in \{C, D\}$. **Bottom:** second-order assessment rule $\varphi(a_{\text{donor}}, s_{\text{recip}}) \in \{G, B\}$. The two norms differ only in the $(C, \text{Bad}) \to B$ vs. G entry of the assessment rule.



(b) Continuous tanh-relaxations used as differentiable opponents. **Top row:** action rule output as a function of $(s_{\text{opp}}, s_{\text{own}}) \in [0, 1]^2$. **Bottom row:** signal rule output as 2a function of $(a_{\text{donor}}, s_{\text{recip}}) \in [0, 1]^2$. The L6 quadrant flip is visible as a smooth diagonal switch in the bottom-right panel.

Figure 4: **L3 (Simple Standing) and L6 (Stern Judging) at a glance.** Discrete rule tables (panel a) and the continuous tanh-relaxations actually used by the differentiable opponents (panel b). The two assessment rules agree on three of four cells; the single disagreement, the cell highlighted by the diagonal switch in L6's bottom-right heatmap, drives the entire qualitative difference between the two norms.

C Concrete Influence-Channel Examples

This appendix expands the two influence channels referenced in Section 4.1 and gives a worked L6 example of reputation-mediated credit assignment. Each example traces a single channel through the coupled web of reputation updates that links a current action or signal of agent i to some future payoff of i. The examples are representative rather than exhaustive: the two channels are not separable, both policies simultaneously affect both sides of i's future payoff (benefits collected as recipient and costs paid as donor) through the same web, and in any realistic network these channels multiply combinatorially with one another and with many more channels not traced here.

C.1 Influence Through the Action Policy π^i

When a future donor conditions its action on i's reputation s^i, a change in i's earlier donation can affect the benefit it later receives. With opponent policies fixed, the recipient j assesses i's donation and appends $\sigma^{i \leftarrow j}$ to i's history. The resulting change in s^i can then alter a future donor k's action

toward i:

$$\pi^i \longrightarrow a^{i\to j} \longrightarrow \sigma^{i\leftarrow j} \longrightarrow s^i \longrightarrow \pi^k(s^i) \longrightarrow r^i_{t'}, \qquad t' > t.$$

This is one action-mediated path, rather than the only way i can influence its own reputation. Changes in i's signal policy can also feed back through later interactions, and higher-order assessments depend on reputations as well as actions.

The future interaction $k \to i$ can extend the same chain. Here i, the current recipient, emits $\sigma^{k\leftarrow i}$ about donor k, updating s^k. If i later acts as donor toward k, its action may depend on this updated s^k, changing its donation cost. Thus, the initial donation can affect both a later benefit and further downstream costs through successive action, assessment, and reputation updates.

C.2 Influence Through the Signal Policy φ^i

A structurally different channel runs through φ^i, the signal policy i uses when i itself is the recipient of someone else's action. Suppose donor j interacts with i at step t, and i via φ^i emits the signal $\sigma^{j\leftarrow i}_t$ that updates j's reputation score s^j. The modified s^j now sits inside the shared reputation state of the game, and two kinds of feedback return to i's future payoff from this modification.

Direct feedback. If i later acts as donor toward j, its action can read the modified s^j, changing the cost it pays. If i instead receives a donation from j, a direct benefit effect through s^j requires j's action policy to read its own reputation, as in the ProudCoop and HybridCoop settings. For a recipient-reputation-only rule $\pi^j(s^i)$, changing s^j alone does not change that donation while s^i is held fixed. In this $j \to i$ interaction, it is i that emits $\sigma^{j\leftarrow i}$; under a second-order assessment, the signal depends on j's action and the recipient reputation s^i. This assessment is emitted after the current payoff and can affect subsequent interactions. The direct cost and benefit routes above involve j's reputation without an additional intermediary agent.

Indirect feedback. Suppose a third agent k later acts as donor toward j. The modified s^j can change k's action, and recipient j then emits $\sigma^{k\leftarrow j}$ about k, updating s^k. Under a second-order assessment, s^j also enters j's evaluation directly as the recipient's own reputation. When i later donates to k, this updated s^k can change i's action and hence its cost. If i instead receives from k, a direct effect through s^k requires k's action rule to depend on its own reputation; a rule reading only s^i has no such direct dependence. This route adds k as an intermediary beyond j. The existence and strength of each effect depend on the policies' input sensitivities, not on the assessment order alone.

Arbitrary-depth recursion. The two examples above stop at one or two intermediate agents, but the mechanism is recursive with arbitrary depth. Each interaction along the chain modifies some agent's reputation, which enters the next interaction's decision, which emits a new signal that reshapes yet another reputation, which enters the interaction after that, and so on. In any realistic network of agents these chains multiply combinatorially: every action or signal i emits at the current step triggers a branching tree of downstream reputation updates, many of which eventually feed back into i's own future rewards through paths the two examples above do not cover. The influence attribution problem identified in Section 4.1 is the problem of accounting for all such paths simultaneously on top of the immediate payoff of the current action.

C.3 A Minimal Example of Influence Attribution

We now make the preceding influence-channel discussion concrete with an L6 example using the interaction order in the left panel of Figure 2, showing how an action affects a later payoff through reputation and how stopping this gradient path changes the learning signal.

The action-only L6 configuration is introduced in Section 5 and summarized as Settings (A.1) and (B.1) in Table 2; its oracle- and learned-opponent versions are reported in Appendices J.2 and J.3, respectively.

There are three distinct agents i, j, k, with the following action policies and signal rules:

- **Learner's action policy.** Agent i uses the trainable policy $a = \pi^i_{\theta^i}(s)$, where s is the recipient's reputation. Only θ^i is updated.

- **Learner's signal rule.** Agent i uses the fixed identity rule $\varphi^i(a) = a$, assigning a signal equal to the observed donor action.
- **Opponents' action policies.** Each opponent $\ell \in \{j, k\}$ uses the fixed rule $\pi^\ell(s) = s$, cooperating in proportion to the recipient's reputation.
- **Opponents' signal rules.** Each opponent follows Stern Judging (L6). In its discrete form, let $a = 1$ denote cooperation and $a = 0$ defection, and let $s = 1$ denote good recipient standing and $s = 0$ bad standing. The assessment is

$$F_{\mathrm{L6}}^{\mathrm{disc}}(a, s) = \begin{cases} 1, & a = s, \\ 0, & a \neq s, \end{cases} \qquad a, s \in \{0, 1\}.$$

Here $a = s$ means either cooperation with a good-standing recipient ($(a, s) = (1, 1)$) or defection against a bad-standing recipient ($(a, s) = (0, 0)$). Both receive a good assessment, encoded as 1. Conversely, $a \neq s$ means defection against a good-standing recipient or cooperation with a bad-standing recipient; both receive a bad assessment, encoded as 0.

Equivalently, the discrete rule can be written algebraically as

$$F_{\mathrm{L6}}^{\mathrm{disc}}(a, s) = as + (1 - a)(1 - s) = \frac{1}{2}\big[1 + (2a - 1)(2s - 1)\big].$$

To differentiate through this assessment, we use a continuous relaxation with donor action $a \in [0, 1]$ and recipient reputation $s \in [0, 1]$. Following Appendix N, we replace the binary signed values $2a - 1$ and $2s - 1$ by the smooth maps $\tanh(\beta(a - \frac{1}{2}))$ and $\tanh(\beta(s - \frac{1}{2}))$, respectively. Their product is positive when action and reputation lie on the same side of $\frac{1}{2}$ and negative when they lie on opposite sides. Rescaling this product to $[0, 1]$ gives the fixed differentiable signal rule

$$\varphi^\ell(a, s) = F_{\mathrm{L6}}(a, s) = \frac{1}{2}\left[1 + \tanh\big(\beta(a - \tfrac{1}{2})\big)\tanh\big(\beta(s - \tfrac{1}{2})\big)\right], \qquad \beta > 0.$$

L6 evaluates both the donor's action and the recipient's reputation, providing a second-order instance of the dependency graph discussed in Section 4.2. The experiments use $\beta = 5$ (Appendix J.3); the calculation below holds for any $\beta > 0$.

To isolate the credit-assignment mechanism, consider an analytical fresh-history example using these true fixed opponent functions. Condition on the interaction order shown in the left panel of Figure 2: $i \to k$ at $t = 0$, followed by $j \to i$ at $t = 1$, with initial recipient reputation s_0^k independent of θ^i. Here $t = 0, 1$ are local interaction indices, and s_t^ℓ denotes agent ℓ's reputation immediately before interaction t; rewards follow Section 3. Assume that i has no prior assessments and use the mean aggregator, so its first assessment determines s_1^i. This fresh-history simplification removes the initialized assessments used in the multi-interaction experiments; the calculation exposes a single channel through the true rules, while Setting (B.1) estimates such channels using learned surrogates. The learner's fixed signal is emitted after the second interaction's payoff and does not affect this two-interaction return.

At $t = 0$, agent i acts as donor:

$$a_0^{i \to k} = \pi_{\theta^i}^i(s_0^k), \qquad r_0^i = -c\, a_0^{i \to k}. \tag{7}$$

The recipient k evaluates this action using its fixed Stern Judging rule:

$$s_1^i = \sigma_0^{i \leftarrow k} = F_{\mathrm{L6}}\big(a_0^{i \to k}, s_0^k\big). \tag{8}$$

At $t = 1$, the fixed donor j cooperates in proportion to i's updated reputation:

$$a_1^{j \to i} = \pi^j(s_1^i) = s_1^i, \qquad r_1^i = b\, a_1^{j \to i}. \tag{9}$$

Thus, i's return over the two interactions is

$$R_{0:1}^i = -c\, a_0^{i \to k} + bF_{\mathrm{L6}}\big(a_0^{i \to k}, s_0^k\big). \tag{10}$$

Interpreting the Gradient Chain. Each derivative measures a local response: for a small change in its input, it gives the first-order change in its output, holding the initial conditions and fixed opponent rules unchanged. In particular, $\partial s_1^i / \partial a_0^{i\to k}$ measures how changing i's donation changes k's assessment and hence i's reputation. The derivative $\partial a_1^{j\to i}/\partial s_1^i = 1$ measures how strongly the future donor j responds to that reputation, while $\partial r_1^i / \partial a_1^{j\to i} = b$ converts the change in j's donation into a change in i's benefit. Their product traces the delayed payoff effect back to the initial action:

$$\frac{\partial r_1^i}{\partial a_0^{i\to k}} = \underbrace{\frac{\partial r_1^i}{\partial a_1^{j\to i}}}_{b} \underbrace{\frac{\partial a_1^{j\to i}}{\partial s_1^i}}_{1} \underbrace{\frac{\partial s_1^i}{\partial a_0^{i\to k}}}_{\partial F_{\mathrm{L6}}/\partial a}.$$

For example, a small increase δa changes the later benefit by approximately $b(\partial F_{\mathrm{L6}}/\partial a)\delta a$, while changing the immediate payoff by $-c\,\delta a$. Adding these effects gives the marginal change in the learner's own return:

$$\frac{\partial R_{0:1}^i}{\partial a_0^{i\to k}} = -c + b\frac{\partial F_{\mathrm{L6}}(a_0^{i\to k}, s_0^k)}{\partial a_0^{i\to k}} \tag{11}$$

$$= -c + \frac{b\beta}{2}\,\mathrm{sech}^2\big(\beta(a_0^{i\to k} - \tfrac{1}{2})\big)\tanh\big(\beta(s_0^k - \tfrac{1}{2})\big)\,. \tag{12}$$

For $s_0^k > \frac{1}{2}$, the mediated term is positive and, when it exceeds c, provides a local incentive to increase cooperation. For $s_0^k < \frac{1}{2}$, the mediated term is negative: cooperating both incurs the immediate cost and worsens the donor's subsequent reputation, providing an incentive to reduce cooperation. The corresponding policy-parameter gradient is

$$\nabla_{\theta^i} R_{0:1}^i = \frac{\partial R_{0:1}^i}{\partial a_0^{i\to k}} \nabla_{\theta^i} \pi_{\theta^i}^i(s_0^k). \tag{13}$$

Here $\nabla_{\theta^i}\pi_{\theta^i}^i(s_0^k)$ describes how changing the learner's own parameters changes its donation; multiplying by $\partial R_{0:1}^i/\partial a_0^{i\to k}$ assigns each parameter the resulting marginal effect on return. For this conditional two-interaction objective, a gradient-ascent step $\Delta\theta^i = \alpha\nabla_{\theta^i} R_{0:1}^i$, with sufficiently small $\alpha > 0$, changes the action to first order by

$$\Delta a_0^{i\to k} \approx \alpha\,\frac{\partial R_{0:1}^i}{\partial a_0^{i\to k}}\,\big\|\nabla_{\theta^i}\pi_{\theta^i}^i(s_0^k)\big\|^2\,.$$

Thus, the update locally increases the donation when its marginal return is positive and decreases it when that return is negative, provided the policy has a nonzero parameter gradient. Agent i changes only its own policy parameters: k's assessment rule and j's action rule remain fixed. Their outputs nevertheless change in response to i's changed action and reputation, and the backward calculation tells i how to use this response to improve its own return.

This contrast illustrates the mechanism underlying the high- versus low-reputation action profiles in Figure 10 (Setting (A.1), Appendix J.2): cooperation with high-reputation recipients and justified defection against low-reputation recipients. The example isolates one conditional credit-assignment path rather than characterizing the optimum or the full expected-return gradient of the multi-interaction training problem.

In the full training problem, the objective is $J^i = \mathbb{E}_\xi[\sum_t r_t^i]$ (Section 3). Under the differentiation conditions stated in Appendix E, its gradient aggregates such contributions over complete interaction trajectories and their matching probabilities. Gradient ascent on J^i therefore aims to improve expected self-return by changing the learner's behavior and accounting for the resulting changes in others' assessments, reputations, and later actions. The two-interaction calculation shows one contribution to that update; a shared policy must combine contributions from different states and paths. In the learned-opponent setting, these sensitivities are computed through surrogates (Section 4.3), and their accuracy depends on how well those models capture the relevant responses.

For comparison, retain the same forward interaction but stop the backward path through s_1^i. The resulting backward derivatives are

$$\left.\frac{\partial s_1^i}{\partial a_0^{i\to k}}\right|_{\mathrm{StopGradient}} = 0, \qquad \left.\frac{\partial R_{0:1}^i}{\partial a_0^{i\to k}}\right|_{\mathrm{StopGradient}} = -c. \tag{14}$$

Both recipient contexts then expose only the immediate cooperation cost to the policy update. Although the forward reputation-mediated benefit remains present, its gradient no longer supplies the context-dependent learning signal in Equation 12.

D Backward Gradient Graph: Visualization and Numerical Verification

This appendix documents a minimal three-agent demo that (i) visualizes the autograd computation graph the reciprocity gradient traverses and (ii) verifies numerically that PyTorch backpropagation crosses agent boundaries without any analytic surgery.

Setup. We instantiate $N = 3$ agents indexed $i = 0$, $j = 1$, $k = 2$, each implemented as a self-contained `DemoAgent` with a tiny Markov-approximation architecture (memory window $= 1$, i.e. observations consist only of the *last* reputation entry of the target). The action net is $\mathrm{MLP}_{1\to(10+\mathrm{idx})\to1}$ with a `Tanh` hidden layer and a `Sigmoid` output, producing an action $a \in [0, 1]$. The gossip (signal) net is $\mathrm{MLP}_{1\to(20+\mathrm{idx})\to1}$ with the same activation pattern, producing a signal $\sigma \in [0, 1]$. We use first-order gossip (`second_order = False`), so the signal net sees only the donor's action. Tiny agent-specific hidden widths (offset by agent index) make the three agents' parameter sets visually distinguishable in the rendered graph.

The environment is `ContinuousRandomMatchingDonationGame` with horizon $T = 3$, benefit $b = 2.0$, and a constant reputation initialization $r^{(0)} = 0.5$ for every agent. We fix the random seed to 42 and place all agents in `eval()` mode to disable `Dropout` so that the forward evaluations used in the finite-difference check are repeatable. The matching sequence is hard-coded to exercise the dependencies visualized below:

$$(t{=}0):\ i \to k, \qquad (t{=}1):\ j \to i, \qquad (t{=}2):\ i \to j.$$

The key hyperparameters are summarized in Table 1.

Setting	**Value**
Number of agents N	3
Horizon T	3
Benefit b	2.0
Cost c	1.0
Initial reputation	constant, $r_m^{(0)} = 0.5$
Signal dim	1
Reputation memory window	1 (Markov approximation)
Action net hidden	$(10 + \mathrm{idx},)$ with `Tanh+Sigmoid`
Signal net hidden	$(20 + \mathrm{idx},)$ with `Tanh+Sigmoid`
Reputation order	first-order (signal $= f(a)$)
Matching sequence	$(0{\to}2)$, $(1{\to}0)$, $(0{\to}1)$
Random seed	42
Mode	`eval()` (no `Dropout`)
Finite-difference step ε	10^{-4}

Table 1: Demo configuration used to generate Figures 5–7 and to verify backpropagation through the reciprocity graph.

Agent architecture. Listing 1 shows the `DemoAgent` class used in this demo. Each agent owns an `action_net` and a `signal_net`, both standard MLPs with `Sigmoid` outputs. The Markov approximation reads only the last entry of the reputation history.

Listing 1: DemoAgent: minimal agent with Markov-approximation action and signal nets.

```
class DemoAgent(nn.Module):
    def __init__(self, name, signal_dim=1, second_order=False):
        super().__init__()
        idx = int(name.split("_")[-1])
        # Agent-specific hidden widths for visual distinction in graph
        self.action_net = build_mlp(signal_dim, (10+idx,), 1,
                                    out_activation=nn.Sigmoid())
```

```
        signal_in = 1 + (signal_dim if second_order else 0)
        self.signal_net = build_mlp(signal_in, (20+idx,), 1,
                                    out_activation=nn.Sigmoid())
        self.register_buffer("reputation_history",
                             torch.empty((1, signal_dim)))

    def take_action(self, recipient_reputation):
        obs = recipient_reputation[-1].view(1, -1)  # Markov: last entry
        return self.action_net(obs).squeeze(0)

    def send_signal(self, donor_action, own_reputation=None):
        a = donor_action.view(1, 1)
        if self.second_order:
            inp = torch.cat([a, own_reputation[-1].view(1,-1)], dim=-1)
        else:
            inp = a
        return self.signal_net(inp).squeeze(0)

    def append_reputation(self, signal_t):
        self.reputation_history = torch.cat(
            [self.reputation_history, signal_t.view(1, -1)], dim=0)
```

Demo script. Listing 2 shows the core of the demo: running one episode with a fixed matching sequence, extracting the autograd gradient $\partial R^0/\partial\theta$, and comparing against a finite-difference estimate. All quantities retain their computation graph because `ContinuousRandomMatchingDonationGame` never calls `.detach()` on intermediate rewards or signals.

Listing 2: Core demo: episode rollout, autograd gradient, and finite-difference verification.

```
torch.manual_seed(42)
agents = [DemoAgent(name=f"Agent_{i}") for i in range(3)]
for a in agents:
    a.eval()

env = ContinuousRandomMatchingDonationGame(
    agents=agents, horizon=3, b=2.0,
    initial_reputation_dist="constant",
    initial_reputation_params={"value": 0.5})

# Fixed matching: t=0: i->k, t=1: j->i, t=2: i->j
fixed_matching = [(0, 2), (1, 0), (0, 1)]
rewards, _, history = env.play_episode(
    matching_sequence=fixed_matching)
R0 = rewards[0]  # Agent 0's cumulative reward (with grad)

# Autograd: differentiate R0 w.r.t. one scalar weight
target = agents[0].action_net[0].weight
grad_auto = torch.autograd.grad(R0, target,
                                retain_graph=True)
g_auto = grad_auto[0][0, 0].item()  # => 0.010034

# Finite difference: perturb the same weight by epsilon
eps = 1e-4
# ... (re-create agents, load weights, perturb by +eps,
#      re-run episode, compute (R0_new - R0_base) / eps)
g_fd = (R0_plus - R0_base) / eps    # => 0.010133

error = abs(g_auto - g_fd)          # => 9.9e-05 < 1e-3
```

Rendered graphs. Figures 5–7 show the rendered computation graphs produced by `torchviz`. We color Agent 0's parameter nodes pink, Agent 1's green, Agent 2's blue, and group each agent's action-net and signal-net parameters in their own cluster so that the reader can visually track how the backward pass enters and exits each agent.

Figure 5 shows the graph used to differentiate Agent 0's action at $t = 2$, $a^i := a_{0\to1}$, with respect to its action-policy parameters θ^0. Write η^0 for its separate signal-policy parameters. The forward

influence chain starts with $a_{0\to 2}$, passes through Agent 2's signal $\sigma_0^{0\leftarrow 2}$ and Agent 1's action $a_{1\to 0}$, then through Agent 0's signal $\sigma_1^{1\leftarrow 0}$ to $a_{0\to 1}$. Backpropagation follows these dependencies in reverse. Because the action network is reused, θ^0 also affects the final action directly. This illustrates how an earlier action can influence a later input to the same agent's action policy through other agents' responses.

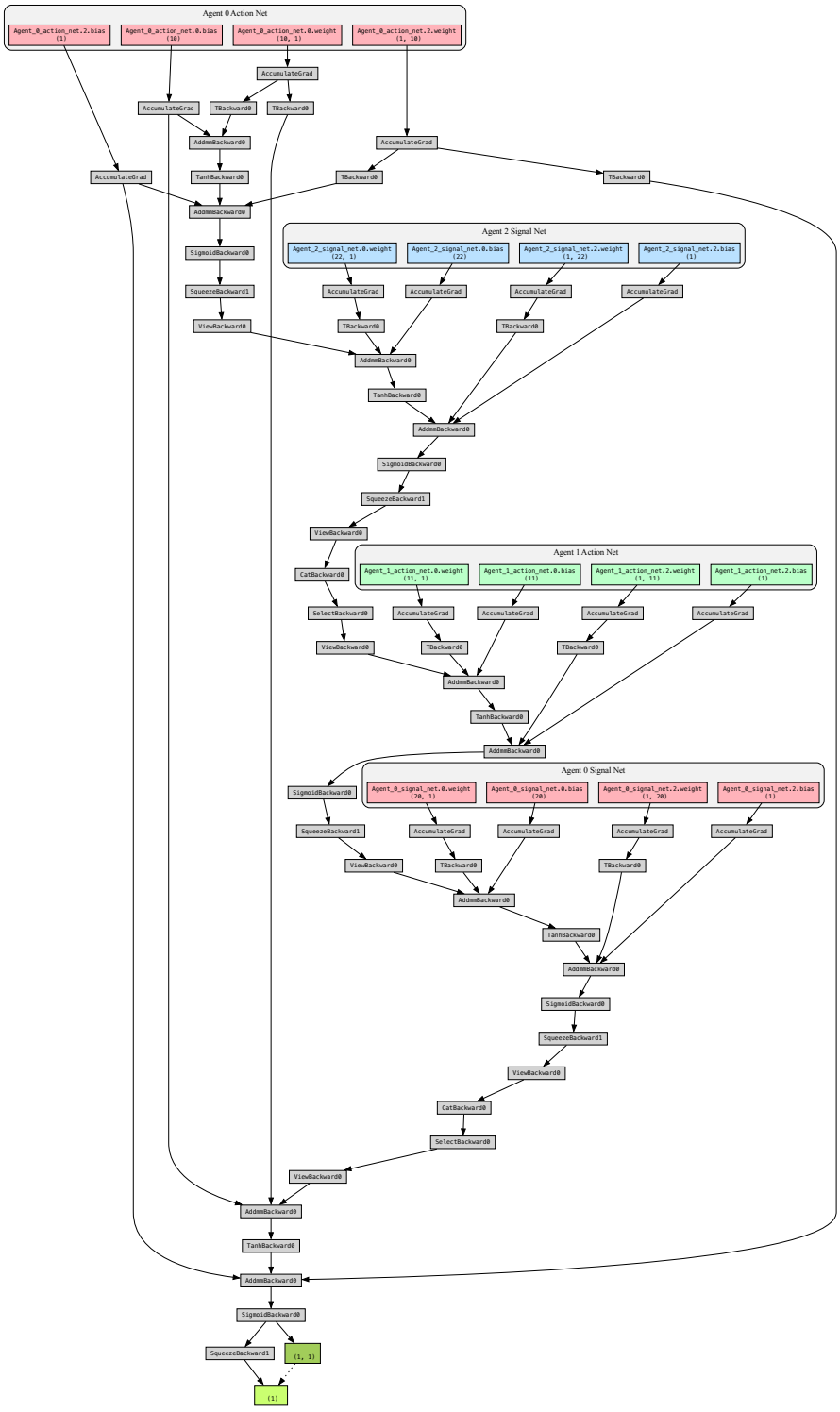


Figure 5: Computation graph used to obtain $\nabla_{\theta^0} a_{0\to 1}$, where $a_{0\to 1}$ is Agent 0's action at $t = 2$. The action parameters θ^0 affect this action directly and through the earlier action at $t = 0$, Agent 2's assessment, Agent 1's donation, and Agent 0's assessment.

