LEARNING AWARENESS FOR EFFECTIVE AND EFFICIENT LEARNING IN SOCIAL DILEMMAS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world multi-agent systems involve social dilemmas, where agents have partially aligned but conflicting interests, and mutual cooperation can benefit agents in long run. Social dilemmas are challenging, as self-interested agents often choose strategies that are suboptimal, particularly when other agents are also learning. Recent learning-aware methods offer a promising direction, but often make restrictive assumptions about how agents understand the coplayers' learning processes. In this work, we conceptualize learning as action within a decision model to help characterize learning awareness (LA). We accordingly introduce the learning awareness order (LA-order) over learning agents, where an LA(k+1) agent can "exploit" an LAk coplayer by leveraging its information about the coplayer's learning process. The LA-order naturally provides a lens to understand a wide spectrum of learners, ranging from fixed random agents (LA0), self-interested naive learners (LA1), to prior learning-aware agents ($LA2^{+}$). Based on this understanding, we introduce a model-free, decentralized multi-agent reinforcement learning (MARL) approach extensible to any LA-orders, which performs learning-aware RL through dynamic grouping of agents, with each agent leveraging solely its own experience. Through evaluation across diverse social dilemma challenges, in both iterated matrix games and state-based benchmarks, we demonstrate the effectiveness of our algorithms, including under ad-hoc generalization, and the value of the LA-order. Experiments also show that agents with adjacent LA-orders mutually influence each other's learning dynamics.

1 INTRODUCTION

Social dilemmas (Von Neumann & Morgenstern, 1947; Rapoport & Chammah, 1965) are ubiquitous in real-life multi-agent systems. At a societal scale, the tragedy of the commons (Hardin, 1968) illustrates how individually rational behavior can produce poor long-term outcomes: when participants use shared resources sustainably, all can continue to benefit, whereas short-term overuse of resources such as water, forests, or fisheries can lead to depletion that ultimately harms everyone. At a smaller scale, a group of engineers can share ideas, code, data, and credit fairly. Each may benefit in the short term from claiming disproportionate credit, but persistent cooperation produces better work for all. In both cases, agents face partially aligned but conflicting interests: cooperation can improve their long-term outcomes, while each agent may have an incentive to pursue its immediate self-interest.

Such dilemmas become particularly challenging in multi-agent reinforcement learning (MARL) (Foerster et al., 2017; Bertrand et al., 2025). When multiple agents learn simultaneously, an agent's effective environment and strategic incentives change as its coplayers update their policies. Consequently, a learning strategy that appears individually advantageous against a fixed coplayer may become suboptimal over repeated interactions with a learning coplayer. Addressing social dilemmas therefore requires reasoning not only about other agents' current behavior, but also about how their behavior *changes* through learning. This challenge is exacerbated by the extensive focus of MARL on purely cooperative settings (Oroojlooy & Hajinezhad, 2023; Du et al., 2023; Albrecht et al., 2024), where agents share a common objective and such social dilemmas do not arise.

Recent learning-aware methods offer a promising direction for addressing such dilemmas in MARL. Informally, learning awareness (LA) concerns an agent's use of information about how its co-players learn, rather than only how they currently behave (He et al., 2016; Albrecht & Stone, 2018). This

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.

can be particularly useful in sequential social dilemmas, where an agent's actions affect not only immediate rewards but also co-players' future policy updates, and hence the long-term consequences.

Existing approaches leverage such an idea in various ways (Foerster et al., 2017; Letcher et al., 2019; Willi et al., 2022; Zhao et al., 2022; Aghajohari et al., 2024; Duque et al., 2024; Lu et al., 2022; Khan et al., 2023; Meulemans et al., 2024), but often rely on strong assumptions about co-player's learning process, or remain limited in how effectively they exploit learning dynamics across heterogeneous co-players, such as settings that requires ad-hoc generalization (Bowling & McCracken, 2005; Stone et al., 2010). This also motivates a more general understanding of learning awareness and its role in social dilemmas. Our central research question can thus be stated as follows:

How can intelligent agents achieve effective learning in multi-agent systems involving social dilemmas, and what role does learning awareness play?

Main contributions. To address the research question, our work can be summarized as follows:

- We conceptualize learning as action within a multi-agent decision model, with information about coplayers' learning processes represented through observations. This framing accommodates diverse learning algorithms and observability assumptions, providing a unified basis for understanding learning awareness (LA) in multi-agent systems. (Section 3.2)
- We characterize learning awareness through the information agents use about coplayers' learning processes and introduce a learning awareness order (LA-order). This facilitates understanding of existing methods and lays a foundation for designing our learning-aware algorithms. (Section 3.3)
- We propose LARL, a model-free, decentralized training algorithm that supports multiple LA-orders by meta-learning from experience spanning coplayers' policy updates. We develop static and population-based variants, encouraging diversity among agents at each LA-order. (Section 4)
- Experiment shows that LARL performs consistently better across well-established social dilemma tasks Cleanup, Coins, and Coop-Mining; we demonstrate the LA order's relativeness nature and mutual influence on learning dynamics over adjacent LA orders; LARL further demonstrates superior adhoc generalization results with unseen coplayers. (Section 5)

