WHAT TO REMEMBER FOR COOPERATION: THE CONTENT AND TIMESCALE OF MEMORY IN SEQUENTIAL SOCIAL DILEMMAS

Anonymous authors
Paper under double-blind review

ABSTRACT

In shared environments, self-interested individuals repeatedly pursue their own objectives, yet locally rational choices can erode cooperation and reduce collective welfare. Sequential social dilemmas (SSDs) provide a formal setting for studying this tension. Existing approaches in SSDs often promote cooperation by modifying incentives or modeling opponents' learning. A more basic question is whether and under what conditions interaction history itself can support learned cooperation. Here, we augment agents' current observations with compact summaries of past interactions and systematically examine how memory representation and horizon shape conditional cooperation. Theoretically, we establish sufficient conditions under which a memory-based strategy achieves a strict local return advantage over any memoryless decision rule. Experiments in Coin Game and CleanUp-lite show that memory representations that link outcomes to the agents that produced them improve cooperation. Extending the memory horizon can further improve cooperation, whereas shuffling events within longer memory sequences weakens it.

1 INTRODUCTION

Cooperation is a central challenge in multi-agent decision making (Busoniu et al., 2008; Du et al., 2023). In shared environments, actions that advance an agent's immediate objective can impose delayed costs on others and ultimately undermine collective welfare (Hardin, 1968). This tension arises in domains ranging from autonomous driving, where aggressive merging can disrupt traffic flow (Schwarting et al., 2019; Liu et al., 2022), to renewable-resource management, where short-term overexploitation threatens long-term sustainability (Clark, 1973; Hilborn et al., 2020). Sequential social dilemmas (SSDs) formalize such conflicts as temporally extended Markov games, providing a natural setting for studying the emergence of cooperation in multi-agent reinforcement learning (MARL) (Leibo et al., 2017; Gronauer & Diepold, 2022).

Two prominent approaches promote cooperation by modifying agents' incentives or explicitly influencing the learning dynamics of their partners. *Reward shaping* methods incorporate intrinsic or extrinsic incentives—such as social preferences, gifting, or punishment—into agents' learning objectives (Hughes et al., 2018; Lupu & Precup, 2020; Schmid et al., 2021). *Opponent shaping* models partners as learning agents whose future updates depend on the learner's current behavior. By using this dependence to shape partners' future learning, the learner can build cooperation when induced updates improve returns for both sides (Foerster et al., 2018; Lu et al., 2022; Aghajohari et al., 2024). Although effective, these approaches often introduce assumptions beyond standard decentralized learning: reward-shaping methods introduce hand-crafted reward terms or explicit channels to reward and punish others, while opponent-shaping methods can require access to private information such as partners' gradients or update rules. In summary, these approaches rely on auxiliary mechanisms to promote cooperation, but leave open a more basic question: in SSDs, can cooperation emerge from interaction history itself?

Repeated matrix games provide a natural foundation for examining this question. Their simple and theoretically transparent structure isolates the tension between individual incentives and collective cooperation. Within this setting, canonical strategies such as Tit-for-Tat (TFT) and Win-Stay, Lose-Shift (WSLS) can sustain reciprocity by conditioning current behavior on previous actions or

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

outcomes (Axelrod & Hamilton, 1981; Nowak & Sigmund, 1993). Related studies similarly show that learning agents with access to finite memory can support the acquisition of reciprocal behavior in repeated games (Bertrand et al., 2025; Su et al., 2025).

These results suggest that memory can support learned reciprocity, however, extending this idea from repeated matrix games to SSDs is nontrivial. In repeated matrix games, actions have explicit cooperative or defective meanings. In SSDs, by contrast, primitive actions such as moving, collecting, cleaning, or attacking have no fixed social interpretation (Leibo et al., 2017). Their significance depends on context and may become apparent only through behavioral patterns that unfold over multiple time steps. The central challenge is therefore not simply whether agents remember, but what information they retain and over what timescale. We study these two dimensions through memory representation, which determines what social information is stored, and memory horizon, which determines how much temporal context informs history-dependent decisions.

The primary contributions of this work are as follows. (1) We introduce Memory-Augmented Mixed-Partner Training (MAMPT), which combines memory-conditioned policies with training against both co-learning partners and fixed mentors, while leaving the learners' environment rewards and optimization objectives unchanged. (2) Theoretically, we show that a memory-conditioned strategy can strictly outperform memoryless policies when distinct interaction histories yield the same current observation. (3) Across SSDs, we show that effective memory depends on actor–outcome association, horizon, and temporal order: short histories suffice in compact games, whereas longer histories help when socially relevant behavior unfolds over extended timescales.

2 Related Work

2.1 Incentives for Cooperation

Cooperation in SSDs is often promoted by modifying agents' incentives. Intrinsic-motivation methods do so by reshaping the agent's own objective. They add social preferences or auxiliary rewards that make agents value fairness, prosocial outcomes, or the influence of their actions on others. Representative examples include inequity aversion (Hughes et al., 2018), prosocial reward shaping (Peysakhovich & Lerer, 2018b), social value orientation (McKee et al., 2020), social influence (Jaques et al., 2019), and reputation (McKee et al., 2021). By changing what agents optimize, these methods can reduce purely self-interested behavior and encourage cooperative outcomes.

Beyond intrinsic motivation, other incentive-based methods introduce external mechanisms through which agents can affect one another's payoffs. Instead of only changing an agent's internal objective, these methods allow agents to reward (Hostallero et al., 2020; Yang et al., 2020; Aghajohari et al., 2024), gift (Lupu & Precup, 2020), or to penalize and sanction undesirable behavior during interaction (Schmid et al., 2021; Yaman et al., 2023; Vinitsky et al., 2023). Such mechanisms can discourage free-riding, support reciprocity, and stabilize cooperative behavior. Together, these methods promote cooperation by modifying agents' objectives or introducing explicit reward and sanction mechanisms. Despite their effectiveness, this line of work has paid relatively limited attention to the role of interaction memory in supporting cooperation.

2.2 Opponent-Aware Cooperation

Opponent Shaping. In MARL, each agent interacts with opponents whose policies may continue to change during training. Opponent shaping methods explicitly account for this dependence by considering how an agent's current behavior can affect an opponent's future updates. A canonical example is Learning with Opponent-Learning Awareness (LOLA) (Foerster et al., 2018), which differentiates through a predicted opponent update. Subsequent work improves the stability, parameterization invariance, and consistency of such learning-aware updates (Letcher et al., 2019; Zhao et al., 2022; Willi et al., 2022). These gradient-based approaches, however, often require access to the opponent's objective, gradients, or update rule and may involve costly higher-order derivatives. Model-Free Opponent Shaping (MFOS) (Lu et al., 2022) removes this white-box requirement by formulating opponent shaping as model-free meta-learning over repeated interactions, although optimizing the resulting meta-game remains sample-intensive.

More recent work pursues lighter-weight forms of learning-aware interaction. Learning with Opponent Q-Learning Awareness (LOQA) (Aghajohari et al., 2024) models opponents as Q-learners and favors actions based on their predicted effects on the opponent's future Q-values. Reciprocal Reward Influence (Reciprocator) (Zhou et al., 2024) instead estimates how an opponent's action affects the learner relative to a counterfactual baseline and incorporates this influence as an intrinsic reciprocal reward. Across these variants, opponent shaping promotes cooperation by modeling or influencing how opponents adapt. However, this line of work typically takes agent memory as given, using it to model opponents rather than studying memory as a mechanism for cooperation in its own right.

Adaptive Policy Selection. Methods in this category typically follow a two-stage procedure: pretraining a set of policies with distinct behavioral styles and then selecting among them online based on observed partner behavior or interaction outcomes. Within SSDs, approximate Markov Tit-for-Tat (amTFT) (Lerer & Peysakhovich, 2017) and Consequentialist Conditional Cooperation (CCC) (Peysakhovich & Lerer, 2018a) enable agents to respond differently to cooperative and defecting partners through history-dependent reciprocity. Both implement this adaptation by switching between separately pretrained cooperative and defecting policies according to hand-designed decision rules. By contrast, our approach takes a different route: we investigate the autonomous emergence of partner-contingent cooperation within a single memory-conditioned policy and study how memory representation, horizon, and temporal order shape this behavior.

2.3 Memory-Conditioned Reciprocity

Classical research on repeated matrix games has long recognized memory as a central mechanism for sustaining cooperation. In the iterated Prisoner's Dilemma, for example, history-dependent strategies such as Tit-for-Tat support reciprocal cooperation by conditioning behavior on the opponent's previous actions (Axelrod & Hamilton, 1981). Repeated-game theory generalizes this idea, showing that sufficiently patient agents can maintain cooperation through history-dependent rewards and punishments (Fudenberg & Maskin, 1986). More specifically, analyses of memory-one and memory-n strategies show that policies conditioned on a finite window of past actions can still produce rich long-term payoff structures in repeated games (Press & Dyson, 2012; Hilbe et al., 2017). Building on this finite memory view, recent work asks whether reciprocal strategies can be learned rather than specified by hand. Self-play Q-learners have been shown to sustain cooperation in the iterated Prisoner's Dilemma (Bertrand et al., 2025), while finite-memory MARL can uncover dominant reciprocal strategies in iterated and evolutionary games (Su et al., 2025). These studies show that finite memory can support learned reciprocity in iterated matrix games, but its effectiveness in SSDs remains unclear.

3 Preliminaries

We model sequential social dilemmas as N-player general-sum Markov games, also known as stochastic games (Shapley, 1953; Hansen et al., 2004). Formally, each game is represented as:

$$\mathcal{M} = \left\langle \mathcal{I}, \mathcal{S}, \{\mathcal{A}_i\}_{i=1}^N, \{\mathcal{O}_i\}_{i=1}^N, \mathcal{P}, \mathcal{E}, \{\mathcal{R}_i\}_{i=1}^N, \gamma \right\rangle,$$

where $\mathcal{I}$ denotes the set of agents, and $s \in \mathcal{S}$ denotes the global state of the environment. Since the game is partially observable, agent i does not observe s directly, but instead receives a local observation $o_i \in \mathcal{O}_i$ sampled from the emission function $\mathcal{E}(\cdot \mid s)$. Given its local information, agent i selects an action $a_i \in \mathcal{A}_i$ according to its policy. The individual actions form a joint action $\mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$. The environment then transitions to s' according to $\mathcal{P} : \mathcal{S} \times A \to \Delta(\mathcal{S})$, and agent i receives reward $\mathcal{R}_i : \mathcal{S} \times A \to \mathbb{R}$. Due to partial observability, each agent follows an observation-conditioned policy $\pi_i : \mathcal{O}_i \to \Delta(\mathcal{A}_i)$, written as $\pi_i(a_i \mid o_i)$. We denote the joint policy by $\pi = (\pi_1, \ldots, \pi_N)$. For discounted returns with factor γ, the state value of agent i under π is defined as:

$$V_i^\pi(s_0) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}_i(s_t, \mathbf{a}_t) \;\middle|\; \mathbf{a}_t \sim \pi_t, s_{t+1} \sim \mathcal{P}(s_t, \mathbf{a}_t)\right].$$

4 MEMORY-AUGMENTED MIXED-PARTNER TRAINING

We introduce **M**emory-**A**ugmented **M**ixed-**P**artner **T**raining (MAMPT), which combines *(i) memory-augmented policies* conditioned on an ordered history of recent interaction events and *(ii) mixed-partner training* with co-learning agents and fixed mentors. The former provides temporal context for history-dependent decisions, while the latter exposes learners to adaptive and stationary partner behaviors. We also derive sufficient conditions for beneficial memory-conditioned cooperation.