Figure 6 shows the graph used to differentiate Agent 1's action at $t = 1$, $a^j := a_{1\to 0}$, with respect to θ^0. This action depends on Agent 0's parameters through the forward path $\theta^0 \to a_{0\to 2} \to \sigma_0^{0\leftarrow 2} \to s^0 \to a_{1\to 0}$. Agent 2's signal network and Agent 1's action network have fixed weights but transmit derivatives with respect to their inputs, allowing the gradient to return to Agent 0's earlier action parameters.

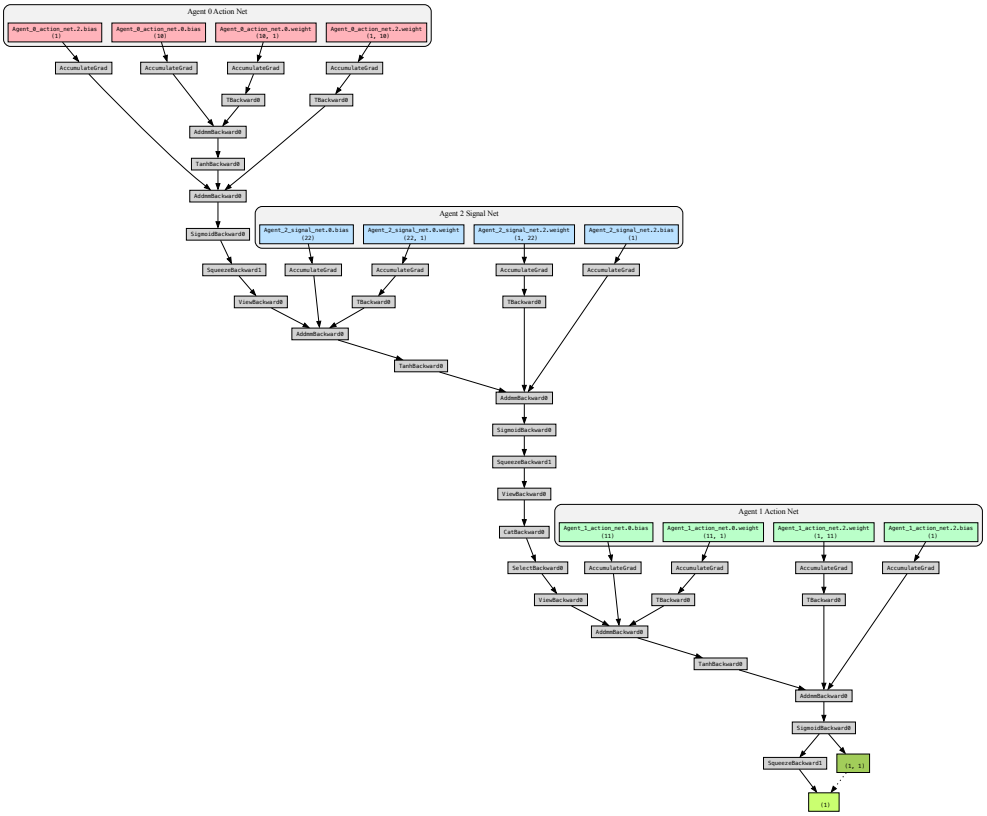


Figure 6: Computation graph used to obtain $\nabla_{\theta^0} a_{1\to 0}$, where $a_{1\to 0}$ is Agent 1's action at $t = 1$. Agent 0's action parameters influence this action through Agent 2's assessment and the resulting update to Agent 0's reputation.

Figure 7 shows the graph for R^0, from which the action- and signal-parameter gradients are obtained. Its connected network blocks are Agent 0's action and signal networks, Agent 1's action network, and Agent 2's signal network. Agent 2 never acts as donor in this sequence, so its action network is unused. Agent 1 emits $\sigma_2^{0\leftarrow 1}$ after the final payoff has been determined; this terminal assessment updates the history but has no path into the three-interaction return. The graph therefore contains only the network blocks that affect R^0.

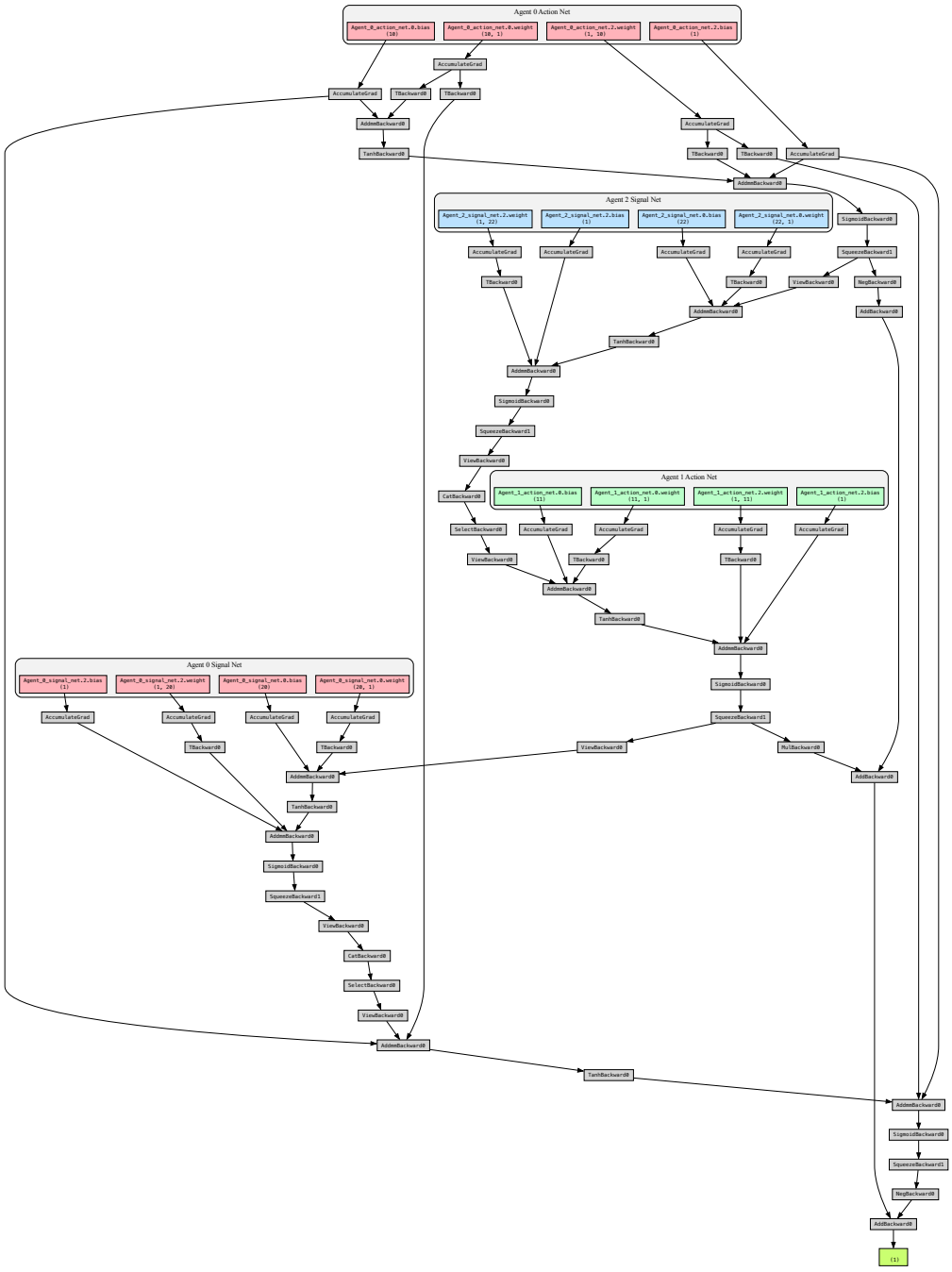


Figure 7: Computation graph for R^0 under the fixed matching $(0 \to 2)$, $(1 \to 0)$, $(0 \to 1)$, used to obtain $(\nabla_{\theta^0} R^0, \nabla_{\eta^0} R^0)$. Four network blocks affect the return: Agent 0's action and signal networks, Agent 1's action network, and Agent 2's signal network.

Cumulative reward. Under this matching sequence, Agent 0's cumulative reward decomposes as

$$R^0 \;=\; -\,c\,a_{0\to 2} \;+\; b\,a_{1\to 0} \;-\; c\,a_{0\to 1}.$$

With $c = 1$, $b = 2$ and the random seed above, the realized quantities are

$$a_{0\to 2} = 0.5030, \quad \sigma_0^{0\leftarrow 2} = 0.3931,$$
$$a_{1\to 0} = 0.5591, \quad \sigma_1^{1\leftarrow 0} = 0.4655,$$
$$a_{0\to 1} = 0.5002, \quad \sigma_2^{0\leftarrow 1} = 0.5560.$$

yielding $R^0 = -0.5030 + 1.1182 - 0.5002 = 0.1150$.

Parameter influence channels. Agent 0's action parameters θ^0 and signal parameters η^0 influence R^0 through the following five stages. These stages describe overlapping paths in the computation graph:

1. **Own action at t=0.** θ^0 determines $a_{0\to 2}$, which directly contributes $-c\, a_{0\to 2}$ to R^0.
2. **Signal back on oneself.** $a_{0\to 2}$ is fed into Agent 2's gossip net, producing $\sigma_0^{0\leftarrow 2}$, which is appended to Agent 0's reputation history.
3. **Future action received.** At t=1, Agent 1's action net reads Agent 0's updated reputation and returns $a_{1\to 0}$, contributing $b\, a_{1\to 0}$ to R^0.
4. **Gossip author channel.** At t=1, Agent 0's signal net produces $\sigma_1^{1\leftarrow 0}$, which is appended to Agent 1's history. This signal depends directly on η^0 and indirectly on θ^0 through its input $a_{1\to 0}$.
5. **Own action at t=2.** At t=2, Agent 0's action net reads Agent 1's updated reputation (which carries $\sigma_1^{1\leftarrow 0}$) and returns $a_{0\to 1}$, contributing $-c\, a_{0\to 1}$ to R^0. Both its direct dependence on θ^0 and the dependence through its reputation input contribute to the gradient.

Stages (2)–(3) carry the action-mediated effect into a future benefit; stages (4)–(5) carry an assessment-mediated effect into a later donation cost. They are connected parts of the same chain, and automatic differentiation combines their contributions with the local action-cost derivatives (Section 4.2).

Numerical verification. Using `torch.autograd.grad(R0, action_net[0].weight)`, we read off the $(0, 0)$-entry of the analytic gradient:

$$\left.\frac{\partial R^0}{\partial\, \texttt{action_net[0].weight}[0,0]}\right|_{\text{autograd}} = 0.010034.$$

We then re-run the full episode with the same seed and matching sequence after perturbing the same scalar weight by $\varepsilon = 10^{-4}$, and form the finite-difference estimate $\widehat{g} = (R^0(+\varepsilon) - R^0(0))/\varepsilon$:

$$R^0(0) = 0.114991, \qquad R^0(+\varepsilon) = 0.114992, \qquad \widehat{g} = 0.010133.$$

The absolute error is $|0.010034 - 0.010133| = 9.9 \times 10^{-5}$, below our 10^{-3} acceptance threshold. For this scalar parameter, the finite-difference estimate agrees with autograd within the stated tolerance. The derivative is obtained through the ordinary rollout graph without a custom backward hook, score-function estimator, or surrogate loss.

E Reciprocity Gradient: Full Derivation and Generality

This appendix gives (i) the step-by-step donation-game derivation of the reciprocity gradient used in Section 4, (ii) a short survey of the game families used in the reputation literature, (iii) an account of why the donation-game choice is the canonical one for the branches we build on, and (iv) the extension of the derivation to continuous-deterministic general-sum games.

E.1 Step-by-Step Donation-Game Derivation

Let $\mathcal{M}$ denote the finite sample space of matching sequences over the horizon T, and let $P_\xi(m)$ denote the joint probability of sequence $m \in \mathcal{M}$ under ξ. The role indicator at step t is a function of ξ alone, statistically independent of the reputation histories that drive the action evaluations, and ξ itself depends on no policy parameter. Under these conditions, the total derivative with respect to π^i can be pushed all the way inside the expectation:

$$
\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\left[\sum_{t=1}^{T} r_t^i\right] &= \frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\Bigg[\sum_{t=1}^{T}\Big(-c\cdot\sum_{j\neq i}\mathbb{1}(i=\mathsf{D}\wedge j=\mathsf{R})\cdot\pi^i(s_t^j) \\
&\qquad\qquad + b\cdot\sum_{j\neq i}\mathbb{1}(j=\mathsf{D}\wedge i=\mathsf{R})\cdot\pi^j(s_t^i)\Big)\Bigg] \\
&= \frac{\mathrm{d}}{\mathrm{d}\theta^i}\sum_{t=1}^{T}\Big(-c\cdot\sum_{j\neq i}\underbrace{\mathbb{E}_\xi\big[\mathbb{1}(i=\mathsf{D}\wedge j=\mathsf{R})\big]}_{=\frac{1}{N(N-1)}}\cdot\mathbb{E}_\xi\big[\pi^i(s_t^j)\big] \\
&\qquad\qquad + b\cdot\sum_{j\neq i}\underbrace{\mathbb{E}_\xi\big[\mathbb{1}(j=\mathsf{D}\wedge i=\mathsf{R})\big]}_{=\frac{1}{N(N-1)}}\cdot\mathbb{E}_\xi\big[\pi^j(s_t^i)\big]\Big) \\
&= \frac{1}{N(N-1)}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\left[\sum_{t=1}^{T}\Big(-c\cdot\sum_{j\neq i}\pi^i(s_t^j)+b\cdot\sum_{j\neq i}\pi^j(s_t^i)\Big)\right] \\
&= \frac{1}{N(N-1)}\sum_{m\in\mathcal{M}}P_\xi(m)\frac{\mathrm{d}}{\mathrm{d}\theta^i}\left[\sum_{t=1}^{T}\Big(-c\cdot\sum_{j\neq i}\pi^i(s_t^j)+b\cdot\sum_{j\neq i}\pi^j(s_t^i)\Big)\right] \\
&= \frac{1}{N(N-1)}\mathbb{E}_\xi\left[\sum_{t=1}^{T}\Big(-c\cdot\sum_{j\neq i}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^i(s_t^j)+b\cdot\sum_{j\neq i}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^j(s_t^i)\Big)\right]
\end{aligned}
$$

Equality (i) substitutes the role-indicator form of the per-step reward from Section 3. Equality (ii) moves the expectation inside the per-step sums and factors each term as $\mathbb{E}_\xi[\mathbb{1}\cdot\pi]=\mathbb{E}_\xi[\mathbb{1}]\cdot\mathbb{E}_\xi[\pi]$ by the independence of the indicator from the reputation histories, with the indicator expectation evaluating to $1/N(N-1)$ for every ordered pair. Equality (iii) collapses the constant prefactor in front of the sum and repackages the per-step terms back inside a single expectation. Equality (iv) expands the expectation explicitly over the matching space $\mathcal{M}$ and moves the total derivative inside the finite sum, valid because $P_\xi(m)$ is policy-independent. Equality (v) repackages the sum over $\mathcal{M}$ back as an expectation. The same derivation carries through verbatim when θ^i is replaced by η^i, yielding the signal-policy gradient.

E.2 Detailed Gradient Expansion

Composing the chain-rule decomposition at a policy output with the reputation-to-reputation highway derivatives from Section 4.2 yields closed forms for the four parameter gradients that appear in the main-text objective. Throughout this subsection we factor out two *entry-channel* terms that capture how the parameter first enters the reputation graph:

$$
H_{\theta^i} := \sum_{k\neq i}\frac{\partial s^i}{\partial\sigma^{i\leftarrow k}}\cdot\frac{\mathrm{d}\sigma^{i\leftarrow k}}{\mathrm{d}a^{i\to k}}\cdot\frac{\partial a^{i\to k}}{\partial\theta^i}, \qquad H_{\eta^i}^k := \frac{\partial s^k}{\partial\sigma^{k\leftarrow i}}\cdot\frac{\partial\sigma^{k\leftarrow i}}{\partial\eta^i}.
$$

H_{θ^i} is a scalar channel that routes θ^i into i's own reputation s^i via every signal past recipients have emitted about i; $H_{\eta^i}^k$ is a per-agent channel that routes η^i into agent k's reputation s^k via the single signal i emitted about k.

Action parameter θ^i. The parameter θ^i governs only i's outgoing actions and reaches others only through s^i:

$$
\frac{\mathrm{d}\pi^i(s_t^j)}{\mathrm{d}\theta^i} = \underbrace{\frac{\partial\pi^i(s_t^j)}{\partial\theta^i}}_{\text{local}} + \underbrace{\frac{\partial\pi^i(s_t^j)}{\partial s_t^j}\cdot\frac{\mathrm{d}s_t^j}{\mathrm{d}s^i}\cdot H_{\theta^i}}_{\text{via } i\text{'s reputation}}, \tag{15}
$$

$$
\frac{\mathrm{d}\pi^j(s_t^i)}{\mathrm{d}\theta^i} = \frac{\partial\pi^j(s_t^i)}{\partial s_t^i}\cdot\frac{\mathrm{d}s_t^i}{\mathrm{d}s^i}\cdot H_{\theta^i}. \tag{16}
$$

Equation 15 describes how i's own future action toward j shifts; the first summand is the direct response of π^i at fixed reputation, the second is the indirect response through i's own reputation chain. Equation 16 describes how a third party j's future action back toward i shifts, which happens solely through i's reputation.

Signal parameter η^i. The parameter η^i governs only the signals i emits about past donors k, reaching others' reputations through every s^k in parallel and not affecting i's own immediate actions:

$$\frac{\mathrm{d}\pi^i(s_t^j)}{\mathrm{d}\eta^i} = \frac{\partial \pi^i(s_t^j)}{\partial s_t^j} \cdot \sum_{k \neq i} \frac{\mathrm{d}s_t^j}{\mathrm{d}s^k} \cdot H^k_{\eta^i}, \tag{17}$$

$$\frac{\mathrm{d}\pi^j(s_t^i)}{\mathrm{d}\eta^i} = \frac{\partial \pi^j(s_t^i)}{\partial s_t^i} \cdot \sum_{k \neq i} \frac{\mathrm{d}s_t^i}{\mathrm{d}s^k} \cdot H^k_{\eta^i}. \tag{18}$$

Each term sums over all third parties k that i has gossiped about, because a single change in η^i perturbs every s^k simultaneously; the reputation-to-reputation derivatives $\mathrm{d}s_t^j/\mathrm{d}s^k$ and $\mathrm{d}s_t^i/\mathrm{d}s^k$ are the same highway transitions derived in Section 4.2. All four expressions are computed by automatic differentiation through the unrolled forward graph and require no hand-coded backward pass.

E.3 Game Families in the Reputation Literature

Research on reputation mechanisms spans evolutionary game theory, classical microeconomics, and multi-agent systems. Depending on the research emphasis, the literature uses three dominant game families.

Donation game. The donation game is a one-sided decision process: the donor chooses whether to pay cost c to provide benefit b to the recipient (with $b > c$), yielding payoffs $-c$ to the donor and b to the recipient in the cooperative case. It is the canonical benchmark in evolutionary game theory, indirect reciprocity, and the evolution of social norms. Nowak & Sigmund (1998) introduced image scoring in the donation game; Ohtsuki & Iwasa (2004, 2006) derived the leading eight in the donation game; and subsequent extensions to private, noisy, and incomplete information (Hilbe et al., 2018; Fujimoto & Ohtsuki, 2023; Schmid et al., 2023) all retain the donation-game payoff structure. The donation game's appeal is that it strips away direct reciprocity and isolates, in minimal parametric form, the exact question of how a third-party observer updates the donor's reputation from a single observed action.

Prisoner's Dilemma. The Prisoner's Dilemma (PD) is a symmetric, simultaneous game in which both agents choose cooperate (C) or defect (D), with payoffs $T > R > P > S$ and $2R > T + S$. It is the canonical model of cooperation under conflicting incentives in classical game theory and is used by recent RL-based indirect-reciprocity studies (Anastassacos et al., 2021; Ren et al., 2025). The donation game is in fact a special case of the PD: playing a simultaneous round of the donation game between two agents induces the PD payoff matrix with $R = b - c$, $T = b$, $S = -c$, $P = 0$, which satisfies the PD inequalities whenever $b > c > 0$.

Trust game. The trust (or investment) game is a sequential two-stage game in which an investor transfers an amount to a trustee, the transferred amount is multiplied by some factor k, and the trustee decides what fraction to return. It is the canonical model in the behavioral-economics literature on reputation in markets and rating systems (Masuda & Nakamura, 2012; Keser, 2003; Sommerfeld et al., 2007; Fehr & Sutter, 2019). The trust game emphasizes moral hazard under enforceable contracts and partner-specific reputation rather than the assessment-rule structure that indirect-reciprocity theory isolates.

E.4 Why the Donation-Game Choice Is Canonical

Our derivation targets the donation game, which is the canonical choice for the indirect-reciprocity and social-norm branches the paper sits in. Three reasons make this choice theoretically complete rather than restrictive.

Cleanest isolation of the indirect-reciprocity mechanism. The donation game has a two-parameter payoff structure (b, c), which lets the gradient derivation stay analytic and keeps the dependence of the reputation update on the donor's single action $a^{i\to j}$ fully explicit. Starting from a general-sum matrix game would entangle the reputation-update channel with direct payoff asymmetries, making it hard to attribute gradient flow to the indirect-reciprocity channel as opposed to direct payoff conflicts. The donation-game reduction isolates the mechanism under study.