2 PRELIMINARY

Decision model: POSG. We model the multi-agent sequential decision process as a partially observable stochastic game (POSG, Kuhn (1953); Becker & Sunberg (2024)), defined as a tuple $\mathtt{M} = (\mathcal{N}, \mathcal{S}, \mathcal{A}, \Omega, \mathcal{O}, \mathtt{P}, \mathtt{R}, \mu, \gamma, \mathtt{H})$, where $\mathcal{N}$ is the finite set of N agents ($i \in \mathcal{N}$), $\mathcal{S}$ is the set of states, $\mathcal{A}^i$ is the set of agent i's actions, $\mathtt{P}(\mathtt{s}_{\mathtt{t}+1}|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$ is the Markovian transition function, $\mathcal{O}^i$ is the set of agent i's observations, $\Omega^i(\mathtt{o}^i_\mathtt{t}|\mathtt{s}_{\mathtt{t}-1}, \mathtt{a}_{\mathtt{t}-1}, \mathtt{s}_\mathtt{t}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{O}^i \to [0,1]$ is agent i's observation function, $\mathtt{R}^i(\mathtt{r}^i_\mathtt{t}|\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) : \mathcal{S} \times \mathcal{A} \times \mathbb{R} \to [0,1]$ is agent i's reward function, $\mu : \mathcal{S} \to [0,1]$ is the initial state distribution, $\gamma \in [0,1]$ is the discount factor, $\mathtt{H} \in \mathbb{N}^+ \cup \{+\infty\}$ is the horizon. At each timestep $\mathtt{t} \in \{0, \dots, \mathtt{H}-1\}$, each agent i chooses an action according to its history-dependent policy $\mathtt{a}^i_\mathtt{t} \sim \pi^i_\mathtt{t}(\cdot|\tau^i_\mathtt{t})$, parameterized by $\theta^i \in \Theta^i$, where $\tau^i_\mathtt{t} \in \mathcal{H}^i_\mathtt{t} = (\mathcal{O}^i_{0:\mathtt{t}}, \mathcal{A}^i_{0:\mathtt{t}-1})$ is agent i's observation-action history. We omit the subscript $\mathtt{t}$ to denote a complete trajectory $\tau^i = \tau^i_\mathtt{H}$. The objective of agent i is to maximize its expected cumulative reward $\mathtt{J}^i(\theta^i, \theta^{-i}) = \mathbb{E}_{\tau\sim\pi_\theta}[\sum_{\mathtt{t}=0}^{\mathtt{H}} \gamma^\mathtt{t} \mathtt{r}^i(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})]$, where τ denotes a joint trajectory sampled from the joint policy $\pi = (\pi^i, \pi^{-i})$. We use $-i$ to denote all agents except i, and omit the superscript i for joint actions $\mathtt{a} = \times_{i\in\mathcal{N}}\mathtt{a}^i$, observations $\mathtt{o}$, rewards $\mathtt{r}$, policies π, trajectories τ, and their corresponding sets.

Unless stated otherwise, we make the following key assumptions throughout this work.

Assumption 2.1. Each agent $i \in \mathcal{N}$ only observes its local observation $\mathtt{o}^i$ and individual reward $\mathtt{r}^i$.

Assumption 2.2. Each agent $i \in \mathcal{N}$ is *rational* that aims to maximize its own expected return $\mathtt{J}^i$.

Common game-theoretic objectives such as Nash equilibria (Nash Jr, 1950), Pareto optimality (Pareto, 2014), social welfare, fairness, etc., are often not addressed following the POSG decision modeling and assumptions, therefore optimizing them are outside the scope of this work. It is worth highlighting that *no reward sharing* among agents is permitted for any learning algorithm that optimizes the per-agent objective.

3 LEARNING AWARENESS CHARACTERIZATION

We present surprising results in Iterated Prisoner's Dilemma (IPD), where self-play direct-gradient learners can achieve cooperative strategies. To understand this phenomenon along with prior learning-awareness literature, we introduce the learning-as-action framework, and characterize the learning awareness and learning awareness order.

3.1 MOTIVATING EXAMPLE: ITERATED PRISONER'S DILEMMA

Figure 1.(a) shows that the randomly initialized agent consistently achieves the always-cooperate policy through self-play (i.e., parameters shared over agents) in IPD. More interestingly, Figure 1.(b,c) show that the agent pretrained against the random policy first leads to the always-defect policy, then it shifts to Pavlov through subsequent self-play. Despite different underlying learning mechanisms, direct-gradient self-play and prior learning-aware approaches demonstrate similar behaviors. They both can "exploit" some coplayer: the naive gradient agent exploits the random policy, while the learning-aware agent exploits the naive gradient agent. Moreover, their exploitating policy can shift to some responsive cooperation policy. To facilitate the understanding of such behavior and their learning dynamics, in the following section, we characterize the learning awareness by first introducing the learning-as-action framework.