4.1 MEMORY-AUGMENTED POLICIES

Policy formulation. In repeated interactions, histories that yield the same current observation may nevertheless warrant different responses. A memoryless policy $\pi_\theta^0(a_t \mid o_t)$ cannot distinguish such histories because it conditions only on o_t. To isolate the role of interaction memory in emergent cooperation, we leave learning objectives unchanged and augment the policy input with the k most recent interaction events in temporal order:

$$h_t = \left(m_t^{(1)}, \ldots, m_t^{(k)}\right), \qquad a_t \sim \pi_\theta^k(\cdot \mid o_t, h_t),$$

where $m_t^{(i)}$ denotes the encoded representation of the i-th most recent completed interaction event observed before the agent selects a_t, with $i = 1$ corresponding to the most recent event. If fewer than k events are available, the remaining positions are padded with a designated null entry. By ignoring h_t, the memory-conditioned policy class can represent every memoryless policy. Section 4.3 provides sufficient conditions under which a memory-conditioned response achieves a strict return advantage over memoryless decision rules.

Interaction memory representations. Whether the policy can exploit this additional input, however, depends on what social information each memory entry retains. In SSDs, events such as resource collection, pollution removal, or attacks have context-dependent social meanings rather than fixed interpretations as cooperation or defection. Inspired by event representations that distinguish an event's actor from its content or outcome (Hoppler et al., 2022; Gildea & Jurafsky, 2002), we define an *actor–outcome* representation in which each event is described by two components: its actor $c_t^{(i)}$ (who acted) and its outcome $y_t^{(i)}$ (what occurred). The i-th memory entry is encoded as:

$$m_t^{(i)} = \phi\left(c_t^{(i)}, y_t^{(i)}\right),$$

where ϕ determines whether neither component, either component alone, or both components are retained. Accordingly, interaction memory representations fall into three categories: (i) *non-interactive* representations retain neither component; (ii) *partially interactive* representations retain exactly one component; and (iii) *jointly interactive* representations retain both components. This taxonomy allows us to examine whether and how actor identity, event outcome, and their joint availability inform memory-conditioned cooperation.

4.2 MIXED-PARTNER TRAINING WITH FIXED MENTORS

To train our memory-augmented agents, we use population-based self-play with independent policy updates (Strouse et al., 2021; Jaderberg et al., 2019). Learners in a population $\mathcal{A}$ generate experience through mutual interaction. Because co-learning partners are updated throughout training, each learner faces a changing partner distribution that may provide limited exposure to behaviors outside the population's current repertoire. We therefore augment $\mathcal{A}$ with a pool $\mathcal{M}$ of fixed rule-based partners, termed *mentors* (Su et al., 2025; Lanctot et al., 2017). By exhibiting cooperative, defensive, and exploitative behaviors, these mentors serve as stationary reference partners and diversify the interaction histories encountered during training.

Training protocol. Each episode begins by uniformly sampling a focal learner $i \in \mathcal{A}$. We then sample its partner j uniformly from $\mathcal{A} \setminus \{i\}$ with probability p_{agent} and from $\mathcal{M}$ with probability $1 - p_{\text{agent}}$. Setting $p_{\text{agent}} = 1$ recovers self-play, whereas smaller values increase the frequency of mentor interactions. If $j \in \mathcal{A}$, both learners collect experience and update their policies independently; if $j \in \mathcal{M}$, only the focal learner updates its policy, while the mentor's policy remains fixed.

This partner assignment determines which learners collect experience and receive policy updates. At each step t, each such learner conditions its policy on the augmented input $x_t = (o_t, h_t)$, where o_t is its current observation and h_t summarizes its recent interaction history. After the joint action, it updates its history to h_{t+1} and stores the transition (x_t, a_t, r_t, x_{t+1}) in its replay buffer. At the end of the episode, these learners independently update their DQN parameters using replayed transitions (Mnih et al., 2015). Appendix B provides the complete training procedure.

Mentor construction. The training protocol above relies on a fixed mentor pool $\mathcal{M}$, which we construct from classical repeated-game strategies (e.g., ALLD, TFT, WSLS, and Grim). To instantiate these strategies in SSDs, each mentor (i) maps an interaction trajectory to a high-level behavior label using environment-specific grounding rules, (ii) applies its decision rule to the resulting label history to select a high-level response, and (iii) realizes that response through the corresponding low-level policy, π_C or π_D, following the modular design of amTFT (Lerer & Peysakhovich, 2017). Appendices C provides the strategy definitions, grounding rules, and policy details.

4.3 Existence of Beneficial Memory-Conditioned Cooperation

We use a simple theoretical analysis to clarify when interaction memory can support cooperation. When the current observation aliases histories that call for different responses, we identify conditions under which a memory-conditioned reciprocal strategy outperforms memoryless decision rules.

Consider an observation o that can arise after either of two history contexts, h_C and h_D, representing cooperative and defective interaction histories, respectively. Each context may summarize multiple concrete histories. Conditional on o, let:

$$\Pr(h = h_C \mid o) = p, \qquad \Pr(h = h_D \mid o) = 1 - p, \qquad p \in (0, 1).$$

In this abstraction, C and D label two available responses. The memory-conditioned reciprocal rule selects C after h_C and D after h_D, formally written as:

$$\pi_{\text{mem}}(C \mid o, h_C) = 1, \qquad \pi_{\text{mem}}(D \mid o, h_D) = 1.$$

A memoryless rule conditions on o and uses the same distribution in both contexts. Let $q \in [0, 1]$ denote its probability of selecting C, and $1 - q$ its probability of selecting D. Let $Q(h, a)$ denote the expected discounted return from taking response $a \in \{C, D\}$ in history context h, and then following the common continuation rule specified below. The memory-conditioned rule achieves:

$$U_{\text{mem}} = pQ(h_C, C) + (1 - p)Q(h_D, D),$$

whereas the memoryless rule yields:

$$\begin{aligned}\bar{U}_{\text{memless}}(q) = p\big[qQ(h_C, C) + (1 - q)Q(h_C, D)\big] \\ + (1 - p)\big[qQ(h_D, C) + (1 - q)Q(h_D, D)\big].\end{aligned}$$

The return difference is:

$$\begin{aligned}\Delta(q) &:= U_{\text{mem}} - \bar{U}_{\text{memless}}(q) \\ &= p(1 - q)\big[Q(h_C, C) - Q(h_C, D)\big] \\ &\quad + (1 - p)q\big[Q(h_D, D) - Q(h_D, C)\big].\end{aligned}$$

Let h_C and h_D identify states S_C and S_D, respectively. In S_C, choosing C yields r_C and preserves the state, whereas choosing D yields $r_C + g$ immediately and moves to S_D. In S_D, both responses preserve the state: D yields r_D, whereas C yields $r_D - \ell$. We assume $r_C > r_D$ and $g, \ell > 0$. After the initial response, we evaluate both choices under the same state-dependent continuation rule, which selects C in S_C and D in S_D. The reached state then persists and yields r_C or r_D per step.

Proposition 1 (Local Memory Advantage)**.** *Let $\gamma \in [0, 1)$ be the discount factor and $p \in (0, 1)$. Under the dynamics above, if $\gamma > g/(g + r_C - r_D)$, then $\Delta(q) > 0$ for every $q \in [0, 1]$.*

The proof is provided in Appendix A. The discount condition makes C preferable to the immediate deviation D in S_C. In S_D, response D avoids the immediate loss ℓ, while both responses have the same continuation value. Thus, the reciprocal rule selects the higher-value response in each context. The following experiments examine whether agents learn such history-dependent responses.

Table 1: Interaction-memory representations evaluated in the Coin Game. The Horizon column uses Event for a one-event window ($k = 1$) and Episode for a recurrent state spanning the full episode. Values are reported as mean (SD) over 5 matched seeds.

Category	Representation	Actor	Outcome	Horizon	Avg. reward	Coop. p_{same}
Non-interactive	Memoryless	✗	✗	–	-4.27 (0.33)	0.472 (0.002)
	Null-content	✗	✗	–	-4.10 (0.26)	0.471 (0.002)
Partially interactive	Actor-only	✓	✗	Event	8.50 (0.28)	0.569 (0.002)
	Outcome-only	✗	✓	Event	-6.42 (0.28)	0.455 (0.002)
Jointly interactive	Actor-outcome	✓	✓	**Event**	**29.04 (0.26)**	**0.812 (0.007)**
	C/D label	✓	✓	Event	15.84 (1.93)	0.637 (0.015)
	GRU-DQN	✓	✓	Episode	12.93 (0.34)	0.654 (0.005)

5 Experiments

We design our empirical evaluation to determine when and how interaction memory supports cooperation in SSDs. Because primitive actions have no fixed interpretation as cooperation or defection, we first treat memory as a representation problem and ask which components of past interactions provide useful social context. We then treat it as a temporal problem, examining how memory horizon and event order determine whether retained histories capture behaviorally meaningful structure. We benchmark the resulting agents against representative baselines and held-out partners, and finally test whether the benefits of memory extend from the Coin Game to CleanUp-lite, where cooperation unfolds over longer resource-management timescales.

5.1 Memory representations shape cooperative behavior

The Coin Game (Lerer & Peysakhovich, 2017) is a two-player spatial social dilemma in which agents collect coins in a shared environment. Collecting own coins leaves the partner unaffected, whereas collecting other coins harms the partner (see Appendix D.1.1 for details). We report two complementary metrics: *Avg. reward*, the mean reward across both agents, captures episode-level social welfare, whereas p_{same}, the fraction of coin pickups in which the coin matches the collector's color, captures cooperation (Lerer & Peysakhovich, 2017; Zhou et al., 2024). Higher values indicate better collective outcomes and fewer partner-harming pickups, respectively.

Following the actor–outcome representation introduced in Section 4.1, we instantiate it in the Coin Game at the level of coin-pickup events. In this environment, the actor is the player who collects the coin, whereas the outcome records whether the pickup is own-color or other-color. Together, these components identify who produced a partner-neutral or partner-harming event. We evaluate three categories: *(i) non-interactive* baselines that provide no event information (Memoryless and Null-content); *(ii) partially interactive* representations that retain only the actor or outcome (Actor-only and Outcome-only); and *(iii) jointly interactive* representations that associate the actor with either the pickup outcome (Actor-outcome) or a derived cooperation/defection label (C/D label). To isolate representation effects, we fix the horizon at $k = 1$ for all explicit event-based variants. We also include GRU-DQN as a recurrent baseline whose hidden state summarizes the episode history up to the current step. Further implementation details are provided in Appendix D.4.1.

Table 1 shows a clear pattern among the controlled $k = 1$ variants: both jointly interactive representations achieve higher mean social welfare and cooperation than all non-interactive and partially interactive variants. Actor-outcome performs best overall, increasing the mean episode return from 8.50 to 29.04 and p_{same} from 0.569 to 0.812 relative to Actor-only. It also achieves higher mean values than the recurrent GRU-DQN baseline on both metrics. The weaker performance of Actor-only and Outcome-only suggests that neither component alone is as informative as preserving their association. This ordering is consistent with complementary roles for the two components: actor information supports attribution to a particular player, whereas outcome information characterizes the event's social consequence.