Homomorphism to symmetric matrix games. Because the donation game embeds as the special PD case $(R, T, S, P) = (b - c, b, -c, 0)$, any equilibrium or evolutionary-stability result we obtain under donation-game dynamics transfers to the corresponding symmetric matrix game by the payoff parameterization. Every classical leading-eight stability result we benchmark against is stated for the donation game for the same reason: it admits a lossless affine mapping into the broader PD parameter family.

Alignment with the indirect-reciprocity comparison set. Every baseline and every classical evolutionary-stability result that our experiments touch (L3, L6, HybridCoop, AllDefector, Proud-Cooperator) is defined for the donation game. Using the same underlying game is what makes the empirical comparison legible to the indirect-reciprocity literature.

E.5 Extension to Continuous Deterministic General-Sum Games

Our main-text policies are already deterministic continuous maps rather than probability distributions over discrete actions, so the most faithful generalization of the derivation is not to a discrete-matrix game but to a continuous-action general-sum game with deterministic policies. Under this view the derivation stays inside the deterministic-policy-gradient (DPG) calculus (Silver et al., 2014; Lillicrap et al., 2015): instead of the log-derivative trick over a discrete distribution, we apply the multivariate chain rule directly to a continuously differentiable payoff function. Below we carry out this extension with no discrete-action scaffolding and keep the same notation as Appendix E.1.

Continuous deterministic setup. *Action space.* The action space is a continuous subset $\mathcal{A} \subseteq \mathbb{R}^d$ rather than a discrete set.

Policy as a deterministic map. The policy $\pi^i(s_t^j)$ is no longer a probability; it directly outputs the continuous action that agent i takes on observing state s_t^j (e.g. the continuous donation amount or the continuous signal intensity used throughout the main text). Hence $\pi^i(s_t^j) \in \mathcal{A}$ is a deterministic continuous map.

Continuously differentiable payoff. The per-step payoff is an arbitrary continuously differentiable function $u^i(a^i, a^j)$: when i plays $x \in \mathcal{A}$ and opponent j plays $y \in \mathcal{A}$, agent i receives $u^i(x, y)$. No symmetry, no matrix structure, no ordering of actions is assumed.

Reputation-chained opponent response. We treat every opponent's policy *function* π^j as fixed (no opponent *learning* model), but the policy *output* $\pi^j(s_t^i)$ still depends on agent i's own policy because the observation s_t^i is the reputation state of agent i, which accumulates signals emitted during prior interactions that themselves depend on π^i. The total derivative $\mathrm{d}\pi^j(s_t^i)/\mathrm{d}\pi^i$ is therefore non-zero in general; it factors as $(\partial\pi^j/\partial s_t^i) \cdot (\mathrm{d}s_t^i/\mathrm{d}\pi^i)$ and is precisely the gradient-highway channel of Section 4.2.

Matching. The matching law of Section 3 is retained, with the undirected interaction indicator $\mathbb{1}(i \leftrightarrow j)$ and $\mathbb{E}_\xi[\mathbb{1}(i \leftrightarrow j)] = 1/N(N-1)$.

Gradient derivation. In the deterministic continuous setting the per-interaction reward is just the payoff evaluated at the two deterministic policy outputs, $r_t^{i,j} = u^i\big(\pi^i(s_t^j), \pi^j(s_t^i)\big)$, and the expectation $\mathbb{E}_\xi$ only covers environmental randomness (matching draws and state draws), not discrete-action sampling. Applying the expectation-derivative interchange from Appendix E.1:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\left[\sum_{t=1}^{T} r_t^i\right] &= \frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\left[\sum_{t=1}^{T}\sum_{j\neq i}\mathbb{1}(i\leftrightarrow j)\cdot u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)\right]\\ &= \sum_{t=1}^{T}\sum_{j\neq i}\underbrace{\mathbb{E}_\xi\big[\mathbb{1}(i\leftrightarrow j)\big]}_{=\frac{1}{N(N-1)}}\cdot\mathbb{E}_\xi\left[\frac{\mathrm{d}}{\mathrm{d}\theta^i}u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)\right]\\ &= \frac{1}{N(N-1)}\mathbb{E}_\xi\left[\sum_{t=1}^{T}\sum_{j\neq i}\frac{\mathrm{d}}{\mathrm{d}\theta^i}u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)\right].\end{aligned}$$

To evaluate $\mathrm{d}u^i/\mathrm{d}\pi^i$, we apply the multivariate chain rule to the two inputs of the continuously differentiable payoff: the partial with respect to the first input (agent i's own continuous action $\pi^i(s_t^j)$) and the partial with respect to the second input (the opponent's continuous action $\pi^j(s_t^i)$):

$$\frac{\mathrm{d}}{\mathrm{d}\theta^i}u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big) = \frac{\partial u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)}{\partial\pi^i(s_t^j)}\cdot\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^i(s_t^j) + \frac{\partial u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)}{\partial\pi^j(s_t^i)}\cdot\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^j(s_t^i).$$

Both terms are retained: the first captures the direct marginal effect of changing π^i on i's own payoff, and the second captures the indirect effect routed through the opponent's state-conditioned response, which is non-zero because the opponent's input s_t^i is the reputation state of i and therefore itself depends on π^i via the signals emitted in prior interactions. Substituting back yields the final continuous deterministic gradient:

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}\theta^i}\mathbb{E}_\xi\left[\sum_{t=1}^{T} r_t^i\right] = \frac{1}{N(N-1)}\mathbb{E}_\xi\Bigg[\sum_{t=1}^{T}\sum_{j\neq i}\Bigg(&\frac{\partial u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)}{\partial\pi^i(s_t^j)}\cdot\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^i(s_t^j)\\ &+\frac{\partial u^i\big(\pi^i(s_t^j),\pi^j(s_t^i)\big)}{\partial\pi^j(s_t^i)}\cdot\frac{\mathrm{d}}{\mathrm{d}\theta^i}\pi^j(s_t^i)\Bigg)\Bigg].\end{aligned}$$

Interpretation. The gradient decomposes into two additive terms with clear physical meaning, both arising from the chain rule on a continuously differentiable payoff u^i and both surviving for any continuous reward structure.

(1) *Direct marginal-payoff term*, $\frac{\partial u^i(\pi^i(s_t^j),\pi^j(s_t^i))}{\partial\pi^i(s_t^j)}\cdot\frac{\mathrm{d}\pi^i(s_t^j)}{\mathrm{d}\theta^i}$. This is the own-input partial of the payoff, contracted with agent i's own policy derivative. It captures the direct effect of changing π^i on i's payoff at a single interaction, holding the opponent's action fixed.

(2) *Reputation-chained opponent-response term*, $\frac{\partial u^i(\pi^i(s_t^j),\pi^j(s_t^i))}{\partial\pi^j(s_t^i)}\cdot\frac{\mathrm{d}\pi^j(s_t^i)}{\mathrm{d}\theta^i}$. This is the opponent-input partial of the payoff, contracted with the total derivative of the opponent's action with respect to π^i. Even though π^j itself is fixed as a function (we do not model its learning), the opponent's input s_t^i is the reputation state of i, which depends on π^i through the signals emitted in prior interactions, so $\mathrm{d}\pi^j(s_t^i)/\mathrm{d}\pi^i \neq \mathbf{0}$. This is precisely the gradient-highway channel of Section 4.2 in continuous form: the reputation-mediated response of the opponent is what carries the analytic signal that justifies cooperation against the direct marginal-payoff penalty.

Theoretical positioning. Anchoring the derivation at the continuously differentiable payoff u^i with a deterministic continuous policy gives a prior-free mathematical basis for reputation-mediated cooperation beyond binary discrete formulations. The continuous leading-eight relaxations used as test-bed opponents in our experiments, and more generally any non-linear continuous reputation update, fit this setting without modification. The first term above is the standard DPG-style own-payoff gradient; the second term is the reciprocity-gradient contribution, made explicit here as the chain-rule product of the opponent's state-sensitivity and the reputation's sensitivity to i's own past actions. Together they confirm that the reciprocity gradient is not a heuristic enhancement but the

faithful total derivative of the cumulative payoff with respect to a deterministic continuous policy whenever reputation couples an agent's past actions to its future opponents' inputs.

F Opponent-Modeling Engineering Details

This appendix collects the engineering choices for the differentiable opponent-modeling procedure used in Section 4.3. These are the choices that make the algorithm work in practice; none of them are core contributions of the paper.

F.1 Full Algorithm

Algorithm 1 gives the per-opponent training loop, and Algorithm 2 specifies every interaction in a real or virtual rollout. Virtual cumulative returns are denoted by $\hat{R}^i$. The learner index i is fixed throughout training. At interaction t, $d = \mathsf{D}_t$ is the donor and $r = \mathsf{R}_t$ is the recipient; either, or neither, may be i. The donor produces the action $a_t^{d\to r}$, and the recipient subsequently produces the assessment $\sigma_t^{d\leftarrow r}$, which is appended to the *donor's* public reputation history.

Policy inputs and sampling. Write h_t^ℓ for agent ℓ's public signal history, including its initial reputation token, and $s_t^\ell = f(h_t^\ell)$ for its aggregated reputation. To make the recorded inputs explicit, the rollout uses the interfaces

$$x_t^\pi = (h_t^r, h_t^d), \qquad x_t^\varphi = (a_t^{d\to r}, h_t^r, h_t^d).$$

These tuples specify the available public inputs; each configured policy or surrogate selects the components it uses. The learner's action encoder reads the recipient history h_t^r. Its signal policy reads the donor action and, when configured, the recipient's own history h_t^r. A first-order assessment uses only the action; a second-order rule such as L6 additionally uses s_t^r; a third-order rule can also use s_t^d. The MLP surrogates use the last entries of the relevant histories, while Transformer surrogates encode the histories. Thus, $\pi^\ell(x_t^\pi)$ and $\varphi^\ell(x_t^\varphi)$ below include this input selection, aggregation, or encoding, rather than requiring every policy to use every component.

The continuous action and signal are network or fixed-rule outputs in $[0, 1]$, not Bernoulli samples. Each rollout call draws fresh initial reputation tokens from the configured initialization law ρ_0 and a fresh matching sequence from ξ. The batched implementation shares this sequence across B trajectories, with separately drawn initial tokens. Optional exploration replaces the learner's action with $U[0, 1]$ during pretraining, or adds clipped Gaussian action noise during subsequent real-data collection. It does not randomize opponent rules or replace the learner's signal rule. Real-data collection uses evaluation mode and no gradient graph; virtual policy updates use training mode, including any configured network dropout.

Implementation details. Each buffer record contains the B observations at one interaction, and an estimator step samples one such record and averages its squared prediction error across the batch. Actions are recorded whenever that opponent is the donor, and signals whenever it is the recipient, including interactions between two opponents. The two targets need not come from the same interaction. In virtual mode, freezing a surrogate's weights does not detach its output or its inputs: the learner's gradient still passes through the surrogate and through subsequent reputation updates. Unused or nonfinite parameter gradients are skipped. The balanced matching regimes give $n_i > 0$; if no learner interaction occurs under another matching law, the implementation returns the unnormalized zero return. Set F only after pretraining: it retains the fitted models throughout training, as in action-only L6; otherwise they are refitted at each outer iteration. The implementation optimizes mean payoff per own interaction with Adam; reported performance is evaluated in the real environment. Appendix D illustrates differentiation through the resulting interaction graph.

Algorithm 1: Reciprocity-gradient training with differentiable opponent models

Input: learner i; fixed opponents; batch size B; matching law ξ; initial-reputation law ρ_0; benefit b, cost c.
Input: outer iterations T_{outer}; real rollout calls N_{play}; estimator steps K_{est}; policy steps N_{train}; buffer window W.
Input: optional exploration: K_{explore} rollout calls and K_{pre} pretraining steps; N_{noise} noisy rollout calls with action-noise scale ν.
Input: surrogate learning rate α_{est}; learner rates $\alpha_\pi, \alpha_\varphi$; permanent-freeze flag F.

Initialize trainable learner networks; retain any prescribed fixed action or signal rule. Let Θ_i contain only the learner parameters trained in this setting.
Initialize $(\hat{\pi}_i^\ell, \hat{\varphi}_i^\ell)$ and separate action/signal buffers $(\mathcal{B}_i^{\ell,\pi}, \mathcal{B}_i^{\ell,\varphi})$ for each $\ell \neq i$.
Initialize the learner's Adam optimizer with the applicable parameter-group learning rates.
if $K_{explore} > 0$ **then**
    Open one exploration-buffer slot; collect K_{explore} real rollout batches using Algorithm 2, with uniform-random learner actions.
    Fit each surrogate with K_{pre} Adam steps on its corresponding buffered MSE loss, as in Phase 2.
    Clear the exploration buffers before collecting the online training data.
for $q = 1, \ldots, T_{outer}$ **do**
    `// Phase 1: collect observations from the real population`
    Open a new slot in each buffer; discard slots older than W outer iterations.
    Collect N_{play} real rollout batches using Algorithm 2, with the current learner and true opponents, without gradients.
    If enabled, collect N_{noise} additional real rollout batches with learner action $\text{clip}_{[0,1]}(\pi^i_{\theta^i}(x^\pi) + \epsilon)$, $\epsilon \sim \mathcal{N}(0, \nu^2)$.
    `// Phase 2: fit action and signal models separately`
    if *not* F **then**
        Enable surrogate parameter gradients.
        foreach *opponent* $\ell \neq i$ *with a nonempty action buffer* **do**
            Initialize an action-estimator Adam optimizer at rate α_{est}.
            for $k = 1, \ldots, K_{est}$ **do**
                Draw (x^π, a) from $\mathcal{B}_i^{\ell,\pi}$; take an Adam step minimizing $\text{mean}[(\hat{\pi}_i^\ell(x^\pi) - a)^2]$.
        foreach *opponent* $\ell \neq i$ *with a nonempty signal buffer* **do**
            Initialize a signal-estimator Adam optimizer at rate α_{est}.
            for $k = 1, \ldots, K_{est}$ **do**
                Draw (x^φ, σ) from $\mathcal{B}_i^{\ell,\varphi}$; take an Adam step minimizing $\text{mean}[(\hat{\varphi}_i^\ell(x^\varphi) - \sigma)^2]$.
    `// Phase 3: differentiate freshly simulated virtual rollouts`
    Freeze all surrogate weights, retaining their derivatives with respect to inputs.
    for $n = 1, \ldots, N_{train}$ **do**
        Draw a fresh virtual rollout batch with Algorithm 2: retain i's live policies and replace every opponent policy by its surrogate.
        Set $L = -\frac{1}{B}\sum_{m=1}^{B} \hat{R}_m^i / \max(1, n_i)$, where n_i counts i's donor and recipient appearances in this rollout.
        Clear learner gradients; if L has a gradient graph, differentiate it with respect to Θ_i through the full virtual history.
        Clip the joint learner-gradient norm at 1 and take an Adam step minimizing L; leave fixed learner rules and surrogate weights unchanged.
Evaluate the final learner against the true opponents without exploration noise or gradients.
return *learner policies* $(\pi^i_{\theta^i}, \varphi^i_{\eta^i})$.

Algorithm 2: One batched rollout: donor action, recipient assessment, and history update

Input: real or virtual mode; i, B, ξ, ρ_0, b, c; learner policies; true opponents or fitted surrogates; real-data buffers; optional learner-action exploration.

For each agent ℓ, draw B initial tokens from ρ_0 and initialize h_0^ℓ with those tokens. Set $R^i \leftarrow \mathbf{0} \in \mathbb{R}^B$ and $n_i \leftarrow 0$.

Draw one matching sequence $M \sim \xi$, shared across the batch.

foreach *interaction* t *with donor* d *and recipient* r *in* M **do**

  Form the pre-interaction action input $x_t^\pi = (h_t^r, h_t^d)$.

  `// Only the donor selects the donation`

  if $d = i$ **then**

    $a_t^{d\to r} \leftarrow \pi^i_{\theta^i}(x_t^\pi)$; apply the selected exploration rule only in real mode.

  else if *real mode* **then**

    Observe the true donor's action $a_t^{d\to r} \leftarrow \pi^d(x_t^\pi)$.

  else

    $a_t^{d\to r} \leftarrow \hat{\pi}_i^d(x_t^\pi)$.

  if $d = i$ **then**

    $R^i \leftarrow R^i - ca_t^{d\to r}$; $n_i \leftarrow n_i + 1$.

  if $r = i$ **then**

    $R^i \leftarrow R^i + ba_t^{d\to r}$; $n_i \leftarrow n_i + 1$.

  Form $x_t^\varphi = (a_t^{d\to r}, h_t^r, h_t^d)$, using histories before the new assessment is written.

  `// Only the recipient assesses the donor`

  if $r = i$ **then**

    $\sigma_t^{d\leftarrow r} \leftarrow \varphi^i_{\eta^i}(x_t^\varphi)$.

  else if *real mode* **then**

    Observe the true recipient's signal $\sigma_t^{d\leftarrow r} \leftarrow \varphi^r(x_t^\varphi)$.

  else

    $\sigma_t^{d\leftarrow r} \leftarrow \hat{\varphi}_i^r(x_t^\varphi)$.

  if *real mode* **then**

    If $d \neq i$, append a detached copy of $(x_t^\pi, a_t^{d\to r})$ to $\mathcal{B}_i^{d,\pi}$.

    If $r \neq i$, append a detached copy of $(x_t^\varphi, \sigma_t^{d\leftarrow r})$ to $\mathcal{B}_i^{r,\varphi}$.

  Append $\sigma_t^{d\leftarrow r}$ to h_t^d to form h_{t+1}^d; leave every other history unchanged. In virtual mode, retain this operation in the gradient graph.

return *the batch of cumulative learner returns* R^i *and participation count* n_i.

F.2 Design Variants

Algorithm 1 admits several design choices, summarized below; the experiments fix the defaults marked (D).

Source of modeling data. (D) *Sliding window*: a buffer of the most recent W outer iterations of observed transitions, used for opponent-model fitting; virtual rollouts draw fresh matching sequences and initial reputations. (a) *Same-batch*: current batch only; simpler, but high variance. (b) *Previous-batch*: cleanly separates modeling and optimization data, at the cost of one-iteration model lag.

Virtual-graph construction. (D) *Fresh simulation*: draw a new matching sequence and initial reputations for each virtual rollout, using the same laws as the real environment and substituting opponent models at non-learner nodes. (a) *Replay*: reuse a recorded policy-independent matching sequence, but recompute all actions, signals, and histories using the current learner and fitted models.

Update frequency. (D) Update opponent models every outer iteration. (a) Every $K > 1$ outer iterations when opponents change slowly. (b) Strictly alternating modeling and optimization phases; cleanest separation but lower data efficiency. (c) *Explore-then-freeze*: at $t = 0$, before policy optimization begins, Agent 0 runs K_{explore}=100 batched rollouts against the true opponents under a uniform-random action policy and records the publicly observable (a, s, σ) tuples each opponent

emits during natural gameplay; the estimators are fitted to this exploration buffer offline and then frozen for the remainder of training. This variant avoids the buffer-clustering drift identified in Setting (B.1) and is what we use to recover the L6 discriminative action rule. The protocol is fully observational: opponents are never queried at synthetic inputs, only observed through actual interactions, so no oracle access is invoked.

Estimator architecture. (D) A small differentiable MLP fitted by mean-squared error on publicly observable interaction data. Layer widths and remaining hyperparameters are listed in the per-experiment paragraphs of Appendix J.

Per-opponent vs. parameter-shared. (D) *Per-opponent*: one pair $(\hat{\pi}_i^j, \hat{\varphi}_i^j)$ per opponent, $O(N)$ parameters. (a) *Parameter-shared*: one pair $(\hat{\pi}(\cdot; z_j), \hat{\varphi}(\cdot; z_j))$ shared across opponents via per-opponent embeddings z_j, $O(1)$ parameters in N. Setting (D) reports a scalability comparison.

F.3 Model Accuracy and Gradient Bias

If $\hat{\pi}_i^j$ matches π^j exactly (and likewise for $\hat{\varphi}_i^j$), the gradient computed through the virtual graph equals the gradient computed through the true computation graph. With imperfect models, the virtual gradient differs from the true gradient by a bias term whose magnitude is bounded by the supremum of the model error along the unrolled trajectory, multiplied by the per-step decay factor 0.25 at each gradient-highway hop.

Empirically we observe a stable *virtual-reward optimism* on the configurations we test. Under the online-update protocol the per-interaction-payoff gap between surrogate and real environment is $+0.1$ to $+0.3$; under the explore-then-freeze protocol used in Setting (B.1) the gap rises to $\approx +1.0$, is present at outer iteration 0 before any policy update, and does not grow over training. The gap is not explained by per-point approximation error (uniform-grid residual has mean ≈ 0 and max magnitude 0.012); it is consistent with compounding of small on-policy residuals through the $T \approx 50$-interaction reputation dynamics, in line with model-based reinforcement learning, where imperfect world models still yield useful policy-improvement gradients (Sutton, 1991). The per-step decay bound caps the bias at a constant offset rather than letting it grow unboundedly; all external performance claims in the paper use the real-environment number, and we report surrogate values only as a training diagnostic.

F.4 Comparison to Prior Approaches to Oracle Access

The reciprocity-gradient expansion of Section E.2 requires backpropagating through other agents' action and gossip functions. A naive implementation gives the learner oracle access to every other agent's parameters, which is implausible when opponents are autonomous. Three families of prior work address this. *Centralized training, decentralized execution* (CTDE) accepts oracle access at training time: DIAL (Foerster et al., 2016) backpropagates through inter-agent message channels, MADDPG (Lowe et al., 2017) trains a centralized critic over joint observations and actions, and BPTA (Li et al., 2024) backpropagates through the joint action-execution order. *Opponent shaping* methods such as LOLA (Foerster et al., 2018), SOS (Letcher et al., 2019), and COLA (Willi et al., 2022) take a *second-order* gradient through one anticipated opponent learning step, explicitly requiring the partial $\partial\theta^j_{\text{new}}/\partial\theta^i$ of the opponent's update on the ego agent's parameters. For fixed, non-learning opponents, this correction is zero, leaving the chosen base best-response estimator. Appendix A explains why opponent shaping and the reciprocity gradient are conceptually orthogonal and can in principle be composed. Our method is, by contrast, *first-order*: we differentiate the ego agent's expected return through one autograd backward pass over the rolled-out interaction graph against fixed opponents, a structurally different and strictly weaker requirement. The single-hop sender-receiver *signaling gradient* of Lin et al. (2023) is the closest first-order analogue; the reciprocity gradient extends the same structural idea to multi-hop reputation chains. *Opponent modeling* replaces access to others' parameters with a learned belief over their behavior (Albrecht & Stone, 2018); the learner differentiates through its own model rather than through the true opponent. We adopt this route in the observational-access regime: the learning agent builds private, differentiable estimators of each opponent and backpropagates the reciprocity gradient entirely through these self-owned models. MFOS (Lu et al., 2022) similarly removes the opponent-update access from LOLA's family via meta-learning over observed trajectories.

G Experiments Overview

Tables 2 and 3 summarize the role of every experiment in the paper. For each experiment we state the scientific question it addresses, the core setting (opponents, information access, trainable components), and the expected outcome that would make the experiment count as a success. The setting labels (A.1)–(A.3), (B.1)–(B.3), and (C)–(E) are stable identifiers used throughout the paper. They map to the topical structure of Section 5 as follows. (A.1)–(A.3) (Section J.2) test best response under oracle opponent access for action only, signal only, and joint training respectively. (B.1)–(B.3) (Section J.3) test the same three settings after replacing oracle access with learned opponent models. (C) (Section J.4) compares against model-free deterministic baselines. (D) (Section J.5) tests scalability via parameter sharing. (E) (Section J.6) tests the indirect-only matching regime. The autograd correctness demonstration referenced in Section 4.2 appears in Appendix D. Detailed hyperparameters, seed counts, and full numeric results are provided in the individual paragraphs of Section 5 and in Appendix I.