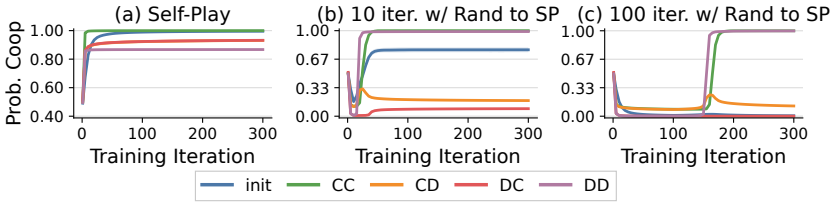


Figure 1: Direct-gradient learner's self-play performance in IPD. Each curve shows the agent's probability of cooperating given the corresponding previous state during training. (a) Self-play converges to always-cooperate. (b,c) Pretraining against the random policy followed by self-play converges to the Pavlov policy (win-stay, lose-switch). Section A.4 provides more details on this setting.

3.2 LEARNING-AS-ACTION DECISION MODEL

We broadly consider an agent's *learning process* as a set of computations it performs to update its policy (Shettleworth, 1984; Lachman, 1997; De Houwer et al., 2013). An agent i being aware of its coplayer j's learning implies some information about j's learning process is available to i. Such information should: (1) *vary over time* to reflect different information at different timesteps, and (2) be dependent on the coplayers' internal state and learning process at each timestep. To this end, it is natural to relate an agent's learning process to an action, and consider the learning-awareness as a part of an observation in a multi-agent decision model. We hereby introduce a learning-as-action formulation which accommodates general multi-agent interaction and learning.

We consider an agent's learning process in the original *inner* POSG $\mathtt{M}$ (Section 2) as an action. Accordingly, we formulate the multi-agent sequential learning process as a high-level *outer* POSG, $\tilde{M} = (\tilde{\mathcal{N}}, \tilde{\mathcal{S}}, \tilde{\mathcal{A}}, \tilde{\Omega}, \tilde{O}, \tilde{P}, \tilde{R}, \tilde{\mu}, \tilde{\gamma}, \tilde{H})$, where $\tilde{\mathcal{N}}$ is the finite set of N agents corresponding to $\mathcal{N}$ in inner $\mathtt{M}$.[1] In the outer POSG, each outer state $\tilde{s} \in \tilde{\mathcal{S}}$ represents the global state of all inner agents and POSG instances, which takes into account both intra-agent information and inter-agent couplings and induces the inner joint policy π, each outer action $\tilde{a}^i \in \tilde{\mathcal{A}}^i$ is all computations encapsulated in the inner agent i's learning algorithm, $\tilde{P}(\tilde{s}_{\tilde{t}+1}|\tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \tilde{\mathcal{S}} \to [0,1]$ is the Markovian transition function that executes the joint outer action, $\tilde{O}^i$ is the set of i's outer observations, $\tilde{\Omega}^i(\tilde{o}^i_{\tilde{t}}|\tilde{s}_{\tilde{t}-1}, \tilde{a}_{\tilde{t}-1}, \tilde{s}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \tilde{\mathcal{S}} \times \tilde{O}^i \to [0,1]$ is i's outer observation function, $\tilde{R}^i(\tilde{r}^i_{\tilde{t}}|\tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \mathbb{R} \to [0,1]$ is i's outer reward function, $\tilde{\mu} : \tilde{\mathcal{S}} \to [0,1]$ is the initial outer state distribution, $\tilde{\gamma} \in [0,1]$ is the outer discount factor, $\tilde{H} \in \mathbb{N}^+ \cup \{+\infty\}$ is the outer horizon representing the number of inner learning iterations. We define one outer timestep $\tilde{t}$ to correspond to

[1] If the context is unambiguous, for each corresponding outer agent $i_n \in \tilde{\mathcal{N}}$ and inner agent $j_n \in \mathcal{N}$ pair indexed by the same $n \in [N]$, we omit the subscript and denote either by i.

one inner timestep $\mathtt{t}$. The former denotes an inner learning iteration, while the latter denotes a single interaction step with the inner environment. At each outer timestep $\tilde{t} \in \{0, \dots, \tilde{H}-1\}$, each $i \in \tilde{\mathcal{N}}$ chooses an outer action according to its history-dependent outer policy $\tilde{a}^i_{\tilde{t}} \sim \tilde{\pi}^i_{\tilde{t}}(\cdot|\tilde{\tau}^i_{\tilde{t}})$, parameterized by $\tilde{\phi}^i \in \tilde{\Phi}_i$, where $\tilde{\tau}^i_{\tilde{t}} \in \tilde{\mathcal{H}}^i_{\tilde{t}} = (\tilde{\mathcal{O}}^i_{0:\tilde{t}})$ is i's outer observation history. For an agent i learning to optimize its outer policy, i.e., the policy that determines its inner learning process, its objective is to maximize its expected cumulative outer reward $\tilde{J}^i(\tilde{\phi}^i, \tilde{\phi}^{-i}) = \mathbb{E}_{\tilde{\tau}\sim\tilde{\pi}_{\tilde{\phi}}}[\sum_{\tilde{t}=0}^{\tilde{H}} \tilde{\gamma}^{\tilde{t}} \tilde{r}^i(\tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}})]$, where $\tilde{\tau}$ denotes a joint outer trajectory sampled from the joint outer policy $\tilde{\pi} = (\tilde{\pi}^i, \tilde{\pi}^{-i})$. To align the optimization of the outer and inner objectives, the outer reward function should be set such that the optimal outer policy yields the optimal inner policy. One choice of is to set them equal: $\tilde{r}^i_{\tilde{t}} = \mathtt{r}^i_{\mathtt{t}}$. We provide examples of instantiating this framing in Section A.2.