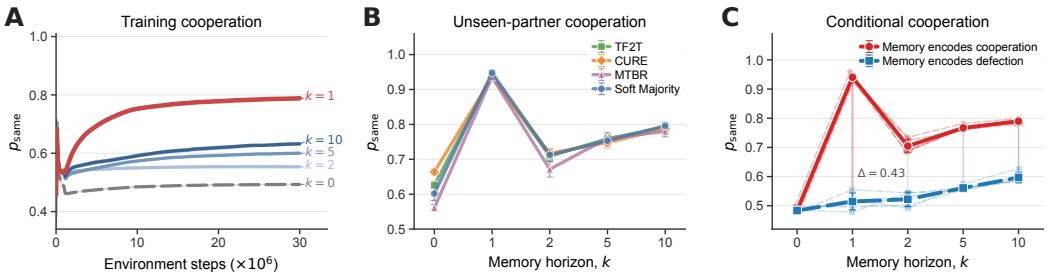


Figure 1: Memory horizon shapes conditional cooperation in the Coin Game. (A) Cooperation during mixed-partner training, measured by p_{same}, varies with memory horizon. (B) Held-out evaluation shows this behavior transfers to unseen rule-based partners. (C) Agents are conditionally cooperative, sustaining it after partner cooperation and reducing it after defection. Lines denote means over 5 matched seeds; error bars denote standard deviations.

5.2 Memory Horizon and Event Order Modulate Cooperation

Having established that what interaction memory retains matters for cooperation, we next ask how much history it should preserve. A single recent event may provide an immediate cue about the partner's behavior, whereas a longer history may place individual events in the context of multiple exchanges. It is therefore unclear whether cooperation improves monotonically as the memory horizon increases. To isolate the effect of memory horizon, we hold the *Actor–outcome* representation fixed and vary $k \in \{0, 1, 2, 5, 10\}$. For $k > 0$, the policy receives the k most recent completed coin-pickup events, ordered from newest to oldest; $k = 0$ denotes the memoryless baseline.

Cooperation varies non-monotonically with memory horizon. At the end of training, all memory-augmented variants attain higher mean cooperation than the memoryless baseline, but cooperation varies non-monotonically with memory horizon. Mean p_{same} peaks at $k = 1$ (≈ 0.8), drops at $k = 2$, and partially recovers at $k = 5$ and $k = 10$ (Figure 1A; see Figure 6 in Appendix D.4.3 for results disaggregated by training partner). To further assess the robustness of this pattern, we evaluate the learned policies against four reciprocal partners held out from training—TF2T, CURE, MTBR, and Soft Majority (Axelrod, 1980; Li et al., 2022; Su et al., 2025). The same qualitative ordering persists across all four partners (Figure 1B), providing further evidence that the horizon dependence is not specific to the partner strategies encountered during training.

Learned policies exhibit memory-conditioned reciprocity. High aggregate cooperation alone does not establish reciprocity: agents should cooperate more following partner cooperation than following partner defection. To examine whether the learned policies exhibit this pattern, we perform a paired memory intervention. For each evaluated environment state, we hold all non-memory inputs fixed while varying whether the remembered partner behavior indicates cooperation or defection. We measure the response gap $\Delta(k)$ as p_{same} under remembered partner cooperation minus that under remembered partner defection. At $k = 0$, the policy cannot distinguish the two intervention conditions, yielding a near-zero response gap. For every $k > 0$, the mean gap is positive and peaks at $k = 1$ ($\Delta = 0.43$; Figure 1C). These results support memory-conditioned reciprocity: even under the same current observation, agents cooperate more when they remember partner cooperation and less when they remember partner defection.

Multi-event memory benefits from chronological order. The preceding results show that cooperation varies with the number of retained events, but they do not reveal whether temporal order contributes beyond event content. To isolate the contribution of order, we compare Chronological and Shuffled variants containing the same k most recent coin-pickup events: the former preserves their newest-to-oldest order, whereas the latter randomly permutes them at each memory update. We evaluate $k \in \{2, 5, 10\}$, omitting $k = 1$, for which shuffling has no effect. Across these horizons, shuffling produces progressively larger reductions in mean p_{same} relative to the Chronological variant: approximately 3, 11, and 14 percentage points, respectively (Table 2). Since the two variants retain

Table 2: Effect of shuffling interaction histories on cooperation in the Coin Game. Values are mean p_{same} at convergence over 5 matched seeds. Mean equally weights self-play and five fixed partners; parentheses report differences from chronological memory at the same horizon.

Horizon	Condition	Self-play	TFT	GTFT	OmegaTFT	Hold-a-Grudge	ALLD	Mean
0	–	0.62	0.59	0.63	0.46	0.40	0.39	0.52
2	Chronological	0.77	0.70	0.73	0.55	0.43	0.40	0.59
	Shuffled	0.69 (−0.08)	0.67 (−0.03)	0.69 (−0.04)	0.52 (−0.03)	0.41 (−0.02)	0.40 (0.00)	**0.56 (−0.03)**
5	Chronological	0.77	0.74	0.75	0.67	0.46	0.40	0.63
	Shuffled	0.65 (−0.12)	0.60 (−0.14)	0.63 (−0.12)	0.48 (−0.19)	0.40 (−0.06)	0.39 (0.00)	**0.53 (−0.11)**
10	Chronological	0.77	0.75	0.76	0.71	0.49	0.39	0.65
	Shuffled	0.61 (−0.16)	0.57 (−0.18)	0.61 (−0.15)	0.46 (−0.25)	0.40 (−0.09)	0.39 (0.00)	**0.51 (−0.14)**

exactly the same events, the growing loss under shuffling cannot be explained by event content alone, suggesting that preserving chronological order becomes increasingly important as the memory horizon grows. The complete training curves are provided in Appendix D.4.4.

5.3 Baselines Comparison and Ablations

We next evaluate MAMPT against four representative baselines and use a 2×2 ablation to isolate the contributions of interaction memory and fixed mentors. The baselines span independent learning (**NL**, a memoryless DQN), opponent-aware learning (**LOLA** (Foerster et al., 2018) and **LOQA** (Aghajohari et al., 2024)), and reciprocity-based reward shaping (**Reciprocator** (Zhou et al., 2024)). The no-mentor ablation retains the same $k = 1$ actor–outcome memory and learning objective as MAMPT but replaces mixed-partner training with population self-play. At each checkpoint, we hold the policies fixed and evaluate same-method agent pairs without further learning.

Baseline comparison. Figure 2A shows that MAMPT rapidly reaches and consistently maintains higher mean cooperation than all four baselines. Among the baselines, LOLA eventually attains the highest cooperation level, but only after a marked delay and with greater variability across seeds. The no-mentor variant also substantially outperforms NL, showing that interaction memory is beneficial under population self-play. Figure 2B further shows how this memory-induced gain manifests behaviorally: relative to NL, the no-mentor variant reduces harmful other-color pickups while increasing own-color pickups. Finally, MAMPT and its no-mentor variant both maintain high return fairness and compare favorably with the baselines (Figure 2C), indicating that their cooperation gains do not come at the expense of balanced returns.

Component ablations. As shown in Figure 2A, the no-mentor variant achieves lower cooperation than full MAMPT, indicating an additional benefit from mentor exposure. The factorial ablation in Appendix D.4.5 further disentangles the effects of interaction memory and training composition. Adding memory increases the conditional-response gap from 8.8 to 14.2 percentage points under population self-play without mentors, and from 14.6 to 43.0 percentage points under mixed-partner training with mentors. The substantially larger increase under mixed-partner training supports the motivation for using mentors. This finding suggests that fixed behavioral references help agents use interaction histories to learn memory-conditioned reciprocity.

5.4 Generalization Across Sequential Social Dilemmas

To test whether memory benefits extend beyond Coin Game, we further evaluate agents in CleanUp-lite (Meulemans et al., 2025), a structurally distinct two-player SSD with tragedy-of-the-commons incentives, derived from Cleanup (Agapiou et al., 2022). Harvesting yields immediate private rewards, whereas cleaning offers no immediate reward but reduces pollution and sustains resource regeneration. Agents may also zap one another, with zap rate serving as a behavioral measure of direct conflict. We encode the k most recent successful cleaning or harvesting events as actor–outcome pairs.

Late in training, mean social welfare ranges from 46 to 49 across variants with $k \in \{5, 10, 20\}$, compared with 33–38 for the memoryless and one-event variants ($k \in \{0, 1\}$; Figure 3A). These welfare gains coincide with lower pollution (Figure 3B), reflecting more effective maintenance of the

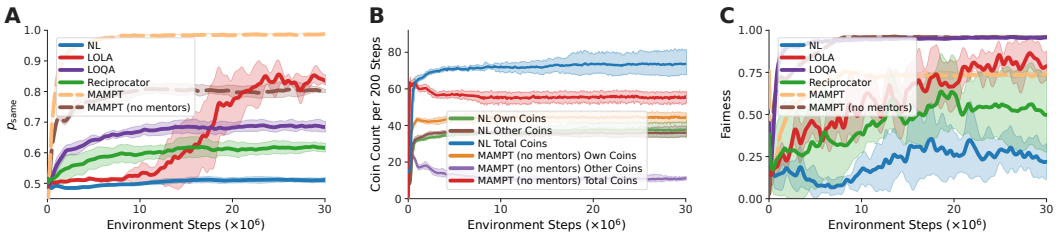


Figure 2: Performance comparison of MAMPT, its no-mentor ablation, and four representative baselines in the Coin Game. (A) Cooperation (p_{same}). (B) Coin pickups per 200 steps. (C) Fairness, with higher values indicating more balanced returns between paired agents. Curves and shaded regions denote means and standard deviations, respectively, over 5 matched seeds.

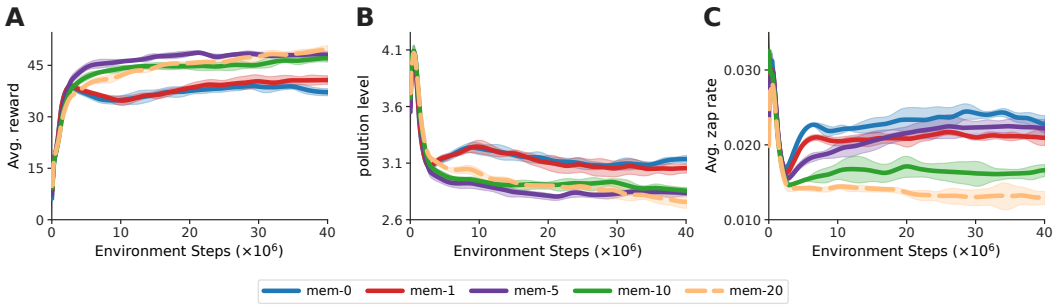


Figure 3: Longer memory improves collective outcomes in CleanUp-lite. (A) Episode-level social welfare. (B) Average pollution level, where lower values indicate better maintenance of the shared resource. (C) Average zap rate, where lower values indicate less direct conflict. Shaded regions denote the standard deviation over 5 matched seeds.

shared resource. This horizon-dependent pattern is consistent with the temporally extended nature of resource management in CleanUp-lite: a single cleaning or harvesting event provides limited evidence about an agent's strategy, whereas longer histories offer richer context for inferring how it balances shared-resource maintenance against private consumption over time. Variants with $k \in \{10, 20\}$ also exhibit lower zap rates than the shorter-horizon variants (Figure 3C), consistent with reduced direct conflict. Together, these results support the generalization of memory benefits across SSDs and are consistent with the hypothesis that the useful memory horizon depends on the temporal scale of socially relevant behavior. The qualitative advantage of longer interaction histories is also observed under pure self-play (see Appendix D.4.6).