ID	Question	Setting	Expected outcome
Best response under oracle opponent access			
(A.1)	Can a single agent recover a discriminative *action* rule against leading-eight opponents?	Agent 0 trains π^0 against frozen L_3/L_6 opponents; φ^0 held at identity; multiple reputation inits.	π^0 becomes context-sensitive ($\mathrm{std}[\pi^0] \geq 0.2$) and reaches a high fraction of the full-cooperation benchmark.
(A.2)	Can a single agent recover a discriminative *signal* rule when the opponent mix forces signal discrimination?	Agent 0 trains φ^0 against frozen ProudCoop(L_3)+AllDefector(L_3); π^0 held at L_3; b=10.	φ^0 differentiates cooperator vs. defector donors; per-interaction payoff approaches 2.25.
(A.3)	Can the reciprocity gradient optimize action and signal *simultaneously* under oracle access?	Agent 0 trains both π^0 and φ^0 from scratch; the true opponents; no warmup, no phasing, no regularizer.	Simultaneous convergence to "proud cooperator"-like discriminative strategy; per-interaction payoff $\approx 99\%$ of reference.
Best response under differentiable opponent models			
(B.1)	Does removing oracle access preserve action-rule recovery against L_6?	Agent 0 trains π^0 using differentiable surrogates $\hat{\pi}^{-0}, \hat{\varphi}^{-0}$ fit from public observations.	Discriminative π^0 still recovered; per-interaction payoff within a small gap of the oracle-access number.
(B.2)	Same question for *signal* optimization under learned opponent models.	Agent 0 trains φ^0 only; surrogate opponent models; ProudCoop+AllD setting.	Discriminative φ^0 recovered with per-interaction payoff $\geq 75\%$ of benchmark under learned models.
(B.3)	Can simultaneous action-signal optimization succeed under learned opponent models?	Agent 0 trains π^0 and φ^0 simultaneously; surrogate models; joint setting HybridCoop+AllD (both networks must be discriminative).	High seed rate with $\mathrm{std}[\pi^0], \mathrm{std}[\varphi^0] \geq 0.2$ simultaneously; per-interaction payoff well above any single-network strategy.

Table 2: **Experiments overview, part 1: method validation, groups (A)–(B) of Section 5.** Group (A) (Section J.2) tests best-response optimization under oracle opponent access, for the action, signal, and both-network settings. Group (B) (Section J.3) tests the same three settings after replacing oracle access with differentiable opponent models fitted from public observations.

H Action-Only and Signal-Only Pathway Ablations

This appendix complements Figure 3 (which shows only the joint training setting) with the two single-pathway ablations: action-only training against L6, and signal-only training against ProudCoop+AllDefector.

ID	Question	Setting	Expected outcome
Baseline comparison			
(C)	Do model-free deterministic baselines (DPG, DDPG, TD3) match the reciprocity gradient?	Same nine settings as (B.1)–(B.3); baselines share the architecture, observation, and budget of the RG learner.	Baselines collapse to constant-output policies on every setting that requires discrimination; RG retains a large per-interaction-payoff margin.
Scalability sweep			
(D)	Does the method retain its per-interaction payoff when (B)'s three successful settings are scaled to larger populations?	Re-run (B)'s action vs L6 (§J.3), signal vs ProudCoop+AllDefector pool (§J.3, c=5), and joint vs HybridCoop+AllDefector pool (§J.3) at $N \in \{5, 10, 15, 20, 25, 30\}$; pool fractions $p \in \{1/5, 3/10, 1/2, 7/10, 4/5\}$ for the two pool-based settings; parameter-shared estimator with $O(1)$ surrogate parameters in N.	Per-interaction payoff is preserved across N at every fixed p, and rises monotonically with p as more cooperators lift the full-cooperation reference.
Indirect-only matching			
(E)	Do (B)'s three successful settings still recover their discriminative best response when direct reciprocity is disabled?	Re-run (B)'s three settings under the (R2) indirect-only matching regime (each ordered pair meets exactly once per episode); N=3 with the (B) opponent populations; N_{play} scaled to keep total transitions per outer iteration comparable across regimes.	Per-interaction payoff and discrimination rates approximately match the (R1) numbers, confirming cooperation is mediated by reputation rather than direct reciprocity.

Table 3: **Experiments overview, part 2: comparison and scope, groups (C)–(E) of Section 5.** Group (C) (Section J.4) compares the reciprocity gradient against three deterministic model-free baselines. Group (D) (Section J.5) tests scalability across population sizes. Group (E) (Section J.6) tests robustness to indirect-only matching.

I Detailed Best-Response Results vs Hard-Coded Opponents

This appendix reports the full sweep of best-response experiments against hard-coded opponents whose results are flat-cooperation, partial, or otherwise non-discriminative. The discriminative settings are summarized in the main text (Setting (A.1) for the L6 action result and Setting (A.2) for the L3 ProudCoop + AllD signal result).

I.1 (A.1.i) Action Optimization vs Identity-Mapping Opponents

Agent 0's action network is trainable; signal is hard-coded as honest identity ($\sigma = a$). Opponents are hard-coded identity agents. The aim is to verify that the reciprocity gradient produces useful policy updates under the simplest possible reputation dynamics. We sweep two reputation aggregators (mean vs. EMA with decay 0.5) and three reputation initializations (const 0.5, all-uniform $U[0, 1]$, and a warmup from the stationary distribution).

Aggregator	const05	all_uniform	warmup_init
mean	3.997	4.020	4.027
EMA	4.273	4.291	4.292

Table 4: Setting (A.1.i): final per-interaction payoff against identity opponents (reference $= 4.5$). Both the reputation aggregator (rows: mean vs. EMA) and the reputation initialization (columns) are environment specifications, not learner-side hyperparameters; values across cells are therefore not directly comparable. Each entry reports the converged payoff under its own environment instance. The learned action policy is *flat* at $\pi \approx 1$ in every variant, matching the theoretically predicted best response under each environment: with honest identity signals, reputation rises unboundedly as the agent cooperates, and the cost $c = 1$ is always repaid.

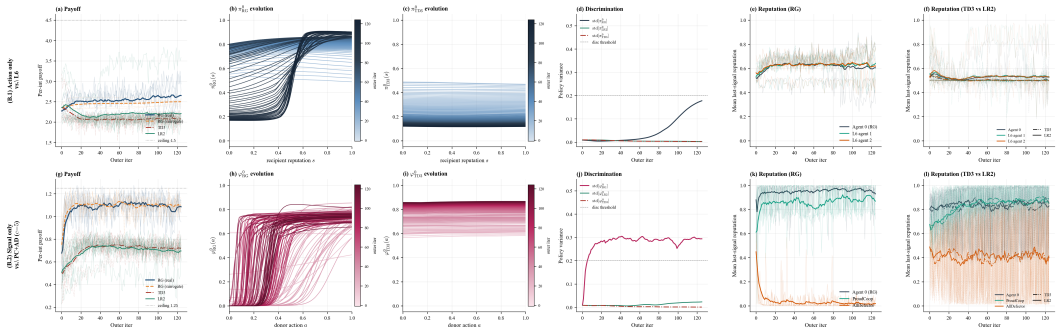


Figure 8: Single-pathway ablations (5 seeds, T_{outer}=125). *Top row.* action-only training against L6, reference $(b-c)/2 = 4.5$. *Bottom row.* signal-only training against ProudCoop+AllDefector with c=5, reference $(b-c)/4 = 1.25$. Columns: per-interaction payoff; RG policy evolution; TD3 policy evolution; profile std for RG, TD3, and LR2 (descriptive only, not a metric); reputation under RG; reputation under TD3 and LR2. Faint curves show individual seeds; bold curves show the bias-corrected EMA of the across-seed mean. RG develops a non-flat learned policy in both ablations; TD3 and LR2 stay constant.

Memory-window ablation. We also vary the opponent's reputation memory window (EMA, warmup_init): $k = 1$ gives 4.360, $k = 3$ gives 4.285, $k = \infty$ gives 4.264. Shorter memory produces faster reputation feedback and therefore a higher converged payoff; longer memory dilutes each cooperative action across more prior observations.

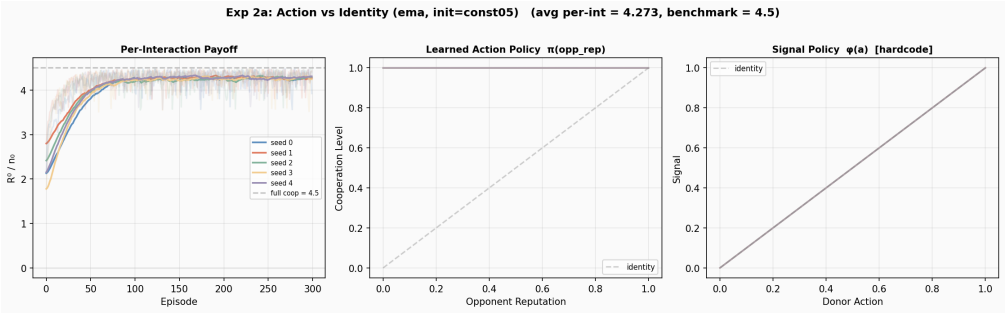


Figure 9: Setting (A.1.i) (EMA, const05). Left: per-interaction payoff climbs from ~ 0.5 to the reference. Right: learned action policy π^0 is flat at ≈ 1 across the reputation grid, confirming unconditional cooperation as the best response to identity opponents.

I.2 (A.2.i) Signal Optimization vs Identity-Mapping Opponents

Agent 0's *signal* network is trainable; the action is hard-coded as the mean-reputation identity. Against identity opponents the predicted best signal is reputation *inflation* ($\varphi^0(a) \to 1$): inflating others' reputation amplifies their cooperative responses to Agent 0 through the first-order feedback loop.

Aggregator	const05	all_uniform	warmup_init
mean	3.002	3.045	3.037
EMA	3.742	3.814	3.810

Table 5: Setting (A.2.i): final per-interaction payoff with signal training only (reference $= 4.5$). Aggregator (rows) and reputation initialization (columns) are environment specifications, not learner hyperparameters; values across cells are not directly comparable. Learned signal is flat at $\varphi^0 \to 1$ in every variant, matching the predicted inflation equilibrium under each environment.

I.3 (A.2.ii) Signal Optimization vs Leading-Eight Opponents

Against structured leading-eight opponents, signal-only training is *easier* than against identity opponents because the opponents' second-order assessment provides a cleaner gradient. Agent 0 trains φ^0 only; action is hard-coded.

	L3		L6	
Init Mode	mean	EMA	mean	EMA
const05	3.79	4.15	3.19	3.62
all_uniform	3.84	4.19	3.58	4.08
warmup_init	3.96	4.23	3.92	4.19

Table 6: Setting (A.2.ii): final per-interaction payoff with signal training against leading-eight opponents (reference $= 4.5$). Each (opponent, init, aggregator) cell is a separate environment variant; values across cells are not directly comparable. Learned signal converges to a near-identity mapping in every variant: when opponents already implement a stable second-order assessment, the best response under each environment is to match it rather than try to inflate.

J Detailed Experimental Results

This section reports the empirical evaluation of the reciprocity gradient. We address five experimental questions: **(i)** under oracle access to opponents' true functions, how close does the analytic gradient land to the full-cooperation reference in each setting (Section J.2); **(ii)** when opponents are replaced with private differentiable surrogates fitted from public observations, how far does the observational-access variant remain from the full-cooperation reference, and what fraction of the oracle performance does it recover (Section J.3); **(iii)** on the same settings, how does the observational-access variant of the reciprocity gradient compare with deterministic policy-gradient baselines (DPG, DDPG, TD3) that are model-free by construction (Section J.4); **(iv)** does the method retain its per-interaction payoff at larger population sizes (Section J.5); and **(v)** does it retain its per-interaction payoff under the indirect-only matching regime that disables direct reciprocity (Section J.6).

J.1 Overview and Metrics

The experiments are organized into five groups (Table 7). **(A)** (Section J.2) trains action, signal, and both policies against frozen opponents under oracle access, isolating the optimization step from any opponent-modeling error and establishing the oracle-access references that the rest of the paper targets. **(B)** (Section J.3) repeats the three settings after replacing oracle access with private differentiable surrogates fitted from public observations. **(C)** (Section J.4) benchmarks the analytic-gradient path against three deterministic model-free actor-critic methods on the same nine settings. **(D)** (Section J.5) tests scalability by re-running all three (B) settings at population sizes $N \in \{5, 10, 15, 20, 25, 30\}$ (the $N = 3$ anchor is (B) itself). **(E)** (Section J.6) tests robustness to the indirect-only matching regime by re-running the same three settings without direct reciprocity.

	Trained	Opponent action	Opponent signal	Access
A	π^0	L6	L6	Oracle
	φ^0	ProudCoop, AllDefector	L3	Oracle
	π^0, φ^0	HybridCoop, AllDefector	L3	Oracle
B	(same three settings as A, correspondingly trained)			Observational
C	π^0, φ^0	9 (Trained $\times$ Opponent) settings; 3 baselines		N/A
D	(same three settings as B, varying population size $N \in \{5, 10, 15, 20, 25, 30\}$)			
E	(same three settings as B, indirect-only matching regime)			
F	all π^i, φ^i	(homogeneous $N = 3$ self-play, no fixed opponents)		

Table 7: Experiment matrix. Opponent labels: L3 (*Simple Standing*, forgiving) and L6 (*Stern Judging*, strict) are second-order leading-eight rules (Ohtsuki & Iwasa, 2004, 2006). Each fully specifies both an action policy (cooperate iff the recipient is in good standing under its own assessment) and a signal policy (the assessment rule itself). Discrete tables are provided in Appendix B and the continuous formulation in Appendix N. ProudCoop, HybridCoop, and AllDefector are action policies designed for the experiments and paired with the L3 signal rule. The functional definitions and objectives of experimental groups A through F are detailed in the preceding main text.

Evaluation Metrics. Our primary evaluation metric is the *per-interaction payoff* of the learning agent, averaged across seeds and reported in absolute units and as a percentage of an analytic per-setting reference determined by b, c, and the fixed opponents' policies. Per-setting references are stated with each result. We use the per-interaction unit rather than per-episode because episode length varies with the matching schedule under regime R1 (Section 3), so per-episode returns are high-variance and not directly comparable across settings.

We also report the sample standard deviation of each learned policy's output, evaluated at convergence on a uniform grid spanning its input domain. We use this profile std as a *qualitative descriptor* of policy shape rather than as a pass/fail metric: a near-zero value identifies constant-output attractors, while a positive value identifies non-constant policies. The descriptor cannot rank two non-constant policies (a step function and a noisy small-amplitude oscillation can have similar std, but only the first is the optimum), so we do not use it to compare methods. It is useful only to separate collapse modes from non-collapse modes when discussing what kind of attractor a method ends up in. Reported $\pm$ values are standard deviations across seeds (not confidence intervals); seed counts and reputation-initialization modes are declared per setting. The continuous formulation of the leading-eight norms used as opponents throughout the paper is derived in Appendix N; Appendix G provides a tabular restatement of the experiments. Reputation-initialization mode is varied per setting throughout the experiments below (declared per result), and a discrete-action extension of the framework is reported in Appendix L.

Compute budget. All experiments were run on a single Apple M4 Pro CPU (no GPU required). The entire experimental pipeline — including all main-text experiments, the autograd demonstration of Appendix D, all seeds, all configurations, and all ablations — completes in approximately 8 hours of wall-clock time. Individual experiment runtimes range from seconds (autograd demo, 3 agents $\times$ 3 timesteps) to $\sim$ 90 minutes ((E) indirect-matching ablation, 60 configurations $\times$ 20 outer iterations $\times$ 5 seeds). The method's modest compute footprint reflects the small population sizes ($N \leq 15$) and the efficiency of analytic gradient computation through the surrogate graph relative to Monte-Carlo policy-gradient estimators.

Policy-variance grid. The policy-variance metric is computed as follows. The learner's action policy π^0 is evaluated at 21 evenly-spaced recipient-reputation values $s \in \{0, 0.05, 0.10, \ldots, 1\}$, each query feeding a single-step reputation history of value s; $\mathrm{std}[\pi^0]$ is the sample standard deviation of the resulting 21-point cooperation-rate profile. The learner's signal policy φ^0 is evaluated analogously on a uniform 21-point grid of donor actions over $[0, 1]$; for second-order signal policies that additionally consume the learner's own reputation, the auxiliary reputation input is held fixed at 0.5 during evaluation. The grid resolution ($n_{\text{grid}} = 21$) is chosen to be fine enough that the profile standard deviation is insensitive to further refinement at the precision at which we report it.

Taxonomy of Opponent Regimes and Sanity Checks. To rigorously evaluate the reciprocity gradient, we conduct extensive experiments across a wide spectrum of opponent populations. We categorize each setting into one of two regimes based on the topological complexity of its analytical optimum.

In *degenerate regimes* — populations dominated by unconditional cooperators, by unconditional defectors, or by leading-eight opponents whose stationary reputation distribution renders the optimal response state-independent — the analytical optimum collapses to a constant-output policy: unconditional cooperation, unconditional defection, or a constant signal output. Across dozens of degenerate configurations against identity, L3, L6, and exotic mixed populations under multiple reputation aggregators and initializations (Appendix I), the reciprocity gradient consistently attains the oracle-access reference, confirming that the underlying state–reward dynamics, matching schedule, and reputation aggregator are unbiased and that the analytical optimum is reachable from random initialization. These configurations serve as a critical sanity check on the environment specification.

The core challenge of indirect reciprocity, however, lies in *discriminative regimes* that demand state-dependent strategies. We construct three such regimes: action training against the second-order leading-eight rule L6 (*Stern Judging*), signal training against an asymmetric ProudCoop+AllD pool, and joint training of both networks against HybridCoop+AllD. The deterministic policy-gradient baselines (Section J.4) collapse to constant-output policies on every discriminative regime, while the reciprocity gradient recovers the conditional structure and reaches 72–99% of the oracle-access reference. The cooperator-fraction sweep in Section J.5 interpolates between the two regimes along $p \in \{0.2, 0.3, 0.5, 0.7, 0.8\}$, with $p \to 0$ and $p \to 1$ approaching degenerate settings and $p = 0.5$ being discriminative. Every quantitative gap between the reciprocity gradient and the deterministic baselines reported in the main text is measured on a discriminative regime; Table 8 summarises the regime-level outcome and per-setting numbers are tabulated in Appendix I.

Regime (representative opponents)	Required best response	RG (% of reference)	Model-free baselines
Degenerate			
Identity opponents (action)	constant $\pi^0 = 1$	89–95%	20–44%
Identity opponents (signal)	constant $\varphi^0 \to 1$	67–85%	53–60%
L3 opponents (action)	constant $\pi^0 = 1$	89–95%	33–49%
L3 opponents (signal)	constant φ^0 matches the norm	84–94%	92–93%
L6 opponents (signal)	constant φ^0 matches the norm	71–93%	79–92%
ProudCoop+AllD (signal, flat-coop attractor)	constant cooperate at $b = 10$	95%	79–92%
Discriminative			
L6 opponents (action)	state-dependent $\pi^0(s)$	72–93% ($\geq 5/5$ disc)	30–49% (collapse to flat)
ProudCoop+AllD (signal, c=5)	state-dependent $\varphi^0(a)$	96% (5/5 disc)	53–61% (collapse to flat)
HybridCoop+AllD (joint)	state-dependent π^0, φ^0	99% (20/20 disc)	69–80% (collapse to flat)

Table 8: Aggregated taxonomy of opponent regimes. *Degenerate* regimes admit a constant-output analytical optimum; *discriminative* regimes require a state-dependent analytical optimum. The reciprocity gradient (RG) reaches near-full-cooperation-reference on every regime, while the deterministic policy-gradient baselines DPG/DDPG/TD3 (Section J.4) collapse to constant-output policies wherever the optimum is state-dependent. Degenerate settings serve as sanity checks: their oracle-access reference is reachable by any constant-output policy and the environment is therefore unbiased. The 19–30 pp margin between RG and the baselines on the joint discriminative regime, the 35–43 pp margin on the signal vs. ProudCoop+AllD (c=5) regime, and the absence of any disc seed in the baseline ensemble together quantify the value of the analytic gradient. Per-setting numbers and learned-policy shapes are listed in Tables 4–6.

J.2 Best Response under Oracle Opponent Access

We first establish the references the rest of the paper targets. Under oracle access to opponents' true action and signal functions, the reciprocity gradient is the exact analytic gradient through the multi-agent rollout; results are therefore free of opponent-modeling error and the question reduces to whether the optimization step alone recovers the predicted discriminative policies. We report

three settings: action only (signal hard-coded), signal only (action hard-coded), and simultaneous optimization of both networks from scratch.

Action optimization against L6. Against the second-order leading-eight rule L6 — *Stern Judging*, the strictest leading-eight norm (Ohtsuki & Iwasa, 2004, 2006); full assessment table and contrast with L3 (*Simple Standing*) given in Appendix B — the reciprocity gradient recovers a discriminative action policy under oracle opponent access. The learned π^0 separates the reputation sub-region in which cooperation pays from the sub-region in which it does not, settling at the rectangular decision boundary that the L6 second-order assessment carves through reputation space (Figure 10). Across 5 seeds, the per-interaction payoff reaches 3.25 against the 4.5 reference (72%); every seed converges to a non-constant π^0 rather than to a constant-output attractor. The discriminative setting is robust across the two reputation aggregators we test (`mean` and EMA); we sweep reputation initialization and hard-coded signal as additional axes, and report the remaining configurations — including all L3 settings and the non-discriminative L6 settings — in Appendix I, where they collapse to flat-cooperation or saddle attractors.

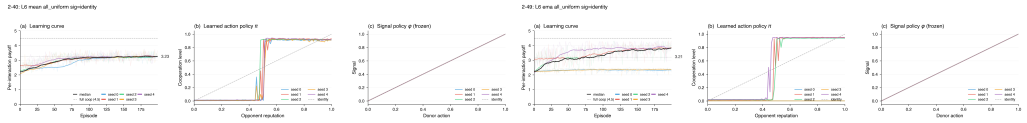


Figure 10: Action-network recovery against L6 opponents with `all_uniform` initialization and identity signal. *Left:* `mean` aggregator. *Right:* EMA aggregator. The learned policy $\pi^0(s)$ drops from near 1 at high recipient reputation to near 0 at low recipient reputation, separating the cooperate-pays sub-region from the punish-pays one. Final per-interaction payoff is 3.25 (`mean`) and 3.24 (EMA) against the 4.5 reference.