3.3 Learning awareness order

Building on the learning-as-action conceptual framework, we characterize learning awareness in terms of an agent's use of information about co-players' learning. Comparing the types of information agents use about each other's learning provides a notion of relative awareness, from which we assign nonnegative integer La-orders on selected subsets under suitable ordering assumptions.

Learning awareness. For each $j \in \tilde{\mathcal{N}} \setminus i$, let $\tilde{y}^{i,j}$ denote information about the inner learning process of j available to agent i, and define $\tilde{y}^{i,j} \coloneqq (\tilde{o}^{i,j}_{0:\tilde{H}-1})$, where $\tilde{o}^{i,j} \in \tilde{\mathcal{O}}^{i,j} \subseteq \tilde{\mathcal{O}}^i$ denotes agent i's outer observation of j's inner learning process. Thus, $\tilde{y}^{i,j}$ is the corresponding part of i's outer history $\tilde{\tau}^i$, including information implicitly encoded in its inner transitions or explicitly modeled or assumed available within its inner learning algorithm. Additionally, we define

$$y_{ij} \coloneqq \big\{\mathrm{type}(v) : v \in \tilde{y}^{i,j}\big\},$$

where $\mathrm{type}(v) \in \mathcal{Y}$ is a labeling map, and $\mathcal{Y}$ is a shared set of information *types*, independent of agent identity or capacity. We assume each listed type is relevant to this learning process. Examples include co-players' gradients (Foerster et al., 2017; 2018), value functions (Aghajohari et al., 2024), advantages (Duque et al., 2024), and interaction histories across learning steps (Lu et al., 2022; Meulemans et al., 2024), summarized in Table 1. Define

$$i \text{ is learning-aware of } j \quad \iff \quad y_{ij} \neq \emptyset. \tag{1}$$

Learning awareness order. Consider a set of agents $\tilde{\mathcal{N}}$. Denote non-learning agents as $\tilde{\mathcal{N}}_0$, and all learning agents as $\tilde{\mathcal{N}}_+ = \tilde{\mathcal{N}} \setminus \tilde{\mathcal{N}}_0$. A non-learning agent is an agent that does not perform any inner policy updates, i.e., $\pi^i_{\tilde{t}} = \pi^i_{\tilde{t}+1}$ for all $\tilde{t} \in \{0, \dots, \tilde{H}-1\}$. For distinct $i, j \in \tilde{\mathcal{N}}_+$, define

$$\mathrm{La}^i \succeq \mathrm{La}^j \quad \iff \quad y_{ji} \subseteq y_{ij}, \qquad \mathrm{La}^i \succ \mathrm{La}^j \quad \iff \quad y_{ji} \subsetneq y_{ij}. \tag{2}$$

and set $\mathrm{La}^i \succeq \mathrm{La}^i$. For every $i \in \tilde{\mathcal{N}}_0$, set $\mathrm{La}^j \succeq \mathrm{La}^i$ for all $j \in \tilde{\mathcal{N}}$ and $\mathrm{La}^j \succ \mathrm{La}^i$ for all $j \in \tilde{\mathcal{N}}_+$. Non-nested sets yield incomparable agents.

For later algorithm design, we assume that a subset of agents satisfies the following condition.

Assumption 3.1 (Ordered agent subset)**.** For a selected nonempty subset $\tilde{\mathcal{I}} \subseteq \tilde{\mathcal{N}}$, the restriction of $\succeq$ to $\tilde{\mathcal{I}}$ is a total preorder. Specifically, for all $i, j, k \in \tilde{\mathcal{I}}$,

$$(\mathrm{La}^i \succeq \mathrm{La}^j) \wedge (\mathrm{La}^j \succeq \mathrm{La}^k) \Longrightarrow \mathrm{La}^i \succeq \mathrm{La}^k \qquad \text{(transitivity)},$$
$$(\mathrm{La}^i \succeq \mathrm{La}^j) \vee (\mathrm{La}^j \succeq \mathrm{La}^i) \qquad \text{(totality)}.$$

This total preorder allows us to assign an nonnegative integer order to each agent in the subset.

Definition 3.2 (La-order)**.** Let $\tilde{\mathcal{I}}$ be a subset satisfying Assumption 3.1. We assign $\mathrm{La}^i \coloneqq 0$ for every $i \in \tilde{\mathcal{I}} \cap \tilde{\mathcal{N}}_0$. For every $i \in \tilde{\mathcal{I}} \setminus \tilde{\mathcal{N}}_0$, we assign consecutive orders $\mathrm{La}^i \in \{1, \dots, K\}$ in increasing order according to $\succeq$, with tied agents sharing an order. We denote the k-th order by $\mathrm{La}k$.