6 Conclusion and Limitations

This work shows that interaction memory can support conditional cooperation in sequential social dilemmas without reshaping rewards or explicitly modeling opponent learning. Theoretically, we derive sufficient conditions under which memory-conditioned responses achieve a strict local return advantage over memoryless rules when distinct histories share the same observation. Empirically, benefits depend on actor–outcome representation, temporal order, and memory horizon: one-event memory performs best under mixed-partner training in Coin Game, whereas longer histories improve collective resource management in CleanUp-lite. Together, these findings highlight memory's role in representing social context for conditional cooperation. Our study nevertheless has three limitations. First, our evaluation covers controlled two-agent SSDs with discrete actions and well-defined event boundaries; generalization to larger populations, continuous actions, and cooperative contributions without clear event boundaries remains untested. Second, memory representations and horizons are manually specified for each task; adapting them automatically to new environments remains future work. Finally, event-wise shuffling disrupts temporal order but may also alter slot-specific input distributions, preventing attribution of its effects solely to temporal order.

AI USE STATEMENT

We used ChatGPT (5.6 Sol and 5.6 Terra) to assist with literature searches, improve wording and clarity across the manuscript, review theoretical arguments, and discuss interpretations of experimental results. The authors manually reviewed the AI-assisted revisions and suggestions and checked the references against the cited sources. AI tools were not used to design the experimental methods, write experimental code, or perform numerical data analysis. The authors take full responsibility for the accuracy and integrity of the final manuscript, including all AI-assisted content.

ETHICS STATEMENT

This work studies interaction memory and cooperation among artificial agents in the simulated Coin Game and CleanUp-lite environments. Cooperation and fairness are evaluated using task-specific measures of behavior and the distribution of returns among agents. These measures do not directly establish fairness or societal benefit in real-world settings. Applications involving people would require additional assessment of their effects on affected individuals and groups.

REPRODUCIBILITY STATEMENT

We provide the theoretical proofs in Appendix A and the training procedure in Appendix B. Appendix C describes the rule-based partner strategies and their grounding in the experimental environments. Appendix D.1 specifies the environments, Appendix D.2 reports the network architectures and hyperparameters, and Appendix D.3 defines the evaluation metrics. Additional experimental details and results are provided in Appendix D.4. The source code is available in an anonymous repository at `https://anonymous.4open.science/r/MAMPT-4E44/`.

REFERENCES

John P Agapiou, Alexander Sasha Vezhnevets, Edgar A Duéñez-Guzmán, Jayd Matyas, Yiran Mao, Peter Sunehag, Raphael Köster, Udari Madhushani, Kavya Kopparapu, Ramona Comanescu, et al. Melting pot 2.0. *arXiv preprint arXiv:2211.13746*, 2022.

Milad Aghajohari, Juan Duque, Timotheus Cooijmans, and Aaron Courville. Loqa: Learning with opponent q-learning awareness. In *International Conference on Learning Representations*, volume 2024, pp. 14792–14807, 2024.

Robert Axelrod. Effective choice in the prisoner's dilemma. *Journal of conflict resolution*, 24(1): 3–25, 1980.

Robert Axelrod and William D Hamilton. The evolution of cooperation. *science*, 211(4489): 1390–1396, 1981.

Bruno Beaufils, Jean-Paul Delahaye, and Philippe Mathieu. Our meeting with gradual: A good strategy for the iterated prisoner's dilemma. In *Artificial Life V: Proceedings of the Fifth International Workshop on the Synthesis and Simulation of Living Systems*, pp. 202. MIT press, 1997.

Quentin Bertrand, Juan Agustin Duque, Emilio Calvano, and Gauthier Gidel. Self-play q-learners can provably collude in the iterated prisoner's dilemma. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 3952–3975. PMLR, 13–19 Jul 2025. URL `https://proceedings.mlr.press/v267/bertrand25a.html`.

Lucian Busoniu, Robert Babuska, and Bart De Schutter. A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 38(2):156–172, 2008.

Colin W. Clark. The economics of overexploitation. *Science*, 181(4100):630–634, 1973. doi:10.1126/science.181.4100.630. URL `https://www.science.org/doi/abs/10.1126/science.181.4100.630`.

Yali Du, Joel Z Leibo, Usman Islam, Richard Willis, and Peter Sunehag. A review of cooperation in multi-agent learning. *arXiv preprint arXiv:2312.05162*, 2023.

Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 122–130, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

James W Friedman. A non-cooperative equilibrium for supergames. *The Review of Economic Studies*, 38(1):1–12, 1971.

Drew Fudenberg and Eric Maskin. The folk theorem in repeated games with discounting or with incomplete information. *Econometrica*, 54(3):533–554, 1986.

Daniel Gildea and Daniel Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002. doi:10.1162/089120102760275983. URL https://aclanthology.org/J02-3001/.

Sven Gronauer and Klaus Diepold. Multi-agent deep reinforcement learning: a survey. *Artificial Intelligence Review*, 55(2):895–943, 2022.

Eric A Hansen, Daniel S Bernstein, and Shlomo Zilberstein. Dynamic programming for partially observable stochastic games. In *AAAI*, volume 4, pp. 709–715, 2004.

Garrett Hardin. The tragedy of the commons. *Science*, 162(3859):1243–1248, 1968. doi:10.1126/science.162.3859.1243.

Christian Hilbe, Luis A Martinez-Vaquero, Krishnendu Chatterjee, and Martin A Nowak. Memory-n strategies of direct reciprocity. *Proceedings of the National Academy of Sciences*, 114(18): 4715–4720, 2017.

Ray Hilborn, Ricardo Oscar Amoroso, Christopher M Anderson, Julia K Baum, Trevor A Branch, Christopher Costello, Carryn L De Moor, Abdelmalek Faraj, Daniel Hively, Olaf P Jensen, et al. Effective fisheries management instrumental in improving fish stock status. *Proceedings of the National Academy of Sciences*, 117(4):2218–2224, 2020.

Sarah Susanna Hoppler, Robin Segerer, and Jana Nikitin. The six components of social interactions: Actor, partner, relation, activities, context, and evaluation. *Frontiers in Psychology*, Volume 12 - 2021, 2022. ISSN 1664-1078. doi:10.3389/fpsyg.2021.743074. URL https://www.frontiersin.org/journals/psychology/articles/10.3389/fpsyg.2021.743074.

David Earl Hostallero, Daewoo Kim, Sangwoo Moon, Kyunghwan Son, Wan Ju Kang, and Yung Yi. Inducing cooperation through reward reshaping based on peer evaluations in deep multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pp. 520–528, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, Heather Roff, and Thore Graepel. Inequity aversion improves cooperation in intertemporal social dilemmas. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/7fea637fd6d02b8f0adf6f7dc36aed93-Paper.pdf.

Max Jaderberg, Wojciech M Czarnecki, Iain Dunning, Luke Marris, Guy Lever, Antonio Garcia Castaneda, Charles Beattie, Neil C Rabinowitz, Ari S Morcos, Avraham Ruderman, et al. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364(6443):859–865, 2019.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, DJ Strouse, Joel Z Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International conference on machine learning*, pp. 3040–3049. PMLR, 2019.

Graham Kendall, Xin Yao, and Siang Yew Chong. *The iterated prisoners' dilemma: 20 years on*, volume 4. World Scientific, 2007.

Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. A unified game-theoretic approach to multiagent reinforcement learning. *Advances in neural information processing systems*, 30, 2017.

Joel Z. Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '17, pp. 464–473, Richland, SC, 2017. International Foundation for Autonomous Agents and Multiagent Systems.

Adam Lerer and Alexander Peysakhovich. Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*, 2017.

Alistair Letcher, Jakob Foerster, David Balduzzi, Tim Rocktäschel, and Shimon Whiteson. Stable opponent shaping in differentiable games. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SyGjjsC5tQ.

Juan Li, Xiaowei Zhao, Bing Li, Charlotte S. L. Rossetti, Christian Hilbe, and Haoxiang Xia. Evolution of cooperation through cumulative reciprocity. *Nature Computational Science*, 2(10): 677–686, 2022. doi:10.1038/s43588-022-00334-w.

Kaiwen Liu, Nan Li, H Eric Tseng, Ilya Kolmanovsky, and Anouck Girard. Interaction-aware trajectory prediction and planning for autonomous vehicles in forced merge scenarios. *IEEE Transactions on Intelligent Transportation Systems*, 24(1):474–488, 2022.

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.

Andrei Lupu and Doina Precup. Gifting in multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pp. 789–797, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

Kevin R. McKee, Ian Gemp, Brian McWilliams, Edgar A. Duèñez Guzmán, Edward Hughes, and Joel Z. Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pp. 869–877, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

Kevin R McKee, Edward Hughes, Tina O Zhu, Martin J Chadwick, Raphael Koster, Antonio Garcia Castaneda, Charlie Beattie, Thore Graepel, Matt Botvinick, and Joel Z Leibo. A multi-agent reinforcement learning model of reputation and cooperation in human groups. *arXiv preprint arXiv:2103.04982*, 2021.

Alexander Meulemans, Seijin Kobayashi, Johannes von Oswald, Nino Scherrer, Eric Elmoznino, Blake A Richards, Guillaume Lajoie, Blaise Aguera y Arcas, and Joao Sacramento. Multi-agent cooperation through learning-aware policy gradients. In *International Conference on Learning Representations*, volume 2025, pp. 45978–46009, 2025.

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.

Per Molander. The optimal level of generosity in a selfish, uncertain environment. *Journal of Conflict Resolution*, 29(4):611–618, 1985.

Martin Nowak and Karl Sigmund. A strategy of win-stay, lose-shift that outperforms tit-for-tat in the prisoner's dilemma game. *Nature*, 364(6432):56–58, 1993.

Alexander Peysakhovich and Adam Lerer. Consequentialist conditional cooperation in social dilemmas with imperfect information, 2018a. URL https://arxiv.org/abs/1710.06975.

Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, pp. 2043–2044, Richland, SC, 2018b. International Foundation for Autonomous Agents and Multiagent Systems.

William H Press and Freeman J Dyson. Iterated prisoner's dilemma contains strategies that dominate any evolutionary opponent. *Proceedings of the National Academy of Sciences*, 109(26):10409–10413, 2012.

Kyrill Schmid, Lenz Belzner, and Claudia Linnhoff-Popien. Learning to penalize other learning agents. In *Artificial Life Conference Proceedings 33*, volume 2021, pp. 59, 2021.

Wilko Schwarting, Alyssa Pierson, Javier Alonso-Mora, Sertac Karaman, and Daniela Rus. Social behavior for autonomous vehicles. *Proceedings of the National Academy of Sciences*, 116(50): 24972–24978, 2019.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

DJ Strouse, Kevin McKee, Matt Botvinick, Edward Hughes, and Richard Everett. Collaborating with humans without human data. *Advances in neural information processing systems*, 34:14502–14515, 2021.

Qi Su, Hongyu Wang, Yu Xia, and Long Wang. A multi-agent reinforcement learning framework for exploring dominant strategies in iterated and evolutionary games. *Nature Communications*, 2025.

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), October 2023. ISSN 1471-2954. doi:10.1098/rspb.2023.1716. URL http://dx.doi.org/10.1098/rspb.2023.1716.

Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to incentivize other learning agents. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 15208–15219. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ad7ed5d47b9baceb12045a929e7e2f66-Paper.pdf.

Stephen Zhao, Chris Lu, Roger B Grosse, and Jakob Foerster. Proximal learning with opponent-learning awareness. *Advances in Neural Information Processing Systems*, 35:26324–26336, 2022.

John L Zhou, Weizhe Hong, and Jonathan C Kao. Reciprocal reward influence encourages cooperation from self-interested agents. *Advances in Neural Information Processing Systems*, 37:59491–59512, 2024.