Signal optimization against ProudCoop+AllD (L3). Against an asymmetric opponent pair — a reputation-conditioned cooperator and an unconditional defector, both emitting gossip under L3 (Simple Standing) — the reciprocity gradient recovers a discriminative signal rule (Table 9). Across 10 seeds per initialization, every seed learns a discriminative signal policy: under `const05` initialization the per-interaction payoff reaches 94% of the 2.25 reference (10/10 disc), and under `all_uniform` it reaches 89% (10/10 disc). Three structural choices drive discrimination on this setting. First, L3 is forgiving: it assigns a good signal when the learner justifiably defects against an all-defector, preserving the gradient channel that drives discrimination. A strict second-order norm such as L6 punishes the same move and destroys the gradient. Second, AllDefector is a constant opponent rather than a reputation-conditioned one, so the signal network faces a clean two-class problem (cooperator vs. defector) rather than a continuous asymmetry. Third, the proud-cooperator's behavior conditions on its *own* reputation, so the reciprocity gradient flows through the learner's signal and back into the proud-cooperator's next action. This self-referential gradient channel lifts the learner from the flat-inflation policy (per-interaction payoff ~ 0.4) into the discriminative regime (≈ 2.1).

Initialization	n	Per-int (mean)	% bench	std$[\varphi^0]$ above 0.20	Notes
`const05`	10	$\mathbf{2.112 \pm 0.024}$	**94%**	**10/10**	most robust setting
`all_uniform`	10	2.008 ± 0.025	89%	10/10	—

Table 9: Signal-network recovery against ProudCoop(L3)+AllDefector(L3); benchmark $(b-c)/4 = 2.25$ at $b = 10, c = 1$. The learned signal policy is reported on a uniform donor-action grid.

Both networks and the asymmetric-LR protocol. We next train π^0 and φ^0 simultaneously from random initialization, with both networks updated against the same scalar reward. Classical opponents do not constrain both networks simultaneously: ProudCoop conditions only on its own reputation (so φ^0 alone suffices), while L3/L6 condition only on the recipient's reputation (so π^0 alone suffices). We therefore design **HybridCoop(L3)**, whose cooperation rate is $\sigma\big(10 \cdot (0.5\, r_{\text{own}} + 0.5\, r_{\text{recipient}} - 0.5)\big)$. The dual conditioning forces both networks to be discriminative at the optimum: π^0 must respond to the recipient's reputation, and φ^0 must shape HybridCoop's reputation through its signals. Two frozen opponents (HybridCoop(L3) and AllDefector(L3)); $b = 10$, $c = 1$, $N = 3$, batch 128, $\delta = 0.98$;

uniform reputation initialization; both networks trainable from random initialization, no warmup, no phasing, no regularization.

The simultaneous-training protocol couples two knobs that follow directly from the entry-channel analysis of Section 4.2 (Equation 4), which predicts the φ^0-gradient is structurally weaker than the π^0-gradient by roughly an order of magnitude:

1. *Asymmetric learning rates*, with π^0-LR $= 3 \times 10^{-5}$ and φ^0-LR $= 3 \times 10^{-3}$. The $100\times$ ratio equalises effective update magnitudes given the $\sim 10\times$ gradient-magnitude imbalance at initialization.
2. *Budget matching*, with total epochs scaled as $1/\mathrm{LR}$ so that LR $\times$ epochs remains constant. This gives Adam's second-moment estimate enough samples to resolve the weak initial φ gradient.

The protocol reaches $\mathbf{2.24 \pm 0.09}$ per interaction ($\mathbf{99}\%$ of the 2.25 benchmark); on $\mathbf{4/5}$ seeds the learned π^0 takes a step shape and φ^0 an S-curve shape (Figure 11). This protocol defines the simultaneous-training recipe used in Setting (B.3).

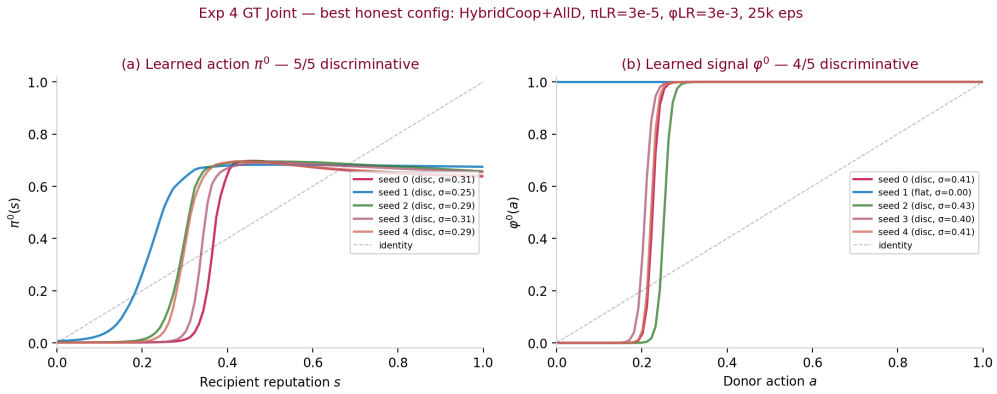


Figure 11: Both-network discriminative recovery under oracle opponent access. Asymmetric-LR protocol, 5 seeds, HybridCoop+AllD, 25,000 epochs, π^0-LR $= 3 \times 10^{-5}$, φ^0-LR $= 3 \times 10^{-3}$. *Left:* learned action policy $\pi^0(s)$; $5/5$ seeds recover the step-function shape. *Right:* learned signal policy $\varphi^0(a)$; $4/5$ seeds recover the S-curve. Simultaneous discrimination of both networks is the oracle-access reference that Setting (B.3) ports to learned opponent models.

J.3 Best Response under Learned Opponent Models

We now drop the oracle assumption: each opponent is replaced by a private differentiable surrogate fitted from public observations (Section 4.3, Algorithm 1). The three settings of Section J.2 are repeated under this stricter setting. The headline is that the both-network setting recovers the oracle-access reference without loss; the surrogate-fitting protocol uses uniform-random exploration episodes followed by fitted-and-frozen surrogates within each inner policy-update loop, a standard data-coverage protocol for on-policy model-based methods.

Action optimization against L6. π^0 is trainable; $\varphi^0(a) = a$ is hard-coded. Two frozen tanh-continuous L6 opponents ($\beta = 5$) occupy the remaining seats; reputations are initialized uniformly over $[0, 1]$. Agent 0 maintains two MLP estimators per opponent (Appendix F.2) and optimizes through a virtual computation graph in which the surrogates replace the true opponent functions. Aggregator: `mean`.

On-policy data collection concentrates near the L6 cooperative attractor and leaves the off-attractor region under-covered, a standard model-based-RL coverage issue (Section J.4's deterministic baselines hit the same coverage limit on this cell). We address it with the standard model-based remedy:

1. *Exploration-based pretraining.* At outer iteration $t = 0$, before policy optimization begins, Agent 0 plays $K_{\text{explore}} = 100$ episodes against the true opponents under a uniform-random action policy $\pi^0(s) \sim U[0, 1]$, recording the publicly observable (a, s, σ) tuples each opponent emits during natural gameplay. The estimators $(\hat{\pi}^j, \hat{\varphi}^j)$ are fitted to this exploration buffer for

$K_{\text{pre}} = 800$ MSE steps. The protocol uses no oracle access: opponents are only ever observed through actual interactions, never queried at synthetic inputs. Terminal MSE on a held-out 32×32 uniform-grid evaluation is $\leq 10^{-4}$.

2. *Estimator freezing.* After pretraining, the surrogate parameters are fixed; the online estimator-update step is skipped throughout policy training. This prevents buffer-clustering drift from re-saddling the estimator.

We additionally inject Gaussian action noise $\sigma_{\text{explore}} = 0.5$ on Agent 0 during data collection and extend training to $T_{\text{outer}} = 400$ outer iterations under a slowed learning rate. With these changes, $7/10$ seeds recover the discriminative action rule at $\text{std}[\pi^0] = 0.43 \pm 0.03$ and per-interaction payoff 3.06 ± 0.20 among the disc seeds, reaching 94% of the oracle-access reference (Table 10, Figure 12); the three non-disc seeds collapse to flat attractors with per-int ~ 2.22.

Configuration	n	Disc seeds	$\text{std}[\pi^0]$	Per-int (mean $\pm$ std)
Oracle access (Setting (A.1))	5	$5/5$	~ 0.34	3.245
Naive online surrogate (no fix)	3	$1/3$	0.11 ± 0.16	2.56 ± 0.32
+ exploration noise $\sigma_{\text{explore}} = 0.2$ only	3	$0/3$	0.00 ± 0.00	2.71 ± 0.20
Explore + freeze + $\sigma_{\text{explore}} = 0.5$, $T_{\text{outer}} = 400$	**10**	$\mathbf{7/10}$	$\mathbf{0.30 \pm 0.20}$	$\mathbf{2.81 \pm 0.42}$
conditioned on disc seeds	7	—	$\mathbf{0.43 \pm 0.03}$	$\mathbf{3.06 \pm 0.16}$

Table 10: Action-network recovery under learned opponent models (L6 $\times$ `all_uniform` $\times$ identity-signal setting). Random-policy exploration followed by frozen surrogates recovers $7/10$ seeds at 94% of the oracle per-interaction target (3.245), compared to $1/3$ for the naive online pipeline and $0/3$ for exploration-noise-only. The protocol uses no oracle access: opponents are observed only through actual interactions during the K_{explore}=100 pre-training episodes. Other aggregator/init settings reach $> 90\%$ of their respective references via the flat-cooperation basin without requiring the fix; see Appendix I.

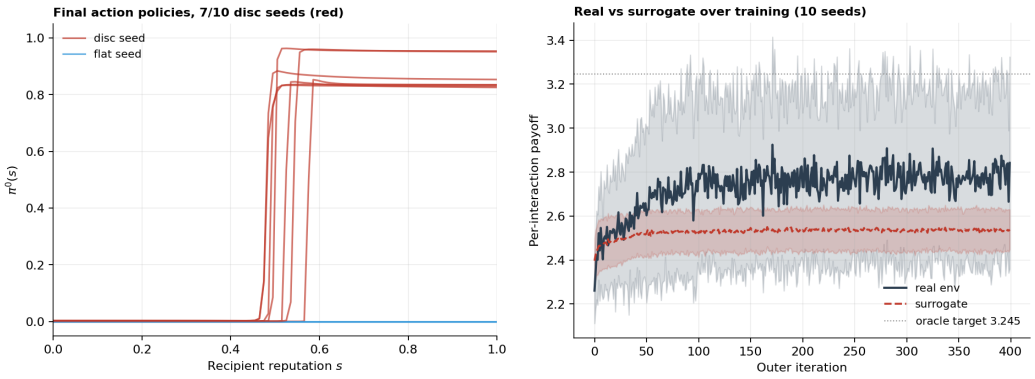


Figure 12: Action-network under learned opponent models, L6 $\times$ `all_uniform` setting with random-policy exploration pretraining followed by estimator freezing, 10 seeds. *Left:* final learned action policy $\pi^0(s)$ across seeds; red curves are the 7 disc seeds (rectangular decision boundary, $\text{std}[\pi^0] \in [0.41, 0.48]$, per-int 2.65–3.34); blue curves are the 3 flat-collapse seeds. *Right:* real-environment (dark) and surrogate-environment (dashed) per-interaction payoff over training, shaded band ± 1 std across seeds. The surrogate sits stably $\approx +1.0$ above the real at every iteration.

The surrogate-environment per-interaction payoff sits stably $\approx +1.0$ above the real-environment payoff at every iteration, including $t = 0$ before any policy update, and does not grow over training. The gap is not estimator drift: the frozen estimator's MSE on a held-out 32×32 uniform-grid evaluation stays at 10^{-5} throughout. The uniform-grid residual has mean ≈ 0 and maximum magnitude 0.012, too small per-point to produce a $+1.0$ gap directly. The mechanism is consistent with the model-based-RL virtual-reward optimism pattern (Sutton, 1991): the policy's on-policy distribution concentrates on the discriminative decision boundary, the estimator's local residual on that sub-manifold is small but systematically positive, and small per-step over-estimations compound through the $T \approx 50$-interaction reputation dynamics into $O(1)$ cumulative bias. The bias is bounded

above by the sigmoid-derivative chain (Section 4.2) and does not destabilize training; the learner converges to a policy whose real-environment evaluation reaches 94% of the oracle target. All external claims in this subsection use real-environment numbers.

Signal optimization. φ^0 is trainable; $\pi^0(r) = r$ is hard-coded. Opponents are the ProudCooperator + AllDefector pair from Setting (A.2). Donation cost is raised to $c = 5$ to separate the non-cooperative and cooperative regimes more cleanly; the per-interaction reference is $(b - c)/4 = 1.25$ since AllDefector never reciprocates. Sweep: two initializations (`const05`, `all_uniform`); 5 seeds each. Other hyperparameters match Setting (B.1).

The surrogate reaches per-interaction payoff 0.82–0.84 (66–68% of the 1.25 reference) under `all_uniform` initialization (Table 11, Figure 13). Discriminative signal learning ($\mathrm{std}[\varphi^0] > 0$) requires `all_uniform`; under `const05` the recovered signal is partial. The pattern matches the buffer-clustering analysis above: clustered initial conditions deny the estimator the off-saddle coverage needed to escape the flat-output basin reported in Setting (B.1).

Initialization	n	Per-int (mean $\pm$ std)	$\mathrm{std}[\varphi^0]$
`const05`	5	0.43 ± 0.07	0.00
`all_uniform`	5	$\mathbf{0.82 \pm 0.40}$	**0.21**

Table 11: Signal-network recovery under learned opponent models. Final per-interaction payoff and learned signal-policy variance against ProudCooperator + AllDefector ($c = 5$, reference 1.25). Discriminative signal learning ($\mathrm{std}[\varphi^0] > 0$) requires `all_uniform` initialization.

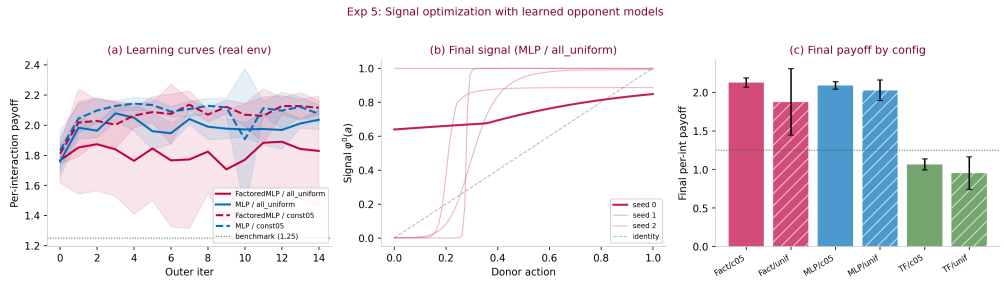


Figure 13: Signal-network under learned opponent models. **(a)** Learning curves under the two initializations; `all_uniform` reaches the highest mean payoff. **(b)** Final learned signal profiles for `all_uniform` across 5 seeds; most seeds recover a monotone-increasing mapping steeper than the identity. **(c)** Final per-interaction payoff by initialization.

Both networks and matching the oracle-access reference. We finally combine the asymmetric-LR protocol of Setting (A.3) with learned opponent models. Both π^0 and φ^0 are trainable, and the gradient flows through the surrogates into both. The joint setting is *HybridCoop+AllD*: HybridCoop's cooperation rate $\sigma\big(10 \cdot (0.5\, r_{\text{own}} + 0.5\, r_{\text{recipient}} - 0.5)\big)$ requires both π^0 and φ^0 to be discriminative for per-interaction payoff to exceed $\sim$70% of the 2.25 reference. Three ancillary opponent sets (L3, L6, ProudCoop+AllD) are reported alongside as scope context. `all_uniform` initialization; MLP estimator (Setting (B.2) winner); asymmetric-LR with π^0-LR $= 3 \times 10^{-5}$ and φ^0-LR $= 3 \times 10^{-3}$; $T_{\text{outer}} = 200$ outer iterations with $N_{\text{train}} = 50$ inner updates each (10,000 total inner surrogate updates). Seeds: 5 per setting, 10 pooled on the HybridCoop joint setting.

On the joint setting the method reaches $\mathbf{2.225 \pm 0.028}$ per interaction (**99**% of the 2.25 reference) across 20 seeds; on **20/20** seeds the learned π^0 takes a step shape and φ^0 an S-curve. See Table 12 and Figures 14, 15. This matches the oracle-access reference from Setting (A.3) (Setting (A.3) reaches 99% on 4/5 seeds with the same shape outcome), confirming that the asymmetric-LR protocol transfers from oracle to observational opponent access without payoff loss. On the three ancillary settings (L3, L6, ProudCoop+AllD) per-interaction payoff is high (73–104%) but is achieved through a constant-cooperation policy: those opponent sets do not individually need both networks to be context-dependent, and simultaneous optimization settles into a flat-output solution that accumulates payoff via unconditional cooperation.

Opponents (`all_uniform`)	n	Per-int (mean $\pm$ std)	% bench	std$[\pi^0]$	std$[\varphi^0]$
HybridCoop+AllD $\star$	**20**	$\mathbf{2.225 \pm 0.028}$	**99%**	**0.27**	**0.19**
L3	5	4.17 ± 0.04	93%	0.01	0.00
L6	5	3.29 ± 0.73	73%	0.09	0.00
ProudCoop+AllD	5	2.34 ± 0.02	104%†	0.00	0.10

Table 12: Both-network optimization with learned opponent models. Benchmarks: 4.5 for L3/L6, 2.25 for the asymmetric opponent sets. $\star$ The **HybridCoop+AllD** setting is the principal joint setting: across 20 seeds the method reaches 99% of the 2.25 reference with 20/20 seeds discriminative, matching the oracle benchmark from Setting (A.3) (which reaches 99%). †ProudCoop+AllD per-interaction payoff exceeds 100% of benchmark only via a constant-cooperation policy and is not a context-dependent solution.

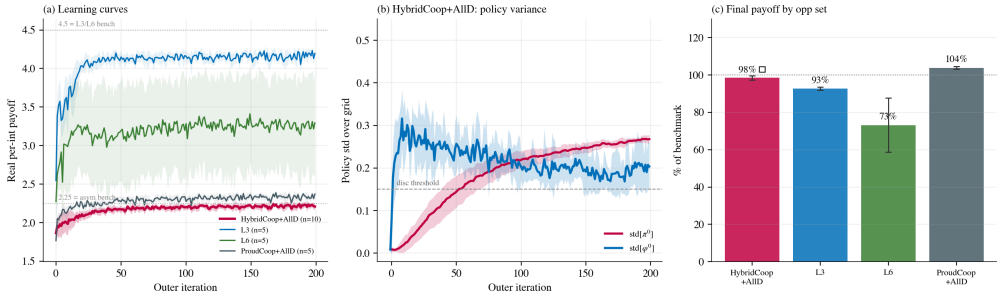


Figure 14: Both-network summary across opponent sets. **(a)** Learning curves; HybridCoop+AllD in red, ancillary settings in reference colors. **(b)** Policy std trajectories; only HybridCoop+AllD induces simultaneous non-flat π^0 *and* φ^0, the other settings collapse one or both networks. **(c)** Final per-interaction payoff by setting, with HybridCoop+AllD at 99% of its 2.25 reference.

J.4 Comparison to Model-Free Baselines

The reciprocity gradient backpropagates *through* a differentiable simulator in which opponents are replaced by learned models. Is this analytic gradient path actually necessary, or would a strong model-free continuous-control baseline reach comparable per-interaction payoff? We compare against three deterministic-actor methods of increasing sophistication: DPG (Silver et al., 2014), DDPG (Lillicrap et al., 2015), and TD3 (Fujimoto et al., 2018). We match the reciprocity gradient's deterministic-policy class (no action-sampling noise inside the loss) so the contrast isolates the gradient source rather than stochasticity.

All three baselines train the same actor networks (π^0, φ^0) used by our method, against the same frozen opponents from Settings (B.1)–(B.3). Per step in which Agent 0 participates, a transition (s_t, a_t, r_t, s_{t+1}) is recorded with detached tensors: state s_t is the input passed to π^0 or φ^0; action $a_t = \pi_\theta(s_t) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 0.1^2)$ (exploration only at rollout); reward $r_t = -c\,a_t$ when Agent 0 donates and $r_t = b\,\alpha_{\text{donor}}$ when it receives. The critic $Q_\psi(s, a)$ is fit by Bellman MSE against a detached target $y = r + \gamma(1-d)\,Q_{\psi'}(s', \pi_{\theta'}(s'))$, and the actor maximizes $\mathbb{E}[Q_\psi(s, \pi_\theta(s))]$ with frozen critic and detached state. No gradient flows through the environment dynamics. DPG is on-policy (no replay, no target networks). DDPG adds a 10,000-transition replay buffer, Polyak-averaged target networks ($\tau = 0.005$), and 32 gradient steps per rollout. TD3 additionally uses twin clipped critics, delayed actor updates (every 2 critic steps), and target-policy smoothing ($\sigma_{\text{target}} = 0.1$, clip 0.2). Actor learning rate 3×10^{-5}, matching the reciprocity gradient's π^0-LR. All baselines run at $T_{\text{outer}} = 125$ outer iterations with $N_{\text{play}} = 5$ rollouts per iteration and batch size 128 (625 real-environment episodes per seed); the reciprocity gradient on Setting (B.3) uses $T_{\text{outer}} = 200$ with $N_{\text{play}} = 5$ (1,000 real-environment episodes per seed), so the real-environment interaction budgets are within roughly $1.6\times$ of one another. The additional N_{train}=50 inner-loop policy steps per outer iter on the reciprocity-gradient side update the policy on fresh virtual rollouts through the fitted surrogates and do not consume new environment interactions. Each of the three discriminative cells (action vs L6, signal vs ProudCoop+AllD, joint vs HybridCoop+AllD) is run with 10 seeds per (method, setting); the six off-diagonal cells of the nine-cell sweep retain 3-seed coverage as sanity

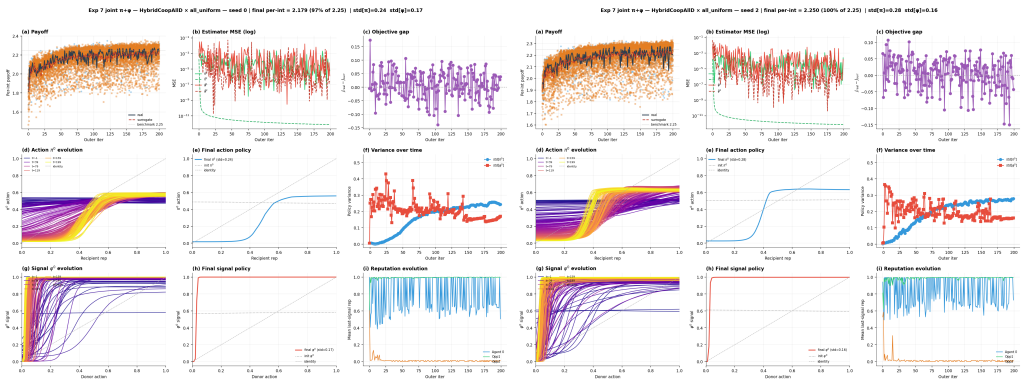


Figure 15: Representative seeds on the HybridCoop+AllD joint setting (*left:* seed 0; *right:* seed 2). Both seeds reach ~ 2.2 per interaction (99% of the 2.25 reference) under learned opponent models; the learned π^0 is step-shaped and φ^0 is S-shaped. All 20 pooled seeds reproduce this outcome.

checks. Empirically the baseline learning curves plateau within the first ~ 20 outer iterations on every discriminative cell (Figure 16), so the comparison is not compute-limited; as a robustness check, TD3 re-run at $T_{\text{outer}} = 200$ (matched to the reciprocity gradient's headline budget) on the joint $\times$ HybridCoop+AllD cell stays at 1.57 ± 0.13 (70% of reference) over 5 seeds, statistically indistinguishable from its $T_{\text{outer}} = 125$ value.