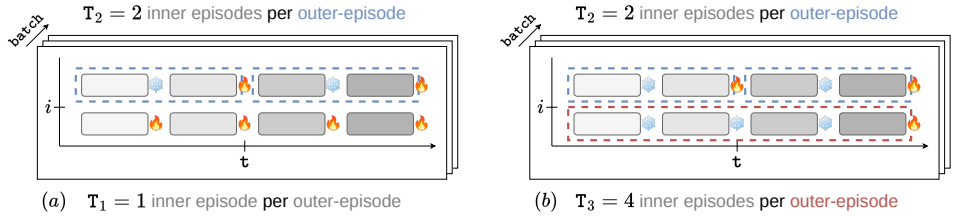
La-orders are *relative* to $\tilde{\mathcal{I}}$: e.g., removing all agents of order La2 shifts higher orders down by one. We provide examples in Section A.2.

Table 1: Learning-aware reinforcement learning methods. Blue means the ego agent's learning awareness information $\tilde{y}^{i,j}$ is obtained via direct observation of agent j, red means it is modeled or assumed accessible, and green means it is a combination of both. "Dec." represents "decentralized".

Approach	Assumptions on coplayers $-i$	Learning awareness information $\tilde{y}^{i,j}$	Model-free	Dec.	LA-order
LOLA	LA1; grad-based	(differentiable) gradients $\nabla_{\theta^{-i}} J^i, \nabla_{\theta^{-i}} J^{-i}$	✗	✓	LA2
LOLA-DICE	LA1; grad-based	(differentiable) gradients $\nabla_{\theta^{-i}} J^{-i}$	✗	✓	LA2+
LOQA	actor-critic; $\pi^{j\in-i} = \text{softmax}(\mathbb{E}_{\pi^{-j}}[Q^j(\mathtt{s},\mathtt{a})])$	$\hat{\pi}^{j\in-i}(\cdot\|\mathtt{s}) = f_{\theta^i}(\tau^j, \hat{Q}^j_{\theta^i}, Q^j)$, f_{θ^i} is C^1 in θ^i.	✗	✗	LA2
AdAlign	actor-critic; $\pi^{j\in-i} = \exp\beta(\mathbb{E}_{\pi^{-j}}[Q^j(\mathtt{s},\mathtt{a})])$	advantages $A^{j\in-i}(\mathtt{s}_t, \mathtt{a}_t)$			
MFOS	LA1; on-policy (inner) episodic update	$[\tau^i_{\theta_1}; \ldots; \tau^i_{\theta_T}], T > 1$	✓	✓	LA2
COALA					
LARL	on-policy (inner) episodic update	$[\tau^i_{\theta_1}; \ldots; \tau^i_{\theta_{T_k}}], k \geq 2, T_k > 1, T_k \,/\, T_{k-1} \in \mathbb{Z}^+_{>1}$	✓	✓	LA2+

4 METHOD

We propose an experience-based learning-awareness RL approach, LARL, where agents with different LA-orders take actions and optimize policies over trajectories over different numbers of consecutive inner episodes. Each agent has a combined inner and outer policy parametrization, and its learning-awareness stems from coplayer-learning-induced distribution shifts across multiple inner episodes, captured through the agent's own experience.



(a) $\mathtt{T}_1 = 1$ inner episode per outer-episode (b) $\mathtt{T}_3 = 4$ inner episodes per outer-episode

Figure 2: (a) LA1 and LA2 agents update their policies after every 1 and 2 inner episodes, respectively. (b) LA2 and LA3 agents update their policies after every 2 and 4 inner episodes, respectively. An on-policy agent with less frequent policy updates *naturally* observes histories that reflects the learning processes of coplayers with more frequent updates.

We first introduce the agent rollout and optimization. Given the LA$[K]$ agents, where $\forall k \in \{1, \ldots, K\}$, the set of LA$k$ agents is denoted by $\mathcal{N}_k$. Every agent $i_k \in \mathcal{N}_k$ treats T_k consecutive inner episodes as its (outer) episode horizon, and performs trajectory rollout and policy optimization over this horizon across all inner episode boundaries. We let $T_1 = 1$ for the LA1 naive agents. For $k \geq 2$, the outer episode satisfies $T_k > 1$ and $T_k \,/\, T_{k-1} \in \mathbb{Z}^+_{>1}$. For instance, Figure 2.(a) illustrates an LA1-vs-LA2 scenario, where the LA1 and LA2 agents update their policies after every 1 and 2 inner episodes, respectively. Through batched rollouts, the LA2 agent's trajectory experiences reflects the non-stationarity across inner episodes induced by the LA1 agent's learning, on which LA2 optimizes its policy conditioned. In Figure 2.(b), the LA3 agent's optimization could exploit the LA2 agent's learning process similarly. In practice, we equip each agent with a GRU-based architecture, and LA2+ agents adopt the RL^2's meta-RL approach that, (2) during rollout, preserves the hidden state across inner episodes boundaries within the same outer episode, and (3) during optimization, performs gradient updates within each group, and update all groups in sequence without aggregating the gradients across groups for the common agent appearing in multiple groups.