A PROOF

Lemma 1 (Memory distinguishes aliased histories)**.** *Consider an agent i with a finite action set $\mathcal{A}_i$ and two histories τ and τ' that induce the same current observation, $o_t^i(\tau) = o_t^i(\tau')$. Let*

$$\mathcal{A}_i^\star(\tau) \coloneqq \arg\max_{a_i \in \mathcal{A}_i} Q_i^\star(\tau, a_i)$$

denote the set of optimal actions after τ, where $Q_i^\star(\tau, a_i)$ is the return from taking a_i and following an optimal continuation policy. If

$$\mathcal{A}_i^\star(\tau) \cap \mathcal{A}_i^\star(\tau') = \varnothing,$$

no memoryless decision rule can select an optimal action under both histories. If the retained memory distinguishes them, $h_t^i(\tau) \neq h_t^i(\tau')$, and the memory-conditioned policy class can represent distinct choices at the resulting augmented inputs, then a memory-conditioned decision rule can select an optimal action under each history.

Proof. Because the two histories induce the same observation, any memoryless decision rule uses the same action distribution under both:

$$\pi_i\big(\cdot \mid o_t^i(\tau)\big) = \pi_i\big(\cdot \mid o_t^i(\tau')\big)\,.$$

Selecting an optimal action under τ requires this distribution to assign probability one to $\mathcal{A}_i^\star(\tau)$; selecting an optimal action under τ' requires it to assign probability one to $\mathcal{A}_i^\star(\tau')$. Their intersection is empty, so no single distribution satisfies both requirements.

If $h_t^i(\tau) \neq h_t^i(\tau')$, the augmented inputs are distinct:

$$\big(o_t^i(\tau), h_t^i(\tau)\big) \neq \big(o_t^i(\tau'), h_t^i(\tau')\big).$$

Choose $a_i^\star \in \mathcal{A}_i^\star(\tau)$ and $a_i^{\star\prime} \in \mathcal{A}_i^\star(\tau')$. By the assumed representational capacity, a memory-conditioned decision rule can satisfy

$$\pi_i\big(a_i^\star \mid o_t^i(\tau), h_t^i(\tau)\big) = 1, \qquad \pi_i\big(a_i^{\star\prime} \mid o_t^i(\tau'), h_t^i(\tau')\big) = 1.$$

It therefore selects an optimal action at each of the two histories. □

Proposition 1. *Restating Proposition 1 of Section 4.3, let $\gamma \in [0, 1)$, $p \in (0, 1)$, $r_C > r_D$, and $g, \ell > 0$. In the stated two-state abstraction, the memory-conditioned response rule chooses C after h_C and D after h_D, while an observation-only response rule chooses C with probability $q \in [0, 1]$ at the shared observation o. Both rules use the same state-dependent continuation after this response. If $\gamma > g/(g + r_C - r_D)$, then $\Delta(q) > 0$ for every $q \in [0, 1]$.*

Proof. As in Section 4.3, let $Q(h, a)$ denote the expected discounted return from taking response a in history context h and then following the common state-aware continuation rule, which selects C in S_C and D in S_D.

We first derive the discount threshold. Choosing C in S_C preserves that state, yielding:

$$Q(h_C, C) = \frac{r_C}{1 - \gamma}.$$

Choosing D yields $r_C + g$ immediately and moves to S_D, where the common continuation earns r_D per step. Hence,

$$Q(h_C, D) = r_C + g + \gamma \frac{r_D}{1 - \gamma}.$$

Their comparison gives:

$$\begin{aligned} Q(h_C, C) > Q(h_C, D) &\iff \frac{r_C}{1 - \gamma} > r_C + g + \gamma \frac{r_D}{1 - \gamma} \\ &\iff \gamma(r_C - r_D) > g(1 - \gamma) \\ &\iff \gamma > \frac{g}{g + r_C - r_D}. \end{aligned}$$

Thus, under the stated condition,

$$A \coloneqq Q(h_C, C) - Q(h_C, D) = \frac{\gamma(g + r_C - r_D) - g}{1 - \gamma} > 0.$$

In S_D, both responses preserve the state and share the same continuation value; their immediate rewards differ by ℓ:

$$Q(h_D, D) - Q(h_D, C) = \ell > 0.$$

Substituting these gaps into the fixed-observation return difference defined in Section 4.3 gives:

$$\begin{aligned}\Delta(q) &= p(1-q)A + (1-p)q\ell \\ &\geq \min\{pA, (1-p)\ell\} > 0,\end{aligned}$$

where the inequality follows because the right-hand side is a convex combination of pA and $(1-p)\ell$. □

Supplementary complete-policy implication. Proposition 1 concerns the first response at an aliased observation under a common continuation. A stronger complete-policy comparison holds if the retained history identifies the current state at *every* decision step. Let $J_p(\pi)$ be the expected discounted return of a complete policy from the initial state mixture p and $1-p$, and let Π_0 contain all observation-only policies, including randomized and time-dependent ones. Let π_{mem} choose C in S_C and D in S_D at each step.

Under the discount condition above, C is preferable to deviating once to the absorbing state S_D, while D is optimal in S_D. These choices satisfy the Bellman optimality equations, so π_{mem} is state-wise optimal and $J_p(\pi_{\mathrm{mem}}) = U_{\mathrm{mem}}$, where $U_{\mathrm{mem}} = pQ(h_C, C) + (1-p)Q(h_D, D)$. For any $\pi_0 \in \Pi_0$, let $q = \Pr_{\pi_0}(C \mid o)$. Since both states produce o, the same q applies in both states. Even granting π_0 optimal state-aware decisions after its first response gives the upper bound:

$$\begin{aligned}\bar{U}_{\mathrm{memless}}(q) \coloneqq\ & p\Big[qQ(h_C, C) + (1-q)Q(h_C, D)\Big] \\ & + (1-p)\Big[qQ(h_D, C) + (1-q)Q(h_D, D)\Big].\end{aligned}$$

Hence,

$$J_p(\pi_0) \leq \bar{U}_{\mathrm{memless}}(q).$$

Consequently,

$$\begin{aligned}J_p(\pi_{\mathrm{mem}}) - J_p(\pi_0) &\geq U_{\mathrm{mem}} - \bar{U}_{\mathrm{memless}}(q) = \Delta(q) \\ &\geq \min\{pA, (1-p)\ell\} > 0.\end{aligned}$$

The bound is uniform over Π_0, and therefore:

$$J_p(\pi_{\mathrm{mem}}) > \sup_{\pi_0 \in \Pi_0} J_p(\pi_0).$$

This supplementary conclusion relies on state identification at every step and does not alter the local scope of Proposition 1.

B ALGORITHM

Algorithm 1 Memory-Augmented Mixed-Partner Training (MAMPT)

Initialize: agent pool $\mathcal{A}$, mentor pool $\mathcal{M}$, memory horizon k, network parameters $\{\theta_i\}_{i\in\mathcal{A}}$, buffers $\{\mathcal{D}_i\}_{i\in\mathcal{A}} = \emptyset$, sampling probability p_{agent}
for each episode e **do**
  Sample focal learner $i \sim \mathcal{A}$ and $u \sim \mathrm{Uniform}(0, 1)$
  if $u < p_{agent}$ **then**
    Sample learning partner $j \sim \mathrm{Uniform}(\mathcal{A})$ (excluding i) and set $\mathcal{U}_e = \{i, j\}$
  else
    Sample mentor $j \sim \mathrm{Uniform}(\mathcal{M})$ and set $\mathcal{U}_e = \{i\}$
  end if
  Initialize histories $\{\tau_\ell = \emptyset\}_{\ell\in\mathcal{U}_e}$ and, if $j \in \mathcal{M}$, the mentor's history $\tau_j = \emptyset$; observe $\{o_0^\ell\}_{\ell\in\mathcal{U}_e}$
  for $t = 0$ to $T - 1$ **do**
    for each learner $\ell \in \mathcal{U}_e$ **do**
      Construct memory $h_t^\ell \leftarrow \mathrm{Mem}_k(\tau_\ell)$ and augmented input $x_t^\ell \leftarrow (o_t^\ell, h_t^\ell)$
      Choose action $a_t^\ell \sim \pi_{\theta_\ell}^\ell(x_t^\ell)$
    end for
    If $j \in \mathcal{M}$, choose a_t^j using its rule and history τ_j
    Execute joint action $\boldsymbol{a}_t$ and observe $\boldsymbol{r}_t, \boldsymbol{o}_{t+1}$
    for each learner $\ell \in \mathcal{U}_e$ **do**
      Update τ_ℓ with the observed interaction and construct $x_{t+1}^\ell \leftarrow (o_{t+1}^\ell, \mathrm{Mem}_k(\tau_\ell))$
      Store $(x_t^\ell, a_t^\ell, r_t^\ell, x_{t+1}^\ell)$ in $\mathcal{D}_\ell$
    end for
    If $j \in \mathcal{M}$, update τ_j with the observed interaction
  end for
  for each learner $\ell \in \mathcal{U}_e$ **do**
    Update θ_ℓ using samples from $\mathcal{D}_\ell$
  end for
end for

C RULE-BASED STRATEGIES

C.1 MATRIX-GAME STRATEGIES

Training rule-based strategies. We construct the training partner pool from eight classical strategies for repeated matrix games. Each strategy operates on an interaction history whose entries are represented as cooperation (C) or defection (D). Unless stated otherwise, the strategies begin with cooperation. Their decision rules are defined as follows.

1. **Always Defect (ALLD).** ALLD selects D at every interaction event, independently of the partner's behavior or the preceding interaction history (Axelrod, 1980).
2. **Tit-for-Tat (TFT).** TFT begins with cooperation and subsequently copies the partner's most recent response. Thus, cooperation is reciprocated with cooperation, whereas defection is met with defection (Axelrod, 1980).
3. **Generous Tit-for-Tat (GTFT).** GTFT follows TFT after cooperation but may forgive a defection. When the partner's latest response is D, GTFT selects C with probability ρ and D with probability $1 - \rho$. We use $\rho = 0.3$ in all experiments (Molander, 1985).
4. **Win-Stay, Lose-Shift (WSLS).** WSLS repeats its previous response when the partner previously cooperated and switches its response when the partner previously defected. Accordingly, its responses to the previous joint outcomes (C, C), (C, D), (D, C), and (D, D) are C, D, D, and C, respectively (Nowak & Sigmund, 1993).
5. **Grim.** Grim, also known as Hold-a-Grudge, cooperates until the first defection by the partner. After observing a defection, it switches permanently to D and does not return to cooperation (Friedman, 1971).

6. **Fool-Me-Once.** Fool-Me-Once retaliates against the partner's first defection by selecting D. It subsequently returns to cooperation if no further defection occurs. Once a second partner defection is observed, the strategy switches permanently to D.
7. **Omega Tit-for-Tat (OmegaTFT).** We use the common configuration $\mathrm{OmegaTFT}(3, 8)$. OmegaTFT behaves like TFT but additionally incorporates deadlock and randomness detection:
   - *Deadlock detection.* The deadlock counter tracks alternating cycles between the joint responses (C, D) and (D, C). When the counter reaches three, OmegaTFT selects C to break the cycle and resets the counter.
   - *Randomness detection.* The randomness counter increases when the partner changes its response between consecutive interaction events or when its latest response differs from OmegaTFT's preceding response. The counter is reduced following repeated cooperative responses. Once it reaches eight, OmegaTFT switches to persistent defection.

   When neither condition is active, OmegaTFT copies the partner's most recent response. It thereby combines the reciprocity of TFT with mechanisms for responding to alternating and irregular behavior (Kendall et al., 2007).
8. **Gradual Tit-for-Tat (GradualTFT).** GradualTFT increases the duration of punishment with the number of observed partner defections. After the nth defection, it selects D for n interaction events and then enters a two-event cooling-off period in which it selects C. Once this response sequence is completed, the strategy returns to cooperation unless another defection has added a new punishment sequence (Beaufils et al., 1997).