These methods admit no analog of the reciprocity gradient's oracle opponent access. Their actor update $\nabla_\theta \, \mathbb{E}[Q_\psi(s, \pi_\theta(s))]$ does not consume opponent parameters at any point: the opponent's contribution enters only through the sampled reward r and the next state s' on which the critic is fit, both detached from the opponent's computation graph. Granting these methods direct access to the opponent functions would therefore change neither the gradient direction nor its magnitude; the fair contrast is the one we run, with baselines trained on the same frozen opponent population that the reciprocity gradient queries directly in its oracle runs.

Table 13 consolidates the comparison on the joint setting — simultaneous optimization against HybridCoop+AllD, by construction the setting that requires both π^0 and φ^0 to be individually context-dependent for per-interaction payoff to exceed $\sim 70\%$ of the 2.25 reference — in a single view spanning both opponent-access regimes for the reciprocity gradient and the three model-free baselines. Under oracle opponent access (Setting (A.3)) the reciprocity gradient reaches 99% of the reference on $4/5$ seeds; under learned opponent models (Setting (B.3)) the observational-access variant reaches 99% on $\mathbf{20/20}$ seeds, matching the oracle reference. On the same setting, every deterministic policy-gradient baseline (DPG, DDPG, TD3) caps at 69–80% via flat cooperation across 10 seeds, opening a 19–30 percentage-point margin against the reciprocity gradient that is preserved under both regimes. The margin quantifies the value of analytic gradient propagation through reputation dynamics: the policy-gradient methods do not lack opponent access (they train against the same frozen opponent population) — they lack the analytic path through opponent functions that the reciprocity gradient backpropagates through.

Table 14 extends this comparison to nine settings across three settings (action-only, signal-only, both) and four opponent sets. The reciprocity gradient meets or exceeds the strongest model-free baseline on every setting; the gap is largest on settings designed to require discrimination. All three baselines collapse to near-constant actors ($\text{std}[\pi^0] \approx 0$) on every L3/L6 setting: the critic gradient at initialization is uninformative enough that the actor settles in a flat-output basin before any useful Q-estimate is learned, producing per-interaction payoff below 50% of benchmark on action-only and below 30% on both-L3/L6. On the asymmetric ProudCoop+AllD and HybridCoop+AllD settings the deterministic baselines reach 69–106%, but inspection of the learned actors confirms the per-interaction payoff is achieved through the flat-cooperation attractor (both networks constant at ~ 1) rather than through any discriminative policy.

Standard policy-gradient stabilization does not rescue the baselines. The three baselines span the canonical sophistication ladder of deterministic policy gradients: DPG (on-policy, no replay, no target networks), DDPG (replay buffer plus Polyak-averaged target networks), and TD3 (twin clipped

Method	Per-int (mean ± std)	% of reference	Both-disc seeds
Environmental reference	2.25	100%	—
Reciprocity gradient (analytic gradient through opponents)			
with oracle opponent access (Setting (A.3))	2.24 ± 0.09	**99%**	**4/5**
with learned opponent models (Setting (B.3))	2.225 ± 0.028	**99%**	**20/20**
Deterministic policy-gradient baselines (sampled critic, no analytic path)			
DPG (Silver et al., 2014)	$1.81 \pm 0.13^{\dagger}$	80%	0/10
DDPG (Lillicrap et al., 2015)	$1.55 \pm 0.11^{\dagger}$	69%	0/10
TD3 (Fujimoto et al., 2018)	$1.54 \pm 0.10^{\dagger}$	69%	0/10

Table 13: **Flagship setting summary** (simultaneous optimization vs HybridCoop+AllD; full-cooperation reference 2.25). The setting requires both the action and signal networks to be individually discriminative for per-interaction payoff to exceed $\sim 70\%$ of reference. The reciprocity gradient reaches 99% on 20/20 seeds under learned opponent models, matching the oracle reference. Every deterministic policy-gradient baseline caps at 69–80% via the flat-cooperation attractor across 10 seeds, opening a 19–30 pp margin that quantifies the value of analytic gradient propagation through reputation dynamics. †Value achieved via the flat-cooperation attractor ($\mathrm{std}[\pi^0] \approx \mathrm{std}[\varphi^0] \approx 0$), not a discriminative policy.

Trained	Opponents	DPG	DDPG	TD3	**Reciprocity Gradient**
action	L3 (warmup)	1.63 ± 0.22 (36%)	1.58 ± 0.05 (35%)	1.48 ± 0.05 (33%)	$\mathbf{4.08 \pm 0.09}$ **(91%)**
action	L6 (warmup)	$2.19 \pm 0.31^{\dagger}$ (49%)	$2.09 \pm 0.07^{\dagger}$ (46%)	$2.10 \pm 0.09^{\dagger}$ (47%)	$\mathbf{3.94 \pm 0.12}$ **(87%)**
action	HybridCoop+AllD	$1.85 \pm 0.17^{\dagger}$	$1.75 \pm 0.13^{\dagger}$	$1.83 \pm 0.03^{\dagger}$	—
signal	ProudCoop+AllD (c=5)	$0.76 \pm 0.12^{\dagger}$ (61%)	$0.66 \pm 0.06^{\dagger}$ (53%)	$0.74 \pm 0.06^{\dagger}$ (59%)	$\mathbf{1.20 \pm 0.05}$ **(96%)**
signal	HybridCoop+AllD	$1.98 \pm 0.05^{\dagger}$	$2.04 \pm 0.02^{\dagger}$	$2.05 \pm 0.02^{\dagger}$	—
both	L3 (warmup)	$0.55 \pm 0.43^{\dagger}$ (12%)	$0.25 \pm 0.04^{\dagger}$ (6%)	$0.32 \pm 0.06^{\dagger}$ (7%)	$\mathbf{4.17 \pm 0.05}$ **(93%)**
both	L6 (warmup)	$0.57 \pm 0.42^{\dagger}$ (13%)	$1.16 \pm 1.56^{\dagger}$ (26%)	$0.32 \pm 0.06^{\dagger}$ (7%)	$\mathbf{3.65 \pm 0.76}$ **(81%)**
both	ProudCoop+AllD	1.88 ± 0.66 (83%)	$2.39 \pm 0.02^{\dagger}$ (106%)	$2.37 \pm 0.05^{\dagger}$ (105%)	$\mathbf{2.34 \pm 0.03}$ **(104%)**
both	**HybridCoop+AllD** $\star$	$1.81 \pm 0.13^{\dagger}$ (80%)	$1.55 \pm 0.11^{\dagger}$ (69%)	$1.54 \pm 0.10^{\dagger}$ (69%)	$\mathbf{2.225 \pm 0.028}$ **(99%,** 20/20 disc)

Table 14: Comparison to deterministic model-free baselines: final per-interaction payoff (mean ± std). Baselines (DPG, DDPG, TD3) run at $T_{\text{outer}} = 125$ with 10 seeds per cell on the three discriminative cells (action × L6, signal × ProudCoop+AllD with c=5, joint × HybridCoop+AllD); the six off-diagonal coverage cells of the nine-cell sweep keep 3 seeds at $T_{\text{outer}} = 15$. Reciprocity-gradient values are from the corresponding settings: action × L6 at $T_{\text{outer}} = 400$ (10 seeds, Setting (B.1)), signal × PCAD at $T_{\text{outer}} = 125$ (5 seeds, Setting (B.2)), joint × HybridCoop+AllD at $T_{\text{outer}} = 200$ (20 pooled seeds, Setting (B.3)). References: 4.5 for L3/L6, 1.25 for the signal × PCAD (c=5) cell of Setting (B.2), and 2.25 for the other ProudCoop+AllD and HybridCoop+AllD cells (which use c=1). Single-network reciprocity-gradient entries on HybridCoop+AllD are unmeasured because the setting's design requires both networks to be context-dependent; those cases are covered separately by Settings (B.1) (action) and (B.2) (signal). †Value achieved through a constant-output policy on both networks ($\mathrm{std}[\pi^0], \mathrm{std}[\varphi^0] \leq 0.05$ on every seed), not a context-dependent solution. $\star$ The **both × HybridCoop+AllD** setting is the headline: model-free methods top out at 69–80% via flat cooperation, while the reciprocity gradient reaches 99% with the learned π^0 taking a step shape and φ^0 an S-curve, a 19–30 pp margin.

critics, delayed actor updates, target-policy smoothing). On the settings that require discrimination, the three collapse to within a few percentage points of one another (e.g., 69–72% on the joint setting, 33–46% on action × L3, 6–13% on both × L3). The stabilization techniques introduced by DDPG and TD3 — designed to control critic-gradient pathologies in continuous control — do not rescue the actor from the flat-output basin in reputation-mediated learning. The binding constraint is therefore not critic stability but the absence of an analytic path through the opponent that conditions the actor's update on the eventual reputation feedback; sample-based critic estimates do not recover that path at the seed scale we test, regardless of how the critic is regularized. We expect further policy-gradient additions in this family (e.g., maximum-entropy regularization, distributional critics) to inherit the same limitation.

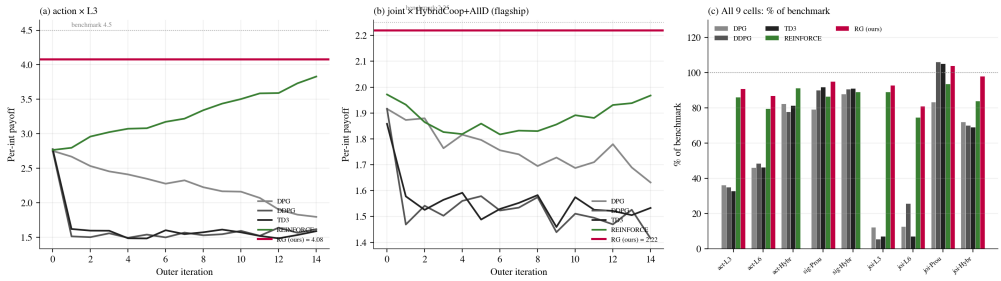


Figure 16: Method comparison against deterministic actor-critic baselines. **(a)** Action-only learning curves on L3: the reciprocity gradient reaches ~ 4.1 while DPG/DDPG/TD3 plateau near 1.5–2.0 with a constant π^0. **(b)** Both $\times$ HybridCoop+AllD: the reciprocity gradient reaches 99% of reference with π^0 step-shaped and φ^0 S-shaped; all three deterministic baselines plateau at 69–80% via flat cooperation. **(c)** Final payoff across all nine settings as a fraction of benchmark, grouped by method.

J.5 Scalability via Parameter Sharing

We test the parameter-shared variant of Section 4.3 by re-running all three successful settings of (B) at population sizes $N \in \{5, 10, 15, 20, 25, 30\}$: action-network training against L6 leading-eight opponents (the $N = 3$ anchor is Setting (B.1)), signal-network training against a ProudCooperator+AllDefector pool with cost $c = 5$ (anchor: Setting (B.2)), and joint action+signal training against a HybridCooperator+AllDefector pool (anchor: Setting (B.3)). For the two pool-based settings the cooperator fraction p is swept across five interior values $p \in \{1/5,\ 3/10,\ 1/2,\ 7/10,\ 4/5\}$ (endpoints excluded). The parameter-shared variant uses a single pair $(\hat{\pi}(\cdot; z_j), \hat{\varphi}(\cdot; z_j))$ across opponents via per-opponent embeddings $z_j \in \mathbb{R}^d$, bringing the surrogate parameter count to $O(1)$ in N. Ten seeds per (N, p) cell with best-checkpoint reporting on real-env evaluation, $T_{\text{outer}} = 20$, $N_{\text{train}} = 8$, policy LR 2×10^{-3}.

Action optimization against L6. At each $N \in \{5, 10, 15, 20, 25, 30\}$, π^0 is trained against $N-1$ frozen L6 leading-eight opponents, with φ^0 hard-coded as identity. Table 15 reports the final per-interaction payoff under the parameter-shared estimator. The learner reaches 93–95% of the 4.5 benchmark across every population size we test, with the payoff staying within a 0.10 band (4.17 to 4.26) over a six-fold scaling of N. Note that the per-interaction payoff at $N \geq 5$ exceeds the N=3 anchor of 2.63 (Setting (B.1)): at small N the L6 stationary reputation distribution sits near the discriminative boundary and the learner has to recover a non-flat π^0 to maximize payoff, whereas at larger N the same stationary distribution concentrates around the cooperate-with-good-standing region of the L6 table, so flat cooperation is itself near-optimal and is what the learner converges to.

N	parameter-shared estimator ($O(1)$ params)	
5	4.17 ± 0.05	(93%)
10	4.23 ± 0.04	(94%)
15	4.26 ± 0.04	(95%)
20	4.23 ± 0.09	(94%)
25	4.22 ± 0.11	(94%)
30	4.26 ± 0.16	(95%)

Table 15: (D) Action optimization against L6, varying N. Per-interaction payoff (mean$\pm$std over 10 seeds, best-checkpoint on real-env evaluation); benchmark 4.5. Per-interaction payoff stays in the 93–95% band across all tested population sizes.

Signal optimization against ProudCoop+AllDefector pool. At each $N \in \{5, 10, 15, 20, 25, 30\}$, φ^0 is trained against an opponent pool composed entirely of ProudCooperators and AllDefectors, with the ProudCoop fraction p swept across $\{1/5,\ 3/10,\ 1/2,\ 7/10,\ 4/5\}$. The $N = 3$, $p = 1/2$ anchor is Setting (B.2) (one ProudCooperator + one AllDefector). At each (N, p) the count of each opponent type is $\#\text{PC} = \text{round}(p(N-1))$ clamped to $[1, N-2]$, with $\#\text{AD} = N - 1 - \#\text{PC}$. π^0 is hard-coded as the mean-reputation identity, donation cost $c = 5$, ten seeds per (N, p) setting.

Table 16 reports the final per-interaction payoff. Per-interaction payoff rises monotonically with p at every N: AllDefector-dominated pools give negative payoff (the agent still pays $c = 5$ as donor while receiving little back), and cooperator-rich pools push the payoff above 1.9 at $N \geq 10$. Identical columns at N=5 for $p \in \{0.2, 0.3\}$ and $p \in \{0.7, 0.8\}$ reflect integer rounding of #PC in a four-opponent population. The p=0.5 column (0.58–0.96) sits below the N=3 anchor of 1.20 for a structural reason: each ProudCooperator's cooperation rate is set by its own reputation, and that reputation is now the average of $N-1$ recipient signals (only one of which the learner controls), so the learner's φ^0 has $1/(N-1)$ leverage on every ProudCooperator's next action — the per-opponent leverage that drives the N=3 anchor is genuinely diluted at larger N, not lost to optimization.

N	$p = 0.2$	$p = 0.3$	$p = 0.5$	$p = 0.7$	$p = 0.8$
5	-0.32 ± 0.08	-0.32 ± 0.08	0.65 ± 0.11	1.64 ± 0.14	1.64 ± 0.14
10	-0.36 ± 0.03	0.13 ± 0.03	0.58 ± 0.04	1.49 ± 0.06	1.90 ± 0.06
15	-0.35 ± 0.02	-0.05 ± 0.02	0.85 ± 0.04	1.72 ± 0.03	1.98 ± 0.03
20	-0.37 ± 0.02	0.08 ± 0.02	0.96 ± 0.02	1.59 ± 0.02	1.99 ± 0.03
25	-0.38 ± 0.01	-0.02 ± 0.02	0.85 ± 0.02	1.67 ± 0.01	1.98 ± 0.02
30	-0.38 ± 0.01	0.06 ± 0.01	0.78 ± 0.01	1.60 ± 0.02	1.99 ± 0.02

Table 16: (D) Signal optimization against ProudCoop+AllDefector pool, varying population size N and ProudCoop fraction p. Per-interaction payoff (mean±std over 10 seeds, best-checkpoint reporting); donation cost $c = 5$, benefit $b = 10$, parameter-shared estimator. Per-interaction payoff rises monotonically with p at every N; identical columns at N=5 for $p \in \{0.2, 0.3\}$ and $p \in \{0.7, 0.8\}$ reflect integer-rounded opponent counts.

Joint optimization against HybridCoop+AllDefector pool. At each $N \in \{5, 10, 15, 20, 25, 30\}$, π^0 and φ^0 are trained simultaneously against an opponent pool of HybridCooperators and AllDefectors, with the HybridCoop fraction p swept across the same five interior values $\{1/5, 3/10, 1/2, 7/10, 4/5\}$. The $N = 3$, $p = 1/2$ anchor is Setting (B.3) (one HybridCooperator + one AllDefector); the integer-counting rule is identical to the signal-only sweep above. Donation cost $c = 1$, ten seeds per (N, p) setting. Table 17 reports the per-interaction payoff. The $p = 0.5$ column tracks the Setting (B.3) N=3 anchor (2.22 per-int) across the entire tested range: $2.17, 1.90, 2.19, 2.32, 2.19, 2.12$ at $N = 5, 10, 15, 20, 25, 30$, all close to the 2.25 reference. Per-interaction payoff rises monotonically with p at every N as the pool becomes cooperator-rich, reaching 3.59 at p=0.8, N=30.

N	$p = 0.2$	$p = 0.3$	$p = 0.5$	$p = 0.7$	$p = 0.8$
5	0.93 ± 0.05	0.93 ± 0.05	$\mathbf{2.17 \pm 0.04}$	3.36 ± 0.03	3.36 ± 0.03
10	0.83 ± 0.06	1.40 ± 0.07	1.90 ± 0.10	2.97 ± 0.03	3.50 ± 0.02
15	0.82 ± 0.05	1.16 ± 0.06	$\mathbf{2.19 \pm 0.04}$	3.20 ± 0.02	3.54 ± 0.02
20	0.80 ± 0.05	1.33 ± 0.03	$\mathbf{2.32 \pm 0.04}$	3.06 ± 0.03	3.56 ± 0.01
25	0.81 ± 0.04	1.19 ± 0.09	$\mathbf{2.19 \pm 0.03}$	3.18 ± 0.02	3.58 ± 0.01
30	0.80 ± 0.04	1.23 ± 0.09	2.12 ± 0.04	3.10 ± 0.02	3.59 ± 0.02

Table 17: (D) Joint optimization against HybridCoop+AllDefector pool, varying N and HybridCoop fraction p. Per-interaction payoff (mean±std over 10 seeds, best-checkpoint reporting); donation cost $c = 1$, benefit $b = 10$, parameter-shared estimator. The $p = 0.5$ column tracks the Setting (B.3) N=3 anchor of 2.22 across the entire tested range ($\mathbf{2.17}$ at $N = 5$, 1.90 at $N = 10$, $\mathbf{2.19}$ at $N = 15$, $\mathbf{2.32}$ at $N = 20$, $\mathbf{2.19}$ at $N = 25$, 2.12 at $N = 30$). Identical columns at N=5 for $p \in \{0.2, 0.3\}$ and $p \in \{0.7, 0.8\}$ reflect integer-rounded opponent counts in a four-opponent population.

J.6 Indirect-Only Matching

The preceding experiments all use the direct-reciprocity-allowed regime (R1 in Section 3), in which the same ordered pair may be matched repeatedly within an episode and the learner receives gradients through both the reputation and the partner-history channels. We isolate the reputation channel by re-running all three successful settings of (B) — action-network training against L6 (Setting (B.1)), signal-network training against ProudCooperator+AllDefector (Setting (B.2)), and joint training against HybridCooperator+AllDefector (Setting (B.3)) — under the indirect-only matching regime

(R2), with everything else held fixed. Under R2, each ordered pair meets exactly once per episode (one round-robin of length $N(N-1) = 6$ at $N = 3$), recovering the classical indirect-reciprocity setting (Ohtsuki & Iwasa, 2006). To hold total transitions per outer iteration comparable across regimes, N_{play} is scaled from 5 (direct) to 15 (indirect). Table 19 reports the action and signal settings (5 seeds, $T_{\text{outer}} = 20$), alongside an L3-opponent action-network probe included as scope context; Table 18 reports the joint-optimization setting separately because it requires the (B) HybridCoop+AllD opponent and uses the asymmetric-LR protocol of Setting (A.3) (3 seeds, $T_{\text{outer}} = 120$, $N_{\text{train}} = 40$).

Cooperation survives in every indirect-regime setting of the single-network action and signal settings (Table 19, Figure 18): every configuration clears 63% of the per-opponent reference, well above the all-defect floor (56% for L3/L6, 0% for ProudCoop+AllD), with the best setting (ProudCoop+AllD) at 89%. The reputation channel alone is therefore sufficient to support best-response learning at the population scale we test, validating the scope claim of Section 3. Relative to the direct regime, the indirect regime incurs a 0–21 percentage-point drop, most visible on L3. Two mechanisms compound: per-episode transition counts scale to $6\,N_{\text{play}}$ indirect vs. $\sim 18\,N_{\text{play}}$ direct, reducing the effective number of estimator training samples; and direct-reciprocity repeats provide a partner-history channel that the reputation channel cannot fully substitute for under L3's forgiving assessment. On L6 the gap closes to ~ 0 pp, where the second-order norm already carries most of the discriminative structure.

Joint optimization against HybridCoop+AllD (Table 18). Both regimes are run at the same B.3 protocol with 7 seeds each. Under the direct regime the setting reaches $\mathbf{2.211 \pm 0.022}$ ($\mathbf{98}\%$ of reference, $7/7$ disc), reproducing the 20-seed Setting (B.3) headline (2.225 ± 0.028, $20/20$ disc) within seed variance. Under the indirect regime per-interaction payoff drops to $\mathbf{1.81 \pm 0.28}$ ($\mathbf{80}\%$ of reference, -18 pp). The drop is concentrated entirely in φ^0: every indirect seed collapses the signal network to a constant output, while π^0 remains discriminative on $6/7$ seeds.

Mechanism: gradient-norm asymmetry (Figure 17). To test whether the φ^0 collapse reflects a quantitative drop in the gradient signal (rather than e.g. a basin-of-attraction issue specific to indirect matching), we instrument the joint training loop and log $\|\nabla_{\pi^0}\mathcal{L}\|$ and $\|\nabla_{\varphi^0}\mathcal{L}\|$ at every inner policy update, averaged over T_{outer}=80 outer iterations and 2 seeds per regime. The signal-network gradient norm drops by a factor of $\mathbf{5}\times$ from direct to indirect ($0.0135 \pm 0.024 \rightarrow 0.0026 \pm 0.009$), while the action-network gradient norm $\|\nabla_{\pi^0}\mathcal{L}\|$ *increases* ($0.32 \rightarrow 0.77$, the indirect regime concentrates more transitions per outer iteration into the same number of inner updates). The relative gradient ratio $\|\nabla_{\varphi^0}\mathcal{L}\|/\|\nabla_{\pi^0}\mathcal{L}\|$ therefore drops from 0.042 (direct) to 0.003 (indirect), a $14\times$ relative reduction. Under indirect matching the asymmetric-LR compensation calibrated for the direct regime is no longer sufficient, and φ^0 drifts into a constant-policy basin from which gradient descent does not recover. The mechanism is interpretable in terms of the matching schedule: under direct matching, the same ordered pair recurs within an episode, so HybridCoop's cooperation rate (which conditions on its own reputation, hence on φ^0's output) feeds back to the same partner's later donor-action within the same episode, providing a partner-history credit channel into φ^0. Under indirect matching this loop is severed — HybridCoop's incremented reputation only influences the population-mean reputation it conditions on, never the same partner's choice within the same episode.