We then present the agent grouping method, which groups agents with the same LA-order (LAk-vs-LAk) and adjacent LA-order (LAk-vs-LAk-1). Every agent $i \in \bigcup_k \mathcal{N}_k$ has a unique identity $n \in [N]$ such that the agent group formed in the environment covers all N roles. For instance, as in Figure 3, the "Tom and Jerry" environment enforces the two agents to be one "Tom" and one "Jerry", regardless of their LA-order. For $k \in [K] \setminus \{1\}$, each $i_k \in \mathcal{N}_k$ has $(N-1)$ unique LA$(k-1)$ coplayer candidates denoted by $\mathcal{N}^{i_k}_{(k-1)} \subset \mathcal{N}_{(k-1)}$. Every i_k in a LAk-vs-LAk-1 grouping should be grouped with its own $\mathcal{N}^{i_k}_{(k-1)}$, and every i_k in a LAk-vs-LAk grouping should be grouped with the complement roles in $\mathcal{N}_k$. In the same example, the same LA-order groups are pairs formed within

Algorithm 1: Static LARL for LA$[K]$ agents

input : num_train_iterations L, LAk agents' grouping scheduler LambSched_k, Sampler without replacement Rand, agents $\bigcup_{k\in[K]} \mathcal{N}_k$

output : updated agents $\bigcup_{k\in[K]} \mathcal{N}_k$

for *train iteration* $l = 1, \ldots, L$ **do**
  for LA-*order* $k \in [K]$ **do**
    // Construct (up to) $|\mathcal{N}_k|$ groups of N agents, where each agent's LA order ($\in \{k, k-1\}$) depends on LambSched_k
    updated LAk agents $\mathcal{D}_k \leftarrow \emptyset$
    for *agent* $i_k \in \mathcal{N}_k$ **do**
      if $i_k \in \mathcal{D}_k$ **then continue**
      agent group $\mathcal{C} \leftarrow \{i_k\}$
      $\lambda_k \leftarrow \text{LambSched}_k(l)$
      // n^{i_k} denotes i_k's agent-in-group idx
      for *agent idx* $n \in [N] \setminus \{n^{i_k}\}$ **do**
        if $\lambda_k < \text{Uniform}[0, 1]$ **then**
          $\mathcal{C} \leftarrow \mathcal{C} \cup \text{Rand}(\mathcal{N}_k^n \setminus \{i_k\})$
        else
          $\mathcal{C} \leftarrow \mathcal{C} \cup \text{Rand}(\mathcal{N}_{(k-1)}^{i_k,n})$
      $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \mathcal{C}_k$
      rollout and update $\mathcal{C}$ policies (Algorithm 3)

Table 2: Payoff matrices of classical social dilemmas in matrix games.

	Cooperate (C)	Defect (D)
Cooperate (C)	1,1	-1,2
Defect (D)	2,-1	0,0

(a) Prisoner's Dilemma

	Swerve (C)	Straight (D)
Swerve (C)	1,1	0,2
Straight (D)	2,0	-1,-1

(b) Chicken Game

	Stag (C)	Hare (D)
Stag (C)	2,2	-1,1
Hare (D)	1,-1	0,0

(c) Stag Hunt

each $\mathcal{N}_k$ group, and the adjacent LA-order groups are pairs formed by every two vertically adjacent nodes, whose frames share the same color. Note that every group includes one "Tom" and one "Jerry".

LARL slowly transitions from LAk-vs-LAk-1 groupings to LAk-vs-LAk groupings, following a per-LA-order grouping scheduler adopted from Lu et al. (2022), unless otherwise specified. Algorithm 1 details the algorithm. We additionally introduces a population-based algorithm variant (Algorithm 2), which follows the same agent definition and grouping rules, but includes a population of agents at each LA-order. It encourages collective behavioral diversity via a policy-divergence-based auxiliary objective.

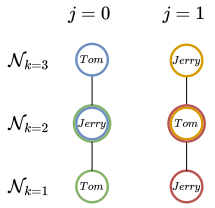


Figure 3: An exemplary agent grouping for $K = 3, N = 2$ (Algorithm 1). Each node represent an agent, and agents shown in the same color within the same $\mathcal{N}_k$ indicate a valid grouping.

5 Experiments

In this section, we present empirical study on a suite of tasks, including iterated matrix games and state-based tasks that encompass diverse social dilemma paradigms. We begin with experiments that show that high LA-order agents exploit low LA-order agents, and the exploiting policies achieve cooperative behavior via intra-play. We then present results that manifest the relativeness of the learning awareness order, and mutual influences on learning dynamics between adjacent LA-order agents. Finally, we show results on ad-hoc generalization where mutually unseen agents are evaluated in cross-play. We demonstrate the effectiveness of LARL and the value of the LA-order.

From the taxonomy of classical matrix games (Bruns, 2015), three symmetric general-sum games have Pareto-optimal outcomes distinct from their pure-strategy Nash equilibria: Chicken, Prisoner's Dilemma, and Stag Hunt, each posing a distinct challenge. We consider tasks with the same cooperation-defection tradeoffs – Cleanup (Meulemans et al., 2024), Coins (Lerer & Peysakhovich,

2017), and Coop-mining (Agapiou et al., 2022) – and their corresponding iterated matrix games – ICG, IPD, and ISH. Detailed description of the tasks can be found in Section B.