Held-out evaluation rule-based strategies. To evaluate generalization beyond the behavioral patterns encountered during training, we use four additional rule-based strategies that are excluded from the training partner pool. These strategies test delayed retaliation, cumulative reciprocity, memory-two conditional behavior, and majority-based responses.

1. **Tit-for-Two-Tats (TF2T).** TF2T begins with cooperation and defects only when both of the partner's two most recent responses are D. A single defection is therefore forgiven, whereas two consecutive defections trigger retaliation. In all other cases, TF2T selects C (Axelrod, 1980).
2. **Cumulative Reciprocity (CURE).** CURE conditions its response on the cumulative imbalance between the two players' defections. Let $N^D_{\mathrm{self}}(t)$ and $N^D_{\mathrm{partner}}(t)$ denote their respective numbers of defections before interaction event t. CURE selects C whenever
$$N^D_{\mathrm{partner}}(t) - N^D_{\mathrm{self}}(t) \leq \Delta,$$
   and selects D otherwise. We use the goodwill tolerance $\Delta = 1$. Thus, CURE tolerates a small cumulative imbalance while retaliating when the partner has defected substantially more often than CURE itself (Li et al., 2022).
3. **Memory-Two Bilateral Reciprocity (MTBR).** We use a deterministic memory-two variant of MTBR adapted to the asynchronous interaction events in Coin Game. The strategy begins with cooperation. After observing only one partner response, it copies that response. Once two responses are available, it applies the following rules:
   - it selects C after two consecutive partner cooperations;
   - it selects D after two consecutive partner defections;
   - when the two players are trapped in an alternating cycle between (C, D) and (D, C), it selects C to break the cycle;
   - otherwise, it copies the partner's most recent response.

   This operational variant retains MTBR's use of two-event history and its explicit mechanism for repairing alternating deadlocks (Su et al., 2025).
4. **Soft Majority.** Soft Majority begins with cooperation and compares the cumulative numbers of the partner's cooperative and defective responses. Let $N^C_{\mathrm{partner}}(t)$ and $N^D_{\mathrm{partner}}(t)$ denote these counts before interaction event t. It selects
$$C \quad \text{if} \quad N^C_{\mathrm{partner}}(t) \geq N^D_{\mathrm{partner}}(t),$$
   and selects D otherwise. Ties are therefore resolved in favor of cooperation.

Both the training and evaluation strategies determine only a high-level response in $\{C, D\}$. To apply them in the Coin Game, each response is grounded in an environment-specific low-level policy: C pursues own-color coins, while D pursues the current coin regardless of color. Realized pickup outcomes are labeled separately, as described in Section C.2.

C.2 Strategy Grounding in Sequential Social Dilemmas

The matrix-game strategies produce high-level responses rather than primitive environment actions. We ground these responses using environment-specific behavior policies.

Coin Game. We label realized own-color pickups as C and other-color pickups as D when constructing interaction histories. These labels describe outcomes, not necessarily the mentor's intended high-level response. Only realized pickups are recorded; timesteps without a pickup are excluded.

- **Cooperation (C).** An own-color pickup is labeled C. When its high-level response is C, the mentor pursues an own-color coin and avoids other-color coins.
- **Defection (D).** An other-color pickup is labeled D. When its high-level response is D, the mentor greedily pursues the current coin regardless of color.

Primitive actions are selected according to the resulting Manhattan distance from the coin, with ties broken uniformly.

CleanUp-lite. CleanUp-lite requires both cleaning and harvesting and therefore does not assign a fixed C/D label to each event. Instead, each mentor selects a cleaning or harvesting target.

1. **AlwaysClean.** Always select cleaning.
2. **AlwaysHarvest.** Always select harvesting.
3. **ReciprocalClean.** Cleans for the first 20 steps and then matches the partner's most recent cleaning or harvesting event.
4. **MostlyClean.** Select cleaning with probability 0.8 and harvesting otherwise.
5. **TurnTakingClean.** Alternates between cleaning and harvesting every 100 environment steps.

The low-level policy moves toward the nearest pollution cell when cleaning is selected and toward the nearest apple when harvesting is selected. If no target is available, the agent patrols the corresponding spawn region. Only successful cleaning and harvesting events are added to the interaction history; movement and zapping steps are excluded.

D EXPERIMENTAL DETAILS

D.1 ENVIRONMENTS

D.1.1 COIN GAME

Coin Game is a two-player grid-world social dilemma in which agents collect colored coins (Lerer & Peysakhovich, 2017). Each agent is assigned a fixed color, red or blue. It is modelled as follows:

- **State:** The world is a 2D grid of size 5×5. The state contains the positions of the two agents, the position of the current coin, and the coin color.
- **Dynamics:** At each timestep, both agents move simultaneously, with movements clipped at the grid boundary. When one or both agents reach the coin cell, the coin is collected. After collection, a new coin is sampled on an unoccupied cell, and its color is sampled from the two agent colors.
- **Observation:** Each agent observes a four-channel grid tensor. The channels encode the spatial locations of the observing agent, the opponent, the red coin, and the blue coin.
- **Action:** There are four actions: $\{\text{move up}, \text{move down}, \text{move left}, \text{move right}\}$.
- **Reward:** When an agent collects an own coin, it receives a reward of $+1$, while its partner receives no penalty. When an agent collects an other coin, it receives a reward of $+1$, while its partner receives a reward of -2.

D.1.2 CLEANUP-LITE

CleanUp-lite is a simplified two-player version of the CleanUp social dilemma, following the compact formulation used in `COALA-PG` experiments (Meulemans et al., 2025). It is modelled as follows:

- **State:** The world is a 2D grid of size 5×4. The leftmost column is the river, and the rightmost column is the orchard. In our implementation, the river also includes a small horizontal extension, while dirt can spawn only in the leftmost river column. Cells in the orchard column can be occupied by apples. The world state also contains the position of each agent and each agent's freeze timer.
- **Dynamics:** The environment evolves at every timestep in the following order:
  1. When there is at least one pollution-spawn cell that is not occupied by dirt, a new patch of dirt is spawned with probability $p_{\text{pollution}} = 0.35$, and placed randomly on one of the free pollution-spawn cells.
  2. When there is at least one orchard cell that is not occupied by an apple, a new apple is spawned with probability $p_{\text{apple}} = 1 - \min(1, P/P_{\text{threshold}})$, where P is the total number of dirt cells in the environment and $P_{\text{threshold}} = 3$. The spawned apple is placed randomly on one of the free orchard cells.
  3. When an agent that is not frozen visits a cell with an apple, it harvests the apple and receives reward $r_{\text{apple}} = 1$. The apple is then replaced by an empty cell.
  4. When an agent that is not frozen visits a cell with a dirt patch, it cleans the dirt patch and replaces it with an empty cell. Cleaning gives no immediate reward.
  5. When an agent that is not frozen chooses the zap action, it attempts to freeze the co-player. If the co-player is at most two cells away in Manhattan distance, the zap succeeds with probability $p_{\text{zap}} = 0.9$. If the zap succeeds, the co-player is frozen for $t_{\text{zap}} = 5$ timesteps and cannot be further zapped during this period.
- **Observation:** The observation contains full information about the environment. Each agent observes its own position, the co-player's position, the full object grid encoded with one-hot object types (empty, apple, dirt), and the frozen state of both agents.
- **Action:** There are six actions: $\{\text{move right}, \text{move left}, \text{move up}, \text{move down}, \text{zap}, \text{do nothing}\}$.
- **Reward:** When an agent picks up an apple, it receives reward $r_{\text{apple}} = 1$ in that timestep. Cleaning dirt and zapping do not produce reward.

D.2 HYPERPARAMETERS

In the main comparisons, we fix the environment hyperparameters and evaluation protocol. Learning hyperparameters are selected using predefined sweeps averaged over 3 tuning seeds. Unless noted otherwise, final performance is evaluated over 5 fresh matched seeds not used for hyperparameter selection.

The general implementation settings, network architectures, and DQN hyperparameters are reported in Tables 3–5.

Table 3: General framework settings.

Framework parameter	**Value**
`deep_learning_framework`	PyTorch
`gpu_model`	NVIDIA RTX 3090
`gpu_memory`	24 GB
`population_size`	16
`p_agent`	0.50
`checkpoint_interval`	5×10^6

Table 4: Network architectures of the memory-augmented agents.

Architecture parameter	**Coin Game**	**CleanUp-lite**
`network_type`	MLP	MLP
`observation_preprocessing`	Flatten	MLP encoder
`obs_encoder_layers`	–	2
`obs_latent_dim`	–	16
`memory_integration`	Concatenation	Concatenation
`num_hidden_layers`	2	2
`hidden_dim`	128	128
`hidden_activation`	Tanh	Tanh
`output_activation`	Linear	Linear
`action_dim`	4	6

Table 5: DQN optimization hyperparameters.

RL hyperparameter	**Coin Game**	**CleanUp-lite**
`algorithm`	DQN	DQN
`optimizer`	Adam	Adam
`learning_rate`	3×10^{-4}	3×10^{-4}
`gamma`	0.96	0.95
`loss_function`	Huber	Huber
`buffer_size`	20,000	20,000
`batch_size`	512	512
`learning_starts`	2,000	2,000
`gradient_steps`	1	1
`target_update_interval`	1,000	1,000
`max_grad_norm`	10	10
`epsilon_start`	0.20	0.20
`epsilon_end`	0.02	0.02
`epsilon_decay_steps`	5×10^5	5×10^5

Coin Game. The Coin Game environment and memory settings are reported in Table 6; mentor strategies are listed in Table 7.

Table 6: Environment and memory-horizon settings for Coin Game.

Parameters	**Value**
`num_agents`	2
`grid_size`	5×5
`action_dim`	4
`max_episode_length`	200
`max_training_timesteps`	3×10^7
`coin_reward`	$+1$
`other_coin_penalty`	-2
`memory_lengths`	$\{0, 1, 2, 5, 10\}$
`shuffled_memory_lengths`	$\{2, 5, 10\}$
`memory_order`	Newest to oldest
`memory_update_event`	Coin pickup

Table 7: Mentor strategies used in Coin Game.

Strategy	**Response rule**	**Parameter**
`TFT`	Copy the partner's last response	Initial C
`GTFT`	TFT with probabilistic forgiveness	`forgive_prob`$= 0.30$
`WSLS`	Stay after C; switch after D	Initial C
`Grim`	Permanent D after the first D	–
`Fool-Me-Once`	Permanent D after the second D	–
`OmegaTFT`	TFT with deadlock/randomness detection	–
`GradualTFT`	n defections after the n-th D	Two-event cooling-off
`ALLD`	Always defect	–

CleanUp-lite. The CleanUp-lite environment and memory settings are reported in Table 8. Its mentor strategies are listed in Table 9.

Table 8: Environment and memory settings for CleanUp-lite.

Parameters	**Value**
num_agents	2
grid_size	5×4
action_dim	6
max_episode_length	500
max_training_timesteps	4×10^7
pollution_spawn_prob	0.35
initial_dirt_count	3
apple_threshold	3
apple_reward	$+1$
cleaning_reward	0
zap_success_prob	0.90
zap_range	2
zap_freeze_steps	5
memory_lengths	$\{0, 1, 5, 10, 20\}$
memory_order	Newest to oldest
memory_update_event	Clean or harvest

Table 9: Mentor strategies used in CleanUp-lite.