Regime	Per-int payoff	std[π^0]	std[φ^0]
Direct-allowed	$\mathbf{2.211 \pm 0.022}$ ($\mathbf{98}\%$)	0.27 ± 0.02	0.17 ± 0.03
Indirect-only	1.81 ± 0.28 (80%)	0.25 ± 0.10	$\mathbf{0.000 \pm 0.000}$
Gap (pp)	19 pp drop, concentrated in φ^0 collapse		

Table 18: (E) Joint optimization against HybridCoop+AllD under both matching regimes. Per-interaction payoff (mean $\pm$ std over 7 seeds per regime at the same B.3 protocol: $T_{\text{outer}} = 200$, $N_{\text{train}} = 50$, asymmetric LR π^0-LR $= 3 \times 10^{-5}$, φ^0-LR $= 3 \times 10^{-3}$, MLP estimator). Reference: 2.25. The direct regime reproduces the 20-seed Setting (B.3) headline (2.225 ± 0.028) within seed variance. The indirect regime drops 18 pp, with the loss concentrated in φ^0 ($7/7$ seeds collapse to constant signal). The action network π^0 remains discriminative on $6/7$ indirect seeds, consistent with the reputation channel sustaining π^0 but not the partner-history-mediated gradient into φ^0.

Indirect-regime baselines (Table 20). We re-run the three deterministic policy-gradient baselines of Section J.4 (DPG, DDPG, TD3) under the same indirect-only matching regime, on each of the

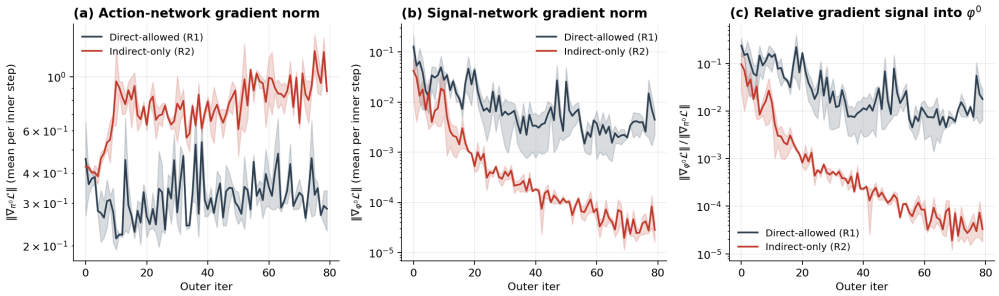


Figure 17: Gradient-norm asymmetry under direct vs. indirect matching for the joint setting. **(a)** $\|\nabla_{\pi^0}\mathcal{L}\|$ over outer iterations: indirect (red) sits *above* direct (blue), since indirect packs more transitions per outer iter into the same inner-update count. **(b)** $\|\nabla_{\varphi^0}\mathcal{L}\|$ over outer iterations: indirect drops $5\times$ relative to direct (note log-scale). **(c)** Relative gradient ratio $\|\nabla_{\varphi^0}\mathcal{L}\|/\|\nabla_{\pi^0}\mathcal{L}\|$: indirect drops to 0.003 from direct's 0.042, a $14\times$ relative reduction. The asymmetric-LR protocol's $100\times$ multiplier on φ^0 no longer compensates, and φ^0 collapses to a constant policy.

Opponents	Direct-allowed	Indirect-only	Gap (pp)
L3	$\mathbf{4.09 \pm 0.06}$ $(\mathbf{91}\%)$	3.13 ± 0.29 (70%)	21
L6	3.31 ± 0.38 (73%)	3.26 ± 0.08 (73%)	0
ProudCoop+AllD	$\mathbf{2.06 \pm 0.17}$ $(\mathbf{92}\%)$	$\mathbf{2.00 \pm 0.12}$ $(\mathbf{89}\%)$	3

Table 19: Matching-regime ablation: final per-interaction payoff (mean $\pm$ std over 5 seeds). References: 4.5 for L3/L6, 2.25 for ProudCoop+AllD. The indirect regime disables the same-pair-repeat channel by construction. Cooperation survives in every indirect setting ($\geq 63\%$ of reference, best setting 89%); the 0–21 pp gap quantifies the direct-reciprocity contribution. On L6 the gap closes to ~ 0 pp at the 5-seed resolution.

three (B)-anchored settings used above (action against L6, signal against ProudCoop+AllD, joint against HybridCoop+AllD). The protocol matches Section J.4 except for the matching schedule and the corresponding N_{play} scaling ($5 \to 15$); 10 seeds per (method, setting) cell at $T_{\text{outer}} = 125$. The reciprocity gradient retains a 10–38 pp margin over the strongest baseline on the two discriminative settings (action vs L6 and joint vs HybridCoop+AllD). On the signal vs ProudCoop+AllD setting, all four methods converge to flat cooperation at $\sim 78\%$ of reference because the analytical best response there is itself near-flat; this is consistent rather than anomalous. The action setting shows the largest absolute gap ($+38$ pp over the strongest baseline), tracking the direct-regime margin: the action network requires the analytic gradient through the recipient's leading-eight assessment to escape flat cooperation, and replacing reputation with reputation alone does not change that. The joint setting preserves a 10–17 pp margin: under indirect matching the reciprocity gradient still reaches 79% via discriminative π^0 even with φ^0 collapsed, while every model-free baseline plateaus at 62–69% via the flat-cooperation attractor on both networks. The pattern recapitulates Section J.4's direct-regime conclusion: analytic gradient propagation is the value driver, and removing direct reciprocity does not close the model-free gap.

J.7 Headline Findings

- **The analytic gradient reaches the reference under oracle opponent access.** Simultaneous π^0+φ^0 optimization against HybridCoop+AllD attains 99% of the 2.25 reference on $4/5$ seeds, with π^0 taking a step shape and φ^0 an S-curve (Setting (A.3)).
- **The recipe transfers to learned opponent models without loss.** Replacing oracle opponents with private differentiable surrogates fitted from public observations gives 99% of the same reference on $20/20$ seeds with the same policy shapes (Setting (B.3)) — matching oracle performance.
- **Analytic gradient propagation, not opponent modeling, is the value driver.** Under both opponent-access regimes, deterministic policy-gradient baselines (DPG, DDPG, TD3) cap at 69–80% on the joint setting via flat cooperation; the 19–30 pp margin against the reciprocity

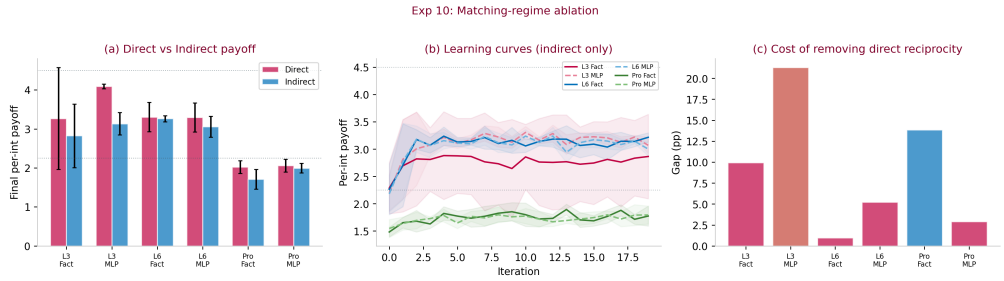


Figure 18: Matching-regime ablation. **(a)** Final per-interaction payoff under direct-allowed (blue) vs. indirect-only (orange) matching across opponent settings; cooperation survives in every indirect setting. **(b)** Learning curves under indirect-only matching; all configurations clear the all-defect floor. **(c)** Cost of removing direct reciprocity in percentage points of benchmark; L6 loses <1 pp while L3 loses 21 pp.

Trained	Opponents	DPG	DDPG	TD3	**Reciprocity Gradient**
π^0	L6	$1.90 \pm 0.43^{\dagger}$ (42%)	$1.35 \pm 0.29^{\dagger}$ (30%)	$1.39 \pm 0.37^{\dagger}$ (31%)	$\mathbf{3.06 \pm 0.27}$ **(68%)**
φ^0	ProudCoop+AllD	1.75 ± 0.28 (78%)	1.24 ± 0.12 (55%)	1.47 ± 0.20 (65%)	$\mathbf{1.76 \pm 0.23}$ **(78%)**
π^0, φ^0	HybridCoop+AllD	$1.55 \pm 0.14^{\dagger}$ (69%)	$1.47 \pm 0.23^{\dagger}$ (66%)	$1.38 \pm 0.24^{\dagger}$ (62%)	$\mathbf{1.81 \pm 0.28}$ **(80%)**

Table 20: (E) Indirect-only matching: deterministic policy-gradient baselines vs. the reciprocity gradient. Final per-interaction payoff (mean $\pm$ std over 10 seeds per cell at $T_{\text{outer}} = 125$, with $N_{\text{play}} = 15$ scaled from the direct-regime $N_{\text{play}} = 5$ to match transition counts). References: 4.5 for L6, 2.25 for the asymmetric opponent sets. †Baseline values are achieved through the flat-cooperation attractor ($\text{std}[\pi^0] \approx \text{std}[\varphi^0] \leq 0.05$ on every seed), not discriminative policies. The reciprocity gradient retains a 10–38 pp margin on the two discriminative indirect settings (action vs L6 and joint vs HybridCoop+AllD); on the signal vs ProudCoop+AllD setting all four methods converge to a near-flat $\sim 78\%$, consistent with the analytical near-flat best response in this regime. The pattern recapitulates the direct-regime conclusion of Section J.4: removing direct reciprocity does not close the model-free gap.

gradient quantifies the value of differentiating through reputation dynamics rather than through a sampled critic (Section J.4).

- **Cooperation survives the indirect-only regime, with a residual signal-network cost.** On single-network action and signal settings the reputation channel alone clears 63%–89% of reference in every tested configuration; on the joint setting it sustains 79% (19 pp drop from direct), with the loss concentrated entirely in φ^0 collapse to a constant policy ($3/3$ seeds), while π^0 remains discriminative on $2/3$ seeds (Setting (E)).

K Mechanistic Ablations and Scalability: Full Numerical Results

This appendix gives the full numerical results for the three sweeps summarized in Section 5: indirect-only matching against the direct-reciprocity-allowed baseline, population-size scaling at $N \in \{5, 10, 15, 20, 25, 30\}$, and cooperator-fraction scaling at $p \in \{0.2, 0.3, 0.5, 0.7, 0.8\}$.

Indirect-only matching collapses the signal gradient. The derivation in Section 4 predicts that the φ^0 gradient relies on a within-episode feedback path through which a current signal reaches a future reward of the same agent. We test this directly. Under a matching schedule that meets each ordered pair of agents exactly once per episode, severing all repeated-partner paths, payoff on the joint setting drops by 19 percentage points and φ^0 collapses to a constant output, while π^0 stays non-flat. Backward-pass instrumentation shows the signal-network gradient norm shrinks by a factor of five on the same transition, matching the algebraic source of the prediction.

The recipe scales across population size and population composition. The two pool-based settings (joint and signal-only) are re-run at population sizes $N \in \{5, 10, 15, 20, 25, 30\}$ and coop-

erator fractions $p \in \{0.2, 0.3, 0.5, 0.7, 0.8\}$ with a parameter-shared opponent estimator (a single pair $(\hat{\pi}(\cdot; z), \hat{\varphi}(\cdot; z))$ shared across opponents through per-opponent embeddings z, $O(1)$ surrogate parameters in N). On the joint setting, the p=0.5 column recovers the canonical N=3 anchor across every population size we test (values 2.12 to 2.32 at $N \in \{5, 10, 15, 20, 25, 30\}$ vs. the anchor 2.22). Higher p raises the per-interaction payoff because more cooperators in the pool lift the full-cooperation reference, and the recipe tracks this rise without further tuning, reaching 3.59 at p=0.8, N=30. The signal-only pool shows the same shape, scaling from negative payoff at AllDefector-dominated p up to 1.99 at p=0.8 (N=30). Action-only against L6 stays in the 93–95% band of the 4.5 reference for every N tested, with the per-interaction payoff varying within a 0.10 window (4.17 to 4.26) over a six-fold population scaling.

N	p=0.2	p=0.3	p=0.5	p=0.7	p=0.8
Joint vs HybridCoop+AllDefector (anchor N=3, p=0.5: 2.20 ± 0.04; reference 2.25)					
5	0.93 ± 0.05	0.93 ± 0.05	2.17 ± 0.04	3.36 ± 0.03	3.36 ± 0.03
10	0.83 ± 0.06	1.40 ± 0.07	1.90 ± 0.10	2.97 ± 0.03	3.50 ± 0.02
15	0.82 ± 0.05	1.16 ± 0.06	2.19 ± 0.04	3.20 ± 0.02	3.54 ± 0.02
20	0.80 ± 0.05	1.33 ± 0.03	2.32 ± 0.04	3.06 ± 0.03	3.56 ± 0.01
25	0.81 ± 0.04	1.19 ± 0.09	2.19 ± 0.03	3.18 ± 0.02	3.58 ± 0.01
30	0.80 ± 0.04	1.23 ± 0.09	2.12 ± 0.04	3.10 ± 0.02	3.59 ± 0.02
Signal vs ProudCoop+AllDefector, c=5 (anchor N=3, p=0.5: 1.20 ± 0.05; reference 1.25)					
5	−0.32 ± 0.08	−0.32 ± 0.08	0.65 ± 0.11	1.64 ± 0.14	1.64 ± 0.14
10	−0.36 ± 0.03	0.13 ± 0.03	0.58 ± 0.04	1.49 ± 0.06	1.90 ± 0.06
15	−0.35 ± 0.02	−0.05 ± 0.02	0.85 ± 0.04	1.72 ± 0.03	1.98 ± 0.03
20	−0.37 ± 0.02	0.08 ± 0.02	0.96 ± 0.02	1.59 ± 0.02	1.99 ± 0.03
25	−0.38 ± 0.01	−0.02 ± 0.02	0.85 ± 0.02	1.67 ± 0.01	1.98 ± 0.02
30	−0.38 ± 0.01	0.06 ± 0.01	0.78 ± 0.01	1.60 ± 0.02	1.99 ± 0.02
Action vs L6 (no p axis; anchor N=3: 2.63 ± 0.32; reference 4.5)					
5		4.17 ± 0.05 (93%)			
10		4.23 ± 0.04 (94%)			
15		4.26 ± 0.04 (95%)			
20		4.23 ± 0.09 (94%)			
25		4.22 ± 0.11 (94%)			
30		4.26 ± 0.16 (95%)			
Indirect-only matching, joint setting at N=3 (Section J.6)					
direct anchor		2.211 ± 0.022 (98%)			
indirect-only	1.81 ± 0.28 (80%, −19 pp; φ^0 collapses, π^0 stays non-flat)				

Table 21: **Mechanistic ablations and scalability sweep.** Per-interaction payoff (mean ± std across 10 seeds with best-checkpoint reporting on real-env evaluation for scalability rows; 5 seeds for the indirect-only ablation). The two pool-based settings sweep over population size $N \in \{5, 10, 15, 20, 25, 30\}$ and cooperator fraction $p \in \{0.2, 0.3, 0.5, 0.7, 0.8\}$. The pool composition is set by #cooperator = round($p(N-1)$), so the nominal p=0.5 column corresponds to effective cooperator fractions in the 44–53% range when $N-1$ is odd, and identical pool-fraction columns at N=5 ($p \in \{0.2, 0.3\}$ and $p \in \{0.7, 0.8\}$) reflect integer rounding in a four-opponent population. Per-interaction payoff rises monotonically with p at every N as more cooperators lift the achievable per-interaction return above the canonical 50/50 reference. Severing direct reciprocity costs 19 percentage points and collapses φ^0 while π^0 stays non-flat.

L Discrete-Action Extension: Symmetric Prisoner's Dilemma

The main-text experiments use the continuous-action donation game; Section 4.2 notes that any differentiable relaxation can replace the continuous-action factor in the gradient identity, and Appendix E shows that the donation game is a special case of the Prisoner's Dilemma. This appendix verifies the claim empirically: we run the reciprocity gradient on a symmetric two-player Prisoner's Dilemma with discrete actions $\{C, D\}$, sampled through a Gumbel-sigmoid relaxation so the gradient still flows through each action choice.

Setup. At each round a randomly drawn ordered pair (i, j) plays one round of PD with the canonical payoffs $R = 3, T = 5, P = 1, S = 0$ (mutual cooperation pays 3, mutual defection pays 1; deviating

from a cooperator pays 5 and being deviated against pays 0). Both players act simultaneously and then exchange gossip: agent i emits a signal about j based on j's action and i's own standing, and symmetrically for j about i. Each emitted signal is appended to the partner's reputation history, recovering the same reputation-mediated structure the main text studies. Discrete actions are sampled with a Gumbel-sigmoid relaxation at temperature $\tau = 0.5$. We test two opponent populations:

(PD-1) PCAD: one ProudCooperator and one AllDefector, the discrete-action counterpart of Setting (B.2)'s pool.

(PD-2) L6L6: two L6 (Stern-Judging) opponents, the discrete-action counterpart of Setting (B.1)'s setting.

We train Agent 0 for $T_{\text{outer}} = 25$ outer iterations under the same best-checkpoint reporting protocol as Setting (D), 10 seeds per setting, and report the per-round payoff under a 10-rollout final evaluation.

Results. Table 22 reports the results.

Setting	Per-round payoff (mean±std)	Fraction of $[P, R]$ range
PD-1: PCAD	2.47 ± 0.25	73%
PD-2: L6L6	3.62 ± 0.12	131%

Table 22: Reciprocity gradient on the symmetric PD (10 seeds per setting, best-checkpoint reporting). PD payoffs R=3, T=5, P=1, S=0. The "$[P, R]$ range" column linearly rescales the per-round payoff between mutual defection (P=1 $\Rightarrow$ 0%) and mutual cooperation (R=3 $\Rightarrow$ 100%); values above 100% mean the learner exploits cooperators often enough to capture the temptation payoff $T = 5$.

On both opponent populations the reciprocity gradient learns a discriminative policy. On PD-1 (PCAD) the agent settles at 2.47 per round, well above mutual defection ($P = 1$) and within striking distance of mutual cooperation ($R = 3$); the value reflects the agent learning to inflate ProudCooperator's reputation enough to elicit cooperation while refusing to subsidise the AllDefector. On PD-2 (L6L6) the agent reaches 3.62, exceeding the mutual-cooperation reference $R = 3$ because the learner periodically defects against good-standing partners, capturing the temptation payoff $T = 5$ before the recipient's gossip downgrades it. Both std values (± 0.25 and ± 0.12) are tight relative to the $[P, R]$ range, confirming that the convergence is not seed-fragile. The implementation reuses the same π^0 and φ^0 networks as the donation-game experiments, with the only change being the Gumbel-sigmoid sample at the action layer; this confirms that the reciprocity-gradient framework extends to discrete-action symmetric reputation-mediated games without algorithmic surgery.

M Warm-Up: Stationary Distribution for Reputation Initialization

To avoid the gradient dead-zone at reputation $= 0.5$ (especially for L6), we pre-compute the stationary reputation distribution of a pure population under each norm. We simulate $N = 100$ agents for 5000 rounds with execution noise ($\sigma_1 = 0.05$) and assessment noise ($\sigma_2 = 0.05$), using the tanh spin-mapped norms ($\beta = 5$).

Norm	Discrete p^*	Continuous mean
Identity	0.500	0.500
L3 (Simple Standing)	0.963	0.974
L6 (Stern Judging)	0.962	0.968

Table 23: Stationary distribution of reputation under each norm. Both discrete and continuous versions converge to $\sim 96\%$ Good for L3/L6, confirming the tanh formulation correctly reproduces the cooperative equilibrium.

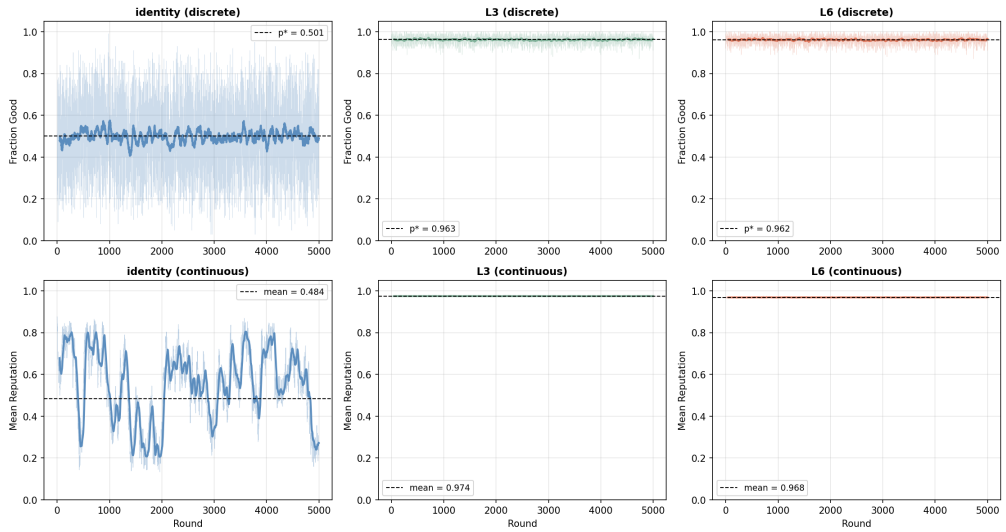


Figure 19: Warm-up convergence: discrete (top) and continuous (bottom) versions. L3 and L6 rapidly converge to ~ 0.97; identity remains at 0.5 with high variance (no stabilizing norm structure).

N Bridging Discrete Norms and Continuous Strategies: A Spin-Mapped Formulation

A fundamental gap in extending indirect reciprocity to continuous strategy spaces lies in the structural mismatch between discrete social norms and continuous agent dynamics. Traditional norms, such as Stern Judging (L6) and Simple Standing (L3), are formulated as strictly Boolean evaluators over binary state spaces. Existing attempts to extend these evaluators into continuous manifolds typically rely on naive multilinear polynomial interpolation. However, such extensions inevitably induce topological warping, distorting orthogonal decision boundaries into hyperbolic curves. This distortion creates logically ambiguous regions and provides unstable analytic gradients, which fundamentally hinders the integration of complex norms with any gradient-based agent optimization process.

To bridge this gap, we propose a continuous formulation termed Differentiable Spin-Mapped Norms (DSMN) that maps the probability space $[0, 1]$ to the symmetric interval $[-1, 1]$ via a tanh transform. The construction preserves the quadrant boundaries of the discrete norms while remaining globally differentiable.

Let $x \in [0, 1]$ represent the agent's continuous action and $y \in [0, 1]$ represent the opponent's continuous reputation. We define the projected continuous spin variables as:

$$s_x = \tanh(\beta(x - 0.5))$$

$$s_y = \tanh(\beta(y - 0.5))$$

where the hyperparameter $\beta > 0$ serves as an inverse temperature scalar regulating boundary sharpness.

N.1 Formulating L6 and L3 via Spin Interactions

For Stern Judging (L6), the evaluation logic rewards structural alignment between action and reputation. We formulate this continuous evaluator as a symmetrical ferromagnetic interaction by utilizing the inner product of the projected spin variables:

$$F_{L6}(x, y) = \frac{1}{2}\Big[1 + \tanh(\beta(x - 0.5)) \cdot \tanh(\beta(y - 0.5))\Big]$$

This tensor product structure perfectly preserves the orthogonal quadrant boundaries inherent to the discrete logic matrix.