5.1 Experimental setup

In each experiment, we train each algorithm for 1000 iterations with 5 independent runs. We consider two training schemes: (i) exploitation, which enforces LAk-vs-LAk-1 groupings and (ii) cooperation, which for each experience-based method, starts from LAk-vs-LAk-1 groupings and gradually transitions to LAk-vs-LAk following a grouping scheduler during training. As described in Algorithm 1, at each iteration, LAk's scheduler determines its coplayer is LAk-1 with probability λ_k, or LAk otherwise. The scheduling parameter λ_k anneals linearly by ϵ_k at each iteration. The exploitation scheme can be viewed as following a constant $\lambda_k = 1$. LARL's LA order is set to 2 unless otherwise specified. We call the learning setting LAk-vs-LAk intra-play, i.e., all agents in the group follow the same algorithm and configuration, yet they do not share parameters.

In exploitation experiments, for each agent we average its evaluation metric over 5 seeds. In cooperation experiments involving LAk-vs-LAk groupings, as there are two agents leveraging the same algorithm, we average its evaluation metric over the 10 seeds of both agents. The quantitative evaluation metrics is the mean episode reward over training iterations. The mean return of each seed is computed by normalizing batched episode return by the batch size and the number of timesteps in the episode. We evaluate the decentralized learning approaches listed in Table 1, where LOLA implements the LOLA-DICE update rule. The LA1 agent is PPO with per-episode updates.

5.2 Main cooperation results

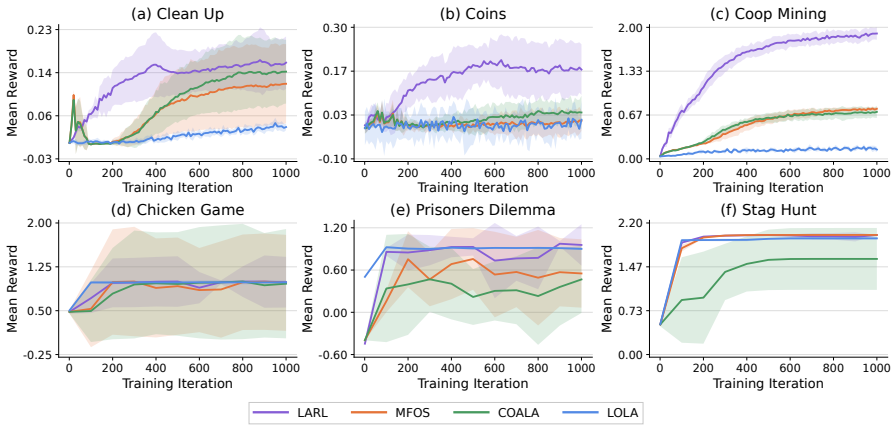


Figure 4: Learning-aware methods promote cooperation. LARL performs best overall.

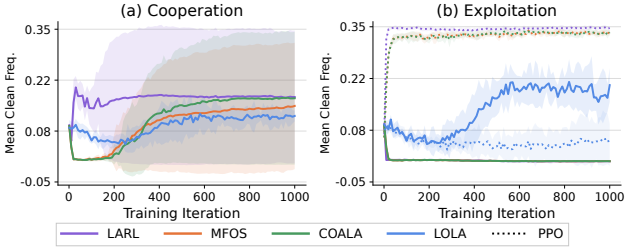


Figure 5: Frequency of cleaning the pollution. The naive coplayer of each LA agent is shown by a dotted line in the same color. (a) corresponds to Figure 4.(a); Learning-aware agents actively clean the pollution. (b) corresponds to Figure 8.(a); Learning-aware agents exploit their naive coplayer and do not clean the pollution.

In this section, we evaluate the strength of cooperative policies learned by each learning-aware approach. Figure 4 shows the mean episode rewards for the learning-aware methods. LARL achieves

Table 3: Experience-based learning-aware methods exploit their naive coplayer. In each task, the largest exploitation gap is marked in **bold**.

Approach	Clean Up	Coins	Coop Mining	Chicken Game	Prisoners Dilemma	Stag Hunt
LARL	0.20 ± 0.02	0.28 ± 0.06	1.82 ± 0.10	1.90 ± 0.00	1.68 ± 0.02	1.92 ± 0.00
r_Δ	**0.08**	**0.58**	0.00	1.92	**2.15**	-0.01
MFOS	0.16 ± 0.01	0.22 ± 0.03	0.26 ± 0.01	1.89 ± 0.00	0.18 ± 0.25	1.88 ± 0.02
r_Δ	0.05	0.44	**0.14**	**1.93**	0.12	0.03
COALA	0.16 ± 0.03	0.22 ± 0.03	0.26 ± 0.01	0.76 ± 0.85	0.05 ± 0.64	1.26 ± 0.86
r_Δ	0.05	0.45	**0.14**	-0.16	-0.62	**0.11**
LOLA	0.03 ± 0.01	-0.02 ± 0.06	0.14 ± 0.01	1.71 ± 0.08	0.54 ± 0.03	1.91 ± 0.01
r_Δ	-0.01	-0.04	-0.00	1.42	0.06	0.00

the overall strongest performance across all tasks and significantly outperforms the baselines in Coins and Coop-Mining, highlighting the *effectiveness* of our simple experience-based learning-awareness. Most methods achieve the optimal reward across all metrix games, yet experience-based methods struggle with large variances in ICG and IPD, suggesting that simple-form iterated matrix games are still challenging for this class of methods. Almost all methods suffer from high variance in Cleanup. Figure 5.(a) shows they result in large variance in the frequency of the clean action, which largely impacts the apples' regrowth rate, thereby affecting the rewards.