Strategy	**Target rule**	**Parameter**
AlwaysClean	Always clean	–
AlwaysHarvest	Always harvest	–
ReciprocalClean	Copy the partner's last event	warmup_steps$= 20$
MostlyClean	Probabilistic cleaning	clean_prob$= 0.80$
TurnTakingClean	Alternate cleaning and harvesting	interval$= 100$

D.3 Evaluation Metrics

Coin Game. Let N_i^{same} and N_i^{other} denote the numbers of own coins and other coins collected by agent i, respectively. We measure cooperation using

$$p_{\text{same},i} = \frac{N_i^{\text{same}}}{N_i^{\text{same}} + N_i^{\text{other}}}.$$

A larger $p_{\text{same},i}$ indicates that agent i more frequently avoids imposing other-coin pickup penalties on other agents. We also report the average number of other-coin pickups, for which lower values indicate fewer harmful interactions.

The average episodic return of agent i is

$$\overline{R}_i = \frac{1}{M} \sum_{m=1}^{M} \sum_{t=0}^{T_m-1} r_{i,t}^{(m)},$$

where M is the number of evaluated episodes. Overall social welfare is measured by the mean return across all N agents:

$$R_{\text{collective}} = \frac{1}{N} \sum_{i=1}^{N} \overline{R}_i.$$

To quantify reward equality, we use a Gini-based fairness score that supports both positive and negative returns:

$$F_{\text{Gini}} = 1 - \text{clip}\left(\frac{\sum_{i=1}^{N} \sum_{j=1}^{N} \left| \overline{R}_i - \overline{R}_j \right|}{2N \sum_{i=1}^{N} \left| \overline{R}_i \right| + \varepsilon}, 0, 1 \right),$$

where ε is a small positive constant. A larger F_{Gini} indicates a more equal distribution of returns, with $F_{\text{Gini}} = 1$ when all agents receive identical returns.

CleanUp-lite. In CleanUp-lite, we evaluate collective performance, resource maintenance, and direct conflict. Average reward is computed across all N agents as

$$\overline{R} = \frac{1}{N} \sum_{i=1}^{N} \overline{R}_i.$$

Because harvesting an apple produces a reward of $+1$, a larger average reward indicates more effective long-term resource management.

Let P_t denote the number of pollution cells at timestep t. The average pollution level is

$$\overline{P} = \frac{\sum_{m=1}^{M} \sum_{t=0}^{T_m-1} P_t^{(m)}}{\sum_{m=1}^{M} T_m}.$$

A smaller $\overline{P}$ indicates better maintenance of the shared resource and more favorable conditions for apple regeneration.

Finally, let N_i^{zap} denote the number of zap attempts made by agent i, and let N_{steps} denote the total number of evaluated environment steps. The mean zap rate across all N agents is

$$\overline{Z} = \frac{1}{N} \sum_{i=1}^{N} \frac{N_i^{\text{zap}}}{N_{\text{steps}}}.$$

A smaller zap rate indicates less direct conflict among agents.

D.4 Additional Experimental Results

This section reports additional experiments that extend the memory analyses in the main text. We first compare a broader set of memory representations in the Coin Game. We then inspect action-level policy changes under controlled memory interventions. Finally, we examine the roles of memory horizon and temporal order in the Coin Game, and further verify that memory benefits also emerge under self-play in CleanUp-lite.

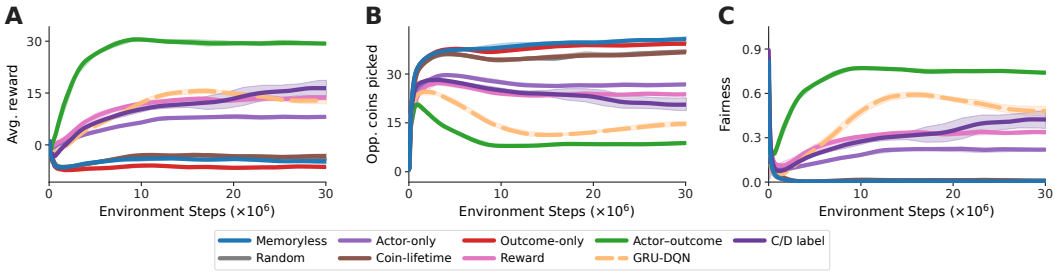


Figure 4: Alternative memory representations in the Coin Game. Training curves compare memoryless agents with event-level, compressed, trajectory-level, reward-based, and random memory representations. (A) Average reward. (B) Other-coin pickups. (C) Fairness. Curves show means over the available runs: five seeds for Actor–outcome, Actor-only, and Outcome-only, and four seeds for each other displayed condition. Shaded regions denote one standard deviation.

D.4.1 Expanded memory representations.

Event construction. We divide each episode into coin-pickup events. An event begins when a coin appears and ends when it is collected. The pickup determines the collector, whether the coin was own-color or other-color for that collector, and the resulting rewards. Memory is updated at each pickup and retained until the next one. The one-event conditions below therefore encode the most recent completed pickup.

Controlled Actor–outcome factorial. We construct a controlled two-by-two factorial that varies whether memory retains (i) the identity of the actual collector and (ii) whether the collector picked up an own-color or other-color coin. Each condition provides the DQN with the same two-entry memory vector: the first entry encodes the collector (Player 0 or Player 1), and the second encodes the pickup outcome (own or other).

When either factor is absent, the corresponding entry is set to a fixed sentinel value, $s_{\text{mask}} = -1$, chosen to lie outside the support of valid actor and outcome codes. This preserves the input dimension while removing event-specific information without introducing a random nuisance value. All four conditions use the same event definition, one-event memory horizon, update schedule, DQN architecture, parameter count, hyperparameters, training and evaluation protocols, and five matched seeds. Thus, the factorial conditions differ only in whether actor and outcome information from the actual pickup event is retained.

- **Null-content control.** Both event attributes are masked, with both entries fixed to s_{mask}. This condition provides the null reference for the factorial.
- **Actor-only.** Only actor information is retained. The first entry records the actual collector, and the second is fixed to s_{mask}.
- **Outcome-only.** Only outcome information is retained. The first entry is fixed to s_{mask}, and the second records the actual own/other pickup outcome.
- **Actor–outcome.** Both factors are retained. The first entry records the actual collector, and the second records the actual own/other pickup outcome.

The Null-content condition serves as the reference cell of the factorial. Comparisons with Actor-only and Outcome-only estimate the contribution of each event attribute when the other is unavailable. Comparisons between Actor–outcome and the two partial conditions test whether joint access to both attributes provides additional benefit beyond either attribute alone.

Additional memory representations. We additionally evaluate several alternative representations, including the original Random condition. Each condition supplies a two-entry memory vector to preserve the input dimension of the downstream DQN. A representation with only one natural value

fixes its second entry at zero, while Memoryless uses zeros for both entries. These representations provide broader baselines but are not part of the controlled two-by-two factorial.

- **Random.** Neither event-specific factor is retained. The first entry is an independently sampled player identity, and the second is an independently sampled own/other code. These entries are unrelated to the actual pickup event.
- **C/D labels.** The first entry records the inferred cooperation or defection label of the focal player, and the second records that of the opponent. Cooperation is encoded as 0 and defection as 1. Labels are inferred from the players' trajectories and the realized pickup in the preceding event.
- **GRU–DQN.** A trainable GRU encodes the trajectory from the beginning of the current episode up to the current step into a two-dimensional hidden state. Each trajectory step contains the positions and actions of both players together with the coin position. Although its output dimension matches the other conditions, the GRU introduces additional trainable parameters; we therefore treat it as a recurrent baseline with episode-level memory rather than a capacity-matched factorial condition.
- **Coin-lifetime.** The first entry records the normalized number of steps from the coin's appearance to its collection, and the second is fixed at zero.
- **Reward.** The first entry records the focal player's normalized reward, and the second records the opponent's normalized reward at the end of the event.
- **Memoryless.** Both entries are fixed at zero. This condition preserves the DQN input size without providing information from previous events.

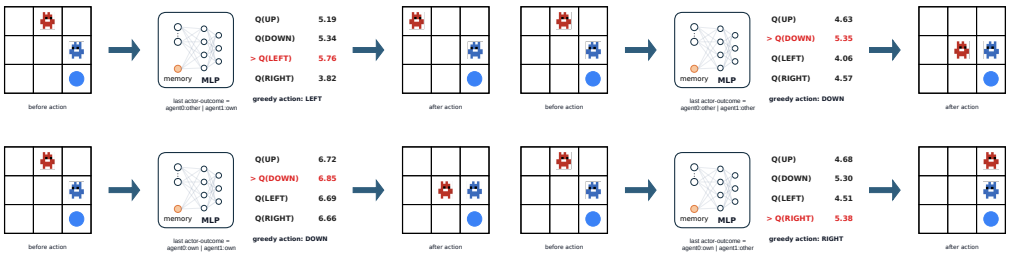
The controlled factorial comparison is summarized in the main text. Actor–outcome achieved the best reward and fairness and the fewest other-coin pickups; Actor-only was intermediate, while Outcome-only did not improve over the Null-content control. Because these four conditions share the input dimension, masking scheme, update timing, architecture, and parameter count, the result cannot be attributed to those factors alone. Figure 4 shows training curves for the available representations, excluding Null-content. Random, C/D labels, GRU–DQN, reward, and the other structured representations provide comparisons outside the factorial.

D.4.2 Microscopic analysis of conditional cooperation.

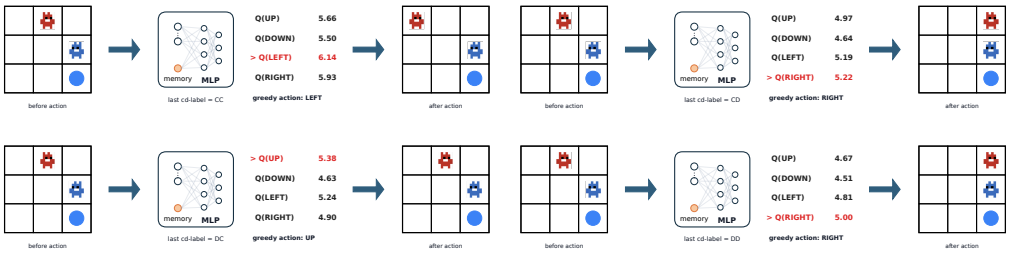
To inspect how learned policies use memory, we hold the current observation fixed, vary the remembered event, and compare the resulting Q-values and greedy actions. Figure 5 shows this intervention for Actor–outcome, C/D-label, Outcome-only, and Random memories. These illustrative examples characterize local action sensitivity; they do not, by themselves, establish a population-level causal effect.

D.4.3 Memory horizon for cooperation

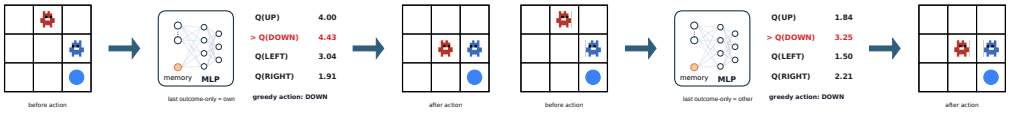
We next evaluated how memory horizon shaped cooperation across partner contexts. Across horizons, we held the event representation, DQN hidden architecture, optimization settings, training budget, partner-sampling distribution, evaluation protocol, and matched seeds fixed. Increasing k enlarged the explicit memory input and, consequently, the input layer, but did not yield monotonic improvements in cooperation. In self-play, memory-augmented agents generally achieved higher cooperation than the memoryless baseline, with one-event memory ($k = 1$) attaining the highest cooperation level (Figure 6A). A similar pattern appeared against cooperative or reciprocal mentors, including TFT, GTFT, WSLS, OmegaTFT, and GradualTFT: one-event memory performed best, whereas longer histories provided no further gains over $k = 1$ (Figure 6B–F). Against punitive or defecting partners, the effect depended on the partner strategy. One-event memory also improved cooperation against Hold-a-Grudge and Fool-Me-Once, whereas cooperation against ALLD varied little across memory horizons (Figure 6G–I). These results show that the benefits of interaction memory depend on both its horizon and the partner's behavior; retaining more events does not necessarily improve cooperation.