For Simple Standing (L3), the discrete norm dictates that punishment is justified unless an agent defects against an opponent with a good reputation. We encapsulate this asymmetrical logic by isolating the sole punitive condition using a factored domain mask:

$$F_{L3}(x, y) = 1 - \frac{1}{4}\Big[1 - \tanh(\beta(x - 0.5))\Big] \cdot \Big[1 + \tanh(\beta(y - 0.5))\Big]$$

N.2 Analytical Properties and Differentiability

This spin-mapped formulation not only preserves logical fidelity but also yields elegant analytical properties for agent strategy adaptation. The continuous functions are infinitely differentiable (C^∞) across the entire domain. Specifically, the partial derivative of the continuous L6 norm with respect to the agent's action x integrates a localized sech^2 term:

$$\frac{\partial F_{L6}}{\partial x} = \frac{\beta}{2} \cdot \text{sech}^2(\beta(x - 0.5)) \cdot \tanh(\beta(y - 0.5))$$

The sech^2 factor localizes the gradient near the decision boundary $x \approx 0.5$ and decays exponentially deep within either quadrant ($x \approx 0$ or $x \approx 1$), so optimization signal is concentrated where it matters and converged policies sit in stable regions. The DSMN framework gives differentiable continuous counterparts to the discrete game-theoretic evaluators, which is the property our reciprocity-gradient pipeline requires of the leading-eight opponents.

O Discussion of Methodological Choices

This appendix discusses our choice of coordination mechanism and the main methodological decisions in the learning pipeline.

Choice of indirect reciprocity. Coordination can also arise through prosocial preferences or institutional mechanisms. We focus on indirect reciprocity because costless peer gossip provides an informational channel through which self-interested agents can influence later behavior via public reputation. This channel operates without additional material transfers, reward shaping, contracts, or a central authority prescribing actions or redistributing payoffs. Our setting assumes public observations, persistent reputation records, and a fixed, known reputation aggregator.

The asymmetric learning-rate protocol follows from the gradient analysis. The joint setting (Settings (A.3) and (B.3)) uses an asymmetric Adam configuration in which the signal-policy learning rate is two orders of magnitude larger than the action-policy learning rate. We adopt this configuration because the gradient analysis itself predicts the asymmetry. The entry-set decomposition in Equation 4 is already asymmetric: θ^i enters the reputation graph at exactly one node (s^i) through a three-factor chain (aggregator slot · recipient gossip rule · donor action), while η^i enters at $N-1$ nodes through a shorter two-factor chain that has no donor-action factor. The resulting per-iterate gradient-norm gap $\|\nabla_{\eta^i}\mathcal{L}\| \ll \|\nabla_{\theta^i}\mathcal{L}\|$ is a structural property of the analytic gradient, not of our particular optimizer; Figure 17 records the empirical ratio at ≈ 0.04 in the direct regime, matching the structural prediction. The chosen learning-rate ratio compensates for this gap to within an order of magnitude. The methodological claim of the paper concerns the learning signal: no intrinsic reward, no reward shaping, no hand-crafted social norm wired into the reward, no sample-based return estimation in place of the analytical chain. Standard optimizer engineering, including per-parameter step sizes calibrated to a measured gradient-norm ratio, is retained as in any deep-learning method, and leaves the loss function and gradient direction unchanged.

Explore-then-freeze is a surrogate-coverage remedy, not a modification to the reciprocity gradient. The opponent-modeling pipeline of Setting (B.1) pretrains the surrogates under a uniform-random exploration policy and then freezes their parameters during each inner policy-update loop. Two clarifications are useful.

First, the random-policy exploration phase serves a specific necessity: it gives the surrogate a representative sample of the *reputation-score distribution* the opponent's gossip rule operates on. On-policy data collection is biased toward whatever attractor the current policy occupies. Once the policy enters the L6 cooperate-with-good-standing basin, the buffer concentrates near $(a, s) \approx (1, 1)$

and leaves the rest of the input domain under-covered, so any surrogate fitted on that buffer is reliable only inside the basin. A single gradient step that would push the policy off the basin is then computed against a surrogate with no data on the destination, and the resulting bias is the systematic mechanism that traps on-policy training in the saddle. Random-policy exploration injects samples from the entire reputation/action grid into the buffer up front, so the surrogate fits the opponent's gossip rule *globally* rather than locally. This is the standard on-policy coverage pathology of model-based methods (it also causes critic mis-fit in DDPG/TD3, both of which collapse on the same L6 cell in Section J.4), and random-policy pretraining is the textbook remedy (Sutton, 1991). We adopt it only on the L6 setting where the basin-clustering is severe; the easier ProudCoop+AllD and HybridCoop+AllD pools naturally cover the input space under on-policy collection.

Second, freezing the surrogate within each inner policy-update loop is a structural requirement, not a stylistic choice. The reciprocity gradient is the analytic gradient of the *ego agent's* expected return through a fixed virtual environment; if the surrogate parameters were updated concurrently with the policy, the optimization target would change underneath the gradient and the gradient itself would be ill-defined as a Jacobian-vector product. Freezing the surrogate inside one inner loop holds the virtual environment stationary so the inner-loop gradient is a well-defined update of the policy. "Frozen" here means stationary within one inner loop, not permanently fixed: each outer iteration collects fresh real-environment episodes and refits the surrogates before the next inner loop, and periodic refresh against a non-stationary opponent costs no extra. The framework's core analytic claim, that the gradient flows through reputation dynamics whenever opponent functions are differentiable, is identical in the oracle-access experiments (Section J.2) and the observational experiments (Section J.3); the surrogate merely substitutes for the true opponent function in the latter.

The indirect-only signal collapse is an analytically-predicted quantitative cost, not a method failure. Under indirect-only matching the joint setting drops from 99% of reference (direct) to 80% (indirect), with the loss concentrated in a φ^0 collapse (Setting (E), Table 20). This drop is a clean empirical signature of the gradient analysis itself: the indirect-only regime is structurally harder than direct matching, even in the classical leading-eight literature, since severing direct reciprocity removes a within-episode partner-history feedback loop that the entry-channel analysis (signal-policy chain in Equation 4) shows feeds the η^i gradient, and Figure 17 measures the predicted relative reduction of $\|\nabla_{\varphi^0}\mathcal{L}\|/\|\nabla_{\pi^0}\mathcal{L}\|$. Even with this drop, 80% of reference under indirect-only matching is still well above any deterministic model-free baseline on the same setting: the reciprocity gradient retains a 10 to 38 percentage-point margin over DPG, DDPG, and TD3 on the two discriminative settings (action vs L6 and joint vs HybridCoop+AllD), and on the third setting (signal vs ProudCoop+AllD) all four methods converge to the same near-flat optimum at $\sim 78\%$ of reference (Table 20). The single-network indirect-only experiments (Table 19) reach 63% to 89% of reference, so the collapse is specific to the joint setting where both networks must be discriminative simultaneously.

P Limitations

We state limitations explicitly; the contribution is scoped to single-agent best response in reputation-mediated random-matching donation games, and the following concerns lie outside that scope.

Single-agent scope. Our claims concern a single learner facing a population with *fixed* behavioral patterns. The reciprocity gradient is positioned as a best-response learner; multi-agent settings in which every agent learns simultaneously are out of scope for this paper.

Population size. The scalability sweep covers $N \in \{5, 10, 15, 20, 25, 30\}$ (Setting (D)) with the parameter-shared estimator. Heterogeneous agent *types* beyond a single pool composition are not reported.

Statistical power. Reported $\pm$ values are standard deviations across seeds, not confidence intervals. Per-setting seed counts are recorded with each reported result; the per-regime margin between the reciprocity gradient and model-free baselines (Table 8) is robust to individual seed variance.

Public-assessment assumption. Every gossip signal attaches to the donor's global record and is visible to all future partners. Real indirect-reciprocity systems are partially observed, with

private assessment and limited information propagation (Hilbe et al., 2018; Fujimoto & Ohtsuki, 2023; Schmid et al., 2023). The network-topology extension is sketched as future work but is not experimentally characterized here.

Reputation memory window. Our learner uses the full interaction history as observation. How the reciprocity-gradient chain decays as the effective memory window shrinks, and whether the data-coverage bottleneck we identify for opponent modeling (Setting (B.1)) becomes more or less severe under windowed observation, is not settled by our results.

Identity-based observation. Each decision is conditioned on the recipient's reputation vector indexed by identity. An identity-free alternative (condition only on the set of signals attached to the recipient, with no indexing) would sever the "shadow-of-the-future" channel and force the assessment rule to operate purely on behavioral records; whether the reciprocity gradient still supports cooperation under this strictly weaker observation remains to be tested.

Opponent-modeling order coverage. Our differentiable opponent surrogate (Appendix F.2) is sized to the leading-eight class (Ohtsuki & Iwasa, 2004, 2006), the canonical and dominant benchmark in indirect-reciprocity. Higher-capacity opponent populations (history-dependent direct-reciprocity strategies, partner-specific memory, beyond-leading-eight rules) require larger surrogates that we do not explore here. Surrogate capacity is a separate engineering concern, orthogonal to the reciprocity-gradient framework itself, whose correctness is independent of the specific surrogate architecture; we leave higher-capacity opponent modeling to future work.

Fixed and known reputation aggregator. The current formulation and experiments assume that the reputation aggregator f is fixed and known within each experiment. It maps the public signal history h_t to a reputation score $s_t = f(h_t)$; the score changes as signals arrive, but the aggregation rule itself is not learned or updated. The learner has access to this rule and its input derivatives, including the sensitivity of a reputation score to each earlier signal. Opponent modeling estimates other agents' action and signal policies, while retaining the known aggregator in the computational graph. We leave settings with unknown or time-varying reputation aggregators to future work.

Q Future Work

The present work studies reputation-mediated credit assignment in a controlled single-learner setting. A next step is to extend the framework to agents that learn simultaneously and investigate when their joint dynamics are stable. This extension would require tracking evolving opponent responses rather than best responding to a fixed population. If this extension proves effective, simulations inspired by e-commerce and digital gig work could be used to examine how strategic behavior changes under different rating rules.

With well-specified and validated platform models, this line of work could eventually support an equilibrium-analysis sandbox. Candidate equilibria could inform counterfactual comparisons, further simulation refinement, and human decision-making. Developing such a sandbox would also require specifying an equilibrium concept and methods for identifying and validating candidate equilibria; stable learning dynamics alone would not establish these properties.

R Reputation-Gradient Ablation and Opponent-Model Baselines

We supplement the main experiments with a controlled reputation-gradient ablation and additional opponent-model baselines. The ablation isolates a backward path while preserving the forward game; the baseline comparisons evaluate alternative learning methods under the protocols specified below. All results concern one learner facing fixed, non-learning opponents.

R.1 Stopping the Reputation-History Gradient

Equation 2 identifies the transit term through reputation, and Appendix C.3 illustrates its role in a minimal L6 example. Here we test the corresponding mechanism over complete training runs in the signal-only and joint settings.

Settings and objective. Both settings contain three agents, with agent 0 as the learner and benefit $b = 10$. In the signal-only setting, the learner trains φ^0 while its action is fixed to the recipient's mean reputation; the opponents are ProudCoop and AllDefector with L3 assessment, and $c = 5$. In the joint setting, the learner trains both π^0 and φ^0 against HybridCoop and AllDefector with L3 assessment, and $c = 1$. The opponent behaviors are described in Section 5, and the continuous assessment rules in Appendix N. Initial reputations are independent uniform draws on $[0, 1]$. Each round randomly orders all six directed pairs; after three mandatory rounds, another round occurs with probability 0.88. Repeated partners are therefore allowed. Training maximizes the learner's episode payoff divided by its number of interactions.

Controlled intervention. The full and stopped branches share the learner architecture, initial parameters, frozen opponent-model bundle, optimizer settings, and random tapes for matching, policy noise, and evaluation. In the stopped branch, every new public assessment is detached immediately before entering its donor's reputation history. If sg denotes this operation, the appended value is $\mathrm{sg}(\sigma_t)$, with the same numerical value as σ_t and derivative zero. Actions, assessments, reputation aggregation, and payoffs are otherwise computed by the same forward equations. At a shared policy and random tape, the forward values coincide; the policies can subsequently diverge because their gradients differ. Immediate donation-cost gradients remain available to the action policy. A costless signal has no remaining reward path, so its parameters receive no update under the stated separation of action and signal parameters and absence of auxiliary losses.

Distinction from policy-gradient baselines. The stopped branch still differentiates the rollout payoff through the remaining computation graph; it adds neither a score-function loss nor a learned critic. By contrast, a REINFORCE-style policy-gradient baseline weights decision log probabilities by the sampled episode payoff, optionally centered by a return baseline. Although the sampled return is detached, the log probabilities remain differentiable, allowing future reputation-mediated payoffs to provide statistical credit to earlier signals without differentiating the reputation dynamics. Its signal-policy update is therefore not generally zero. The exactly zero signal gradient in this control follows from truncating the pathwise estimator under the stated parameter separation and absence of auxiliary losses. This ablation isolates the contribution of RG's reputation-history derivative; it does not replace a comparison with a score-function policy-gradient baseline.

Training and evaluation. We use ten paired seeds, 200 policy updates, and batches of 128 trajectories, giving 25,600 unique training trajectories per branch and seed. Policies use the full-history Transformer architecture, with dropout disabled and logistic-normal latent standard deviation 0.5. The signal learning rate is 10^{-2}; joint action learning uses 10^{-3}, with gradient clipping at norm 1. A type-conditioned opponent-model bundle is fitted once per setting to 80 batches of public interactions using 1,600 fitting steps, then frozen and shared across branches. Model acceptance includes held-out observational errors and known-function grid checks; this controlled validation is therefore oracle-assisted. The primary endpoint executes the learned policy's deterministic median against the true fixed opponents on 64 batches of 128 trajectories. These supplementary protocols are reported separately from Figure 3. Sample and optimizer-update budgets are matched; the full-gradient branch's additional diagnostic passes do not match backward computation or wall-clock cost.

Setting	RG (full)	History stop	RG − stop	Unchanged signal parameters
Signal-only	1.0902 ± 0.0187	0.3695 ± 0.1878	0.7208	10/10
Joint	2.1776 ± 0.0392	1.5878 ± 0.0479	0.5898	10/10

Table 24: Matched reputation-history gradient control. Payoff per learner interaction is evaluated in the true environment using the deterministic-median policy; entries are mean ± population standard deviation over ten paired seeds. The fourth column is the mean paired payoff difference. The last column counts seeds whose final signal parameters exactly equal their initialization.

R.2 Additional Opponent-Model Baselines

We compare RG with adaptations that add opponent models or recurrent opponent representations to a local-credit learner. These comparisons ask whether such additions close the observed return gap under their stated training rules. They complement the matched backward-path ablation above, but do not themselves isolate a single causal change.

LOLA-OM-DDPG. LOLA anticipates an opponent's learning update (Foerster et al., 2018); its relationship to RG is discussed in Appendix A. Because the true opponents here do not learn, their actual opponent-shaping term is zero. Our adapted baseline fits differentiable opponent clones to public decisions and computes a counterfactual lookahead correction while retaining the historical DDPG actor–critic, replay, target networks, and local-credit base update. Clones use 100 pretraining batches and 800 fitting steps, likelihood standard deviation 0.05, and one online fitting step per outer iteration. The virtual step size is 2×10^{-6}. One independently sampled Hessian–vector correction is added to the first actor substep of each outer iteration; the remaining substeps use the base DDPG update. The clone update is hypothetical and never changes the real opponents. Setting the correction coefficient to zero recovers the base DDPG implementation.

LIAM and NAM with local credit. Local Information Agent Modelling (LIAM) learns a recurrent representation through opponent-observation and behavior reconstruction (Papoudakis et al., 2021). Our continuous-action adaptation uses an LSTM with hidden dimension 128 and a 20-dimensional embedding. Control gradients are stopped at the LIAM embedding, and a separate reconstruction loss trains its encoder and decoder. No Agent Modelling (NAM) uses the same recurrent representation without that reconstruction objective; its encoder instead receives the control gradient. Controllers use public decision context and local history, without true opponent parameters, future schedules, or reconstruction targets as policy inputs. These are task-specific adaptations, rather than unmodified implementations of the original LIAM algorithm.

The signal-only LIAM/NAM controllers use deterministic DDPG actors, action-value critics, replay, and target networks. The joint comparison retains a recurrent logistic-normal score-function actor–critic with parameter-disjoint action and signal heads. In the joint case, "DDPG-matched" describes the credit boundary, not the optimizer: action targets contain the current donation cost and bootstrap from the next learner action, while signal targets contain the current recipient benefit and bootstrap from the next learner signal. These targets use discount 0.98 and omit rewards from the other decision pathway. We label the table rows *LIAM local-credit* and *NAM local-credit* to make this distinction explicit. Their comparison changes both representation supervision and encoder-gradient routing, so it is not a single-factor reconstruction-loss ablation.

Signal-only and joint comparisons. The populations, material payoffs, matching rule, and deterministic-median true-environment evaluation follow the corresponding settings above. Initial reputations are uniform except in the retained signal-only plain-DDPG cell, which starts all three reputations at 0.5; that cell is not a matched-initialization comparison. DDPG and LOLA-OM-DDPG use 125 outer iterations, five rollout batches per iteration, and 32 replay update loops per iteration. Their actor and critic learning rates are 3×10^{-5} and 10^{-3}; replay capacity is 10,000, minibatch size 256, exploration standard deviation 0.1, and target-update coefficient 0.005. Signal-only LIAM/NAM use this DDPG budget, with reconstruction learning rate 7×10^{-4} for LIAM. Joint LIAM/NAM use 200 batches of 128 episodes, latent standard deviation 0.5, and actor–critic and reconstruction base learning rates 3×10^{-4} and 7×10^{-4}, with one common learning-rate scale chosen on separate

calibration seeds. Joint LIAM/NAM treat the frozen opponent model as a black-box training environment; evaluation uses the true fixed opponents. All cells report ten training seeds. Architectures, objectives, initialization parameters, and optimizer budgets are not paired across method families.

Method	Signal-only ProudCoop + AllD	Joint HybridCoop + AllD
RG	1.0902 ± 0.0187	2.1776 ± 0.0392
DDPG	0.66 ± 0.06	1.55 ± 0.11
LOLA-OM-DDPG	0.6393 ± 0.0408	1.5659 ± 0.0726
LIAM local-credit†	0.5530 ± 0.0087	1.579 ± 0.055
NAM local-credit†	0.5529 ± 0.0087	1.584 ± 0.059

Table 25: Additional opponent-model baselines in the three-agent signal-only and joint settings. Payoff per learner interaction is measured in the true environment with deterministic-median decisions. Every cell uses ten training seeds and reports mean $\pm$ population standard deviation. † LIAM/NAM use deterministic DDPG controllers in signal-only and the DDPG-credit-matched recurrent score-function controllers described in the text in joint. Signal-only plain DDPG starts all reputations at 0.5; other cells use uniform starts. Methods differ in objectives, architectures, budgets, and initialization.

Interpretation. In the signal-only and joint comparisons, the added opponent-model and representation baselines do not close the observed payoff gap to RG. LIAM and NAM have close point estimates in these cells, without a consistent directional advantage; this does not establish statistical equivalence. More generally, these local-credit adaptations do not establish how unmodified full-return LOLA, LIAM, or other policy-gradient estimators would perform. The matched reputation-history control in Appendix R.1 provides the direct evidence for the contribution of the removed backward path.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and the introduction state two contributions: (i) the influence-attribution problem (Section 4.1), (ii) the reciprocity gradient with differentiable opponent modeling (Sections 4, 5).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: A dedicated Limitations section (Appendix P) discusses single-agent scope, population size, statistical power, the public-assessment assumption, reputation memory window, identity-based observation, and opponent-modeling order coverage.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The continuous random-matching donation game is formally defined in Definition 3.1; the reciprocity-gradient derivation is presented step-by-step in Section 4 and expanded with full chain-rule expansion in Appendix E. Assumptions (public assessment, full reputation history, learner observability) are stated explicitly in Section 3 and revisited in the Limitations.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The full algorithm is given in Appendix F.1; opponent populations (Leading Eight L3/L6, ProudCoop, AllDefector, HybridCoop) are precisely specified in Appendix B; all hyperparameters (learning rates, T_{outer}, N_{train}, exploration noise, batch sizes) are listed in the experimental sections and Appendix J.1; baseline configurations (DPG/DDPG/TD3) match published reference implementations.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Source code is not released as supplementary material with this submission. We will release the code under an open-source license once the codebase and accompanying reproduction materials have been organized for public release. At submission time the algorithmic content is fully specified in Appendix F.1 (pseudocode), Section 5 (per-experiment hyperparameters), and Appendix J.1 (compute and infrastructure).

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Optimizer (Adam), learning rates per pathway (3×10^{-5} for π, 3×10^{-3} for φ, 5×10^{-3} for TD3 actor at the default and the LR-matched setting), T_{outer}=125, N_{train}, exploration noise std, batch size, surrogate-pretraining transition count, and seed counts are all reported in Section 5 and Appendix J.1; the choice rationale (LR matching across pathways, cost c=5 for Section J.3) is documented inline.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: All quantitative claims report mean $\pm$ standard deviation across seeds (e.g. 2.22 ± 0.03 on the joint flagship). Per-seed counts (5–10 depending on setting) and the convention that $\pm$ values are seed-level standard deviations rather than confidence intervals are explicitly documented in the *Statistical power* paragraph of the Limitations section.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Experiments were run on the CPU of a single Apple M4 Pro machine, without GPU acceleration or distributed training. The overall compute budget and per-experiment runtimes are described in Appendix J.1.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

Justification: The work is purely simulation-based with synthetic agents in an abstract donation game; it involves no human subjects, no scraped data, and no released model with deployment risk. The authors have reviewed the NeurIPS Code of Ethics and found no point of conflict.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [N/A]

Justification: The paper studies the foundational mathematical structure of analytic gradients in reputation-mediated multi-agent games. It has no direct path to deployed systems, no released models, and no datasets. Any downstream application would require substantial additional engineering well outside the scope of this work, so we judge that an explicit positive/negative impact analysis would be speculative beyond what is appropriate for a foundational simulation paper.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: No pretrained language models, image generators, or scraped datasets are released. The simulation framework implements an abstract donation game and poses no plausible misuse risk.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All algorithmic baselines (DPG (Silver et al., 2014), DDPG (Lillicrap et al., 2015), TD3 (Fujimoto et al., 2018)) and conceptual frameworks (Leading-Eight norms (Ohtsuki & Iwasa, 2004, 2006), differentiable opponent modeling (Foerster et al., 2018; Letcher et al., 2019; Willi et al., 2022)) are properly cited at first use. Software dependencies (PyTorch, NumPy, Matplotlib) are open-source under permissive licenses respected by our implementation.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: No new assets (code, data, or models) are released with this submission.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or human-subject research. All experiments are entirely simulation-based.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve human subjects, so IRB review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: The core methodology (reciprocity gradient, differentiable opponent modeling, Leading-Eight environments, deterministic actor-critic baselines) involves no LLM as an important, original, or non-standard component. LLMs were used only as general writing assistants, which the NeurIPS LLM policy does not require to be declared.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.