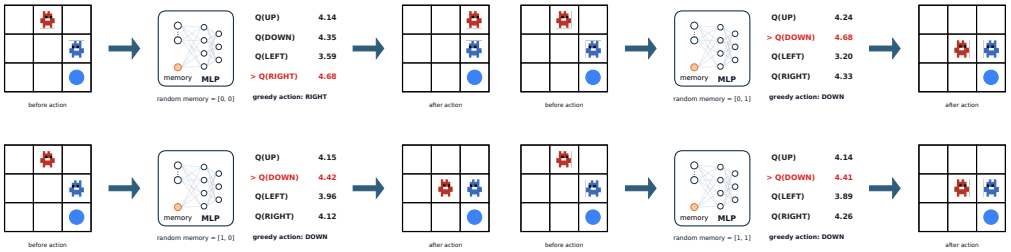
(a) Actor-outcome memory.



(b) C/D-label memory.



(c) Outcome-only memory.



(d) Random memory.

Figure 5: Action-level analysis of memory-conditioned policies. The current observation is held fixed while the memory input is varied. Each example reports the resulting Q-values, greedy action, and next state. The panels illustrate how each trained policy's action values change when only the remembered event is varied.

D.4.4 Shuffle temporal order for cooperation

To examine temporal organization at fixed memory horizon, we compared ordered and event-wise shuffled histories for $k \in \{2, 5, 10\}$; both conditions retained the same k most recent Actor–outcome entries and shared the input dimension, DQN configuration, training and evaluation protocols, and five matched seeds. Ordered histories arranged records from newest to oldest, whereas shuffled histories applied a new random permutation after each coin pickup and retained it until the next pickup. Shuffling therefore preserved the events and input dimension while disrupting each event's

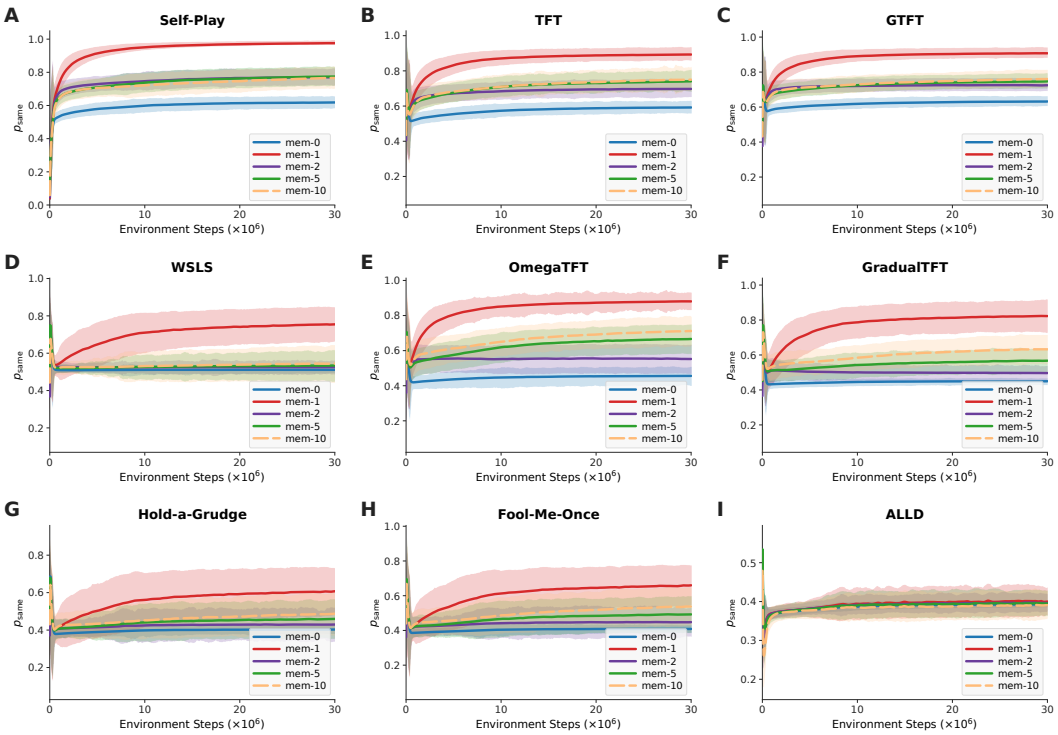


Figure 6: Memory horizon shapes cooperation across partner contexts in the Coin Game. (A) Mean cooperation level in self-play between learning agents with the same memory horizon. (B–F) Learners evaluated against cooperative or reciprocal mentors, including TFT, GTFT, WSLS, OmegaTFT, and GradualTFT. (G–I) Learners evaluated against punitive or defecting partners, including Hold-a-Grudge, Fool-Me-Once, and ALLD. Curves show the mean over 5 matched seeds, and shaded regions denote one standard deviation.

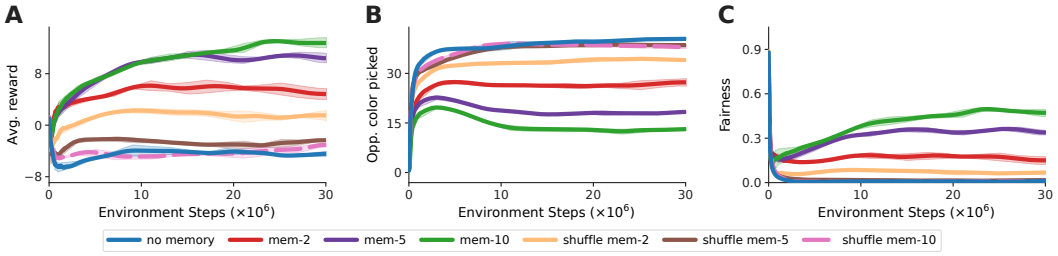


Figure 7: Memory horizon and temporal order in the Coin Game. Ordered memory histories are compared with shuffled histories that preserve the same events and input size. (A) Average reward. (B) Other-coin pickups. (C) Fairness. Curves show the mean over 5 matched seeds, and shaded regions denote one standard deviation.

assignment to a temporal slot. Longer ordered histories improved reward and fairness and reduced other-coin pickups, whereas shuffling weakened these gains, particularly at $k = 5$ and $k = 10$ (cf. Figure 7 and Table 10).

D.4.5 Training-mechanism ablation for conditional cooperation

The preceding results show that memory-augmented agents respond differently to cooperative and defecting partners. However, it remains unclear whether this conditional behavior arises from interaction memory alone or from its combination with the diverse partner behaviors encountered during mixed-partner training. To separate these effects, we compare pure self-play with mixed-

Table 10: Mean p_{same} at convergence in self-play and against mentors, averaged over 5 matched seeds. Parenthetical values in the shuffled-memory rows indicate changes relative to the corresponding original-memory.

Condition	Self-play	TFT	GTFT	WSLS	OmegaTFT	GradualTFT	Hold-a-Grudge	Fool-Me-Once	ALLD
Memoryless	0.618	0.592	0.632	0.511	0.456	0.450	0.403	0.409	0.393
Original, $k = 2$	0.770	0.697	0.725	0.526	0.552	0.496	0.430	0.448	0.396
Shuffled, $k = 2$	0.692 (−0.078)	0.668 (−0.029)	0.686 (−0.039)	0.516 (−0.010)	0.523 (−0.029)	0.467 (−0.029)	0.413 (−0.017)	0.426 (−0.022)	0.395 (−0.001)
Original, $k = 5$	0.774	0.741	0.747	0.534	0.666	0.567	0.460	0.492	0.395
Shuffled, $k = 5$	0.645 (−0.129)	0.602 (−0.139)	0.632 (−0.115)	0.509 (−0.025)	0.482 (−0.184)	0.455 (−0.112)	0.404 (−0.056)	0.410 (−0.082)	0.392 (−0.003)
Original, $k = 10$	0.766	0.750	0.760	0.544	0.712	0.632	0.487	0.538	0.391
Shuffled, $k = 10$	0.608 (−0.158)	0.572 (−0.178)	0.607 (−0.153)	0.508 (−0.036)	0.456 (−0.256)	0.449 (−0.183)	0.402 (−0.085)	0.406 (−0.132)	0.391 (0.000)

Table 11: Training-mechanism ablation in the Coin Game.

Training	Memory	After coop. (%)	After def. (%)	Gap (pp)
Self-play	None	56.3 (0.2)	47.5 (0.3)	8.8
Self-play	$k = 1$	69.7 (0.2)	55.5 (1.1)	14.2
Mixed-partner	None	63.1 (0.2)	48.5 (0.6)	14.6
Mixed-partner	$k = 1$	**94.3 (0.1)**	**51.3 (1.9)**	**43.0**

partner training, both with and without one-event interaction memory. For each frozen policy, we condition p_{same} on whether the partner's preceding pickup was cooperative or defective. As shown in Table 11, one-event memory increases the conditional-response gap from 8.8 to 14.2 percentage points under self-play. Under mixed-partner training, the gap increases from 14.6 without memory to 43.0 with memory. This comparison suggests that mixed-partner training amplifies the effect of interaction memory on conditional cooperation.

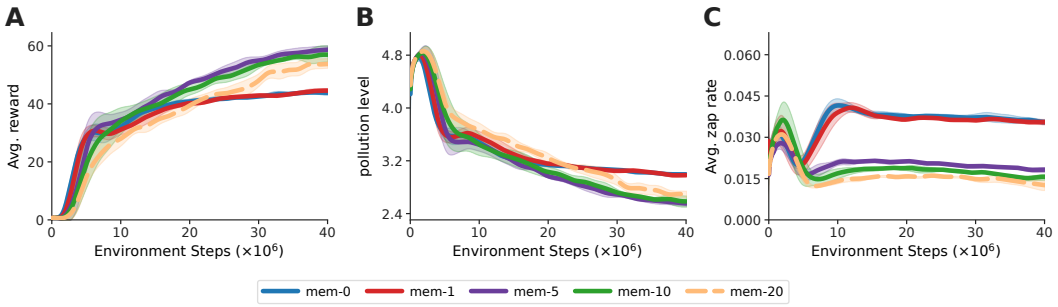


Figure 8: Memory effects under pure self-play training in CleanUp-lite. Agents interact only with learning agents during training. (A) Average reward across both agents. (B) Average pollution level. (C) Average zap rate. Curves show the mean over 5 matched seeds, and shaded regions denote one standard deviation.

D.4.6 Pure self-play training in CleanUp-lite

To evaluate whether the benefits of memory depend on exposure to fixed mentors, we repeat the CleanUp-lite experiments under pure self-play. As shown in Figure 8, agents with $k \geq 5$ achieve final rewards approaching 60, compared with approximately 44 for agents with $k \leq 1$, while also maintaining lower pollution. The longest-memory agents additionally exhibit the lowest zap rates. Although pure self-play learns more slowly than mixed-partner training, it achieves comparable or better late-training outcomes, indicating that fixed mentors accelerate early learning but are not required for the benefits of memory to emerge.