5.3 LA AGENT EXPLOITS NAIVE COPLAYER

This section demonstrates the learning-aware agents exploit the naive coplayer. Table 3 summarizes the mean episode rewards of learning-aware agents and their performance gap with coplayers at the final iteration. In general, experience-based methods consistently exploit their coplayers and achieve higher rewards, with occasional exceptions of COALA in ICG and IPD. LARL overall achieves the largest performance gap across all tasks. The behavioral patterns in Figure 5.(b) presents the exploitation phenomenon in Cleanup. Except LOLA, each learning-aware agent shows a freeloaded behavior by not cleaning the pollution, while the exploited coplayer needs to clean the pollution to gain rewards.

5.4 IMPORTANCE OF LA ORDER

We explore the utility of the LA-order, by presenting experiments on LARL with higher LA-order. We consider three exploitation settings: (expl1) LA2-vs-LA1 (reusing results from Section 5.3), (expl2) LA3-vs-LA1, and (expl3) LA3-vs-LA2-vs-LA1, where all agents with the same LA-order use the identical configuration. To note the missing LA2 in (expl2), we mark the LA3 agent in (expl2) as $\mathrm{LA}\tilde{3}$.

Figure 6 presents the mean episode reward (first row) and clean frequency (second row) in Cleanup, where each column corresponds to a different setting. As shown in the first and second columns, $\mathrm{LA}\tilde{3}$ in (expl2) demonstrates the same behavior and similar performance as the LA2 in (expl1), who exploits the LA1 coplayer by not cleaning the pollution. As shown in the third column, agents in (expl3) demonstrates tremendously different behavior than agents in (expl2) and (expl1). In particular, the LA3 agent exploits the LA2 by not cleaning the pollution; however, the exploited LA2 preserves the cleaning behavior in its group with LA1, leading LA1 to not clean the pollution either. Despite such behaviors, all agents' achieved rewards still reflects the LA-order hierarchy, yet none of the three agents matches the performance of their counterpart in (expl1) or (expl2). These observations support the claim that LA-orders are *relative*: removing the LA2 from (expl3) shifts the LA3 down to an LA2 agent. They also demonstrate that agents in adjacent LA-order groups *mutually* influence each other's learning dynamics and subsequent behavior. The three cooperation settings also support such claims. See results in Section B.4.

5.5 STRONG AD-HOC GENERALIZATION

Previous sections presents training time results; however, more realistic scenarios requires agents to generalize to unseen coplayers. In this section, we evaluate all learning-aware methods in the ad-hoc setting, where each agent is evaluated against every other mutually unseen agent in cross-play. For each state-based task, we sample an agent from each prior experiment to form a pool. In the pool, we additionally include agents trained with the population-based LARL variant (Algorithm 2), with max LA-order $K = 2$ and population size $|\mathcal{N}_k| = 3$. We evaluate every unorderedpair of agents in the

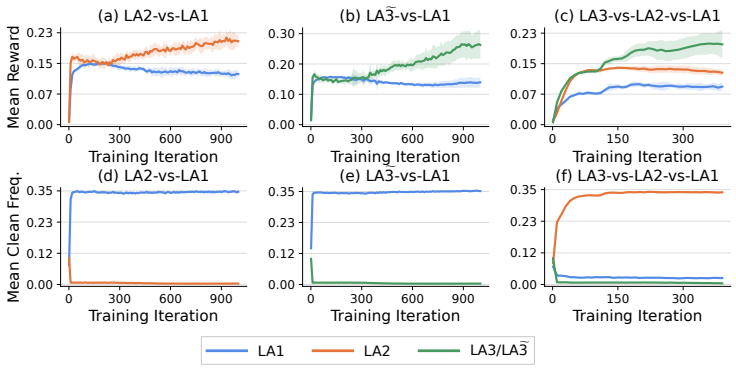


Figure 6: High LA-order agents exploit lower LA-order agents. LA3̃-vs-LA1 means an LA3 agent is directly grouped with an LA1 agent. (a-b,d-e) LA3̃-vs-LA1 setting demonstrates the same behavior as LA2-vs-LA1, suggesting LA-orders are *relative*. (c,f) LA3-vs-LA2-vs-LA1 demonstrates different behavior than LA3̃-vs-LA1, highlighting the mutual influence between adjacent LA-order agents.

pool, exlucluding self-play to conform to the ad-hoc setting. The cross-play matrix Figure 7 indicates that LARL achieves the strongest performance across all coplayers, while the population-based LARL variant does not bring significant difference. More cross-play results can be found in Section B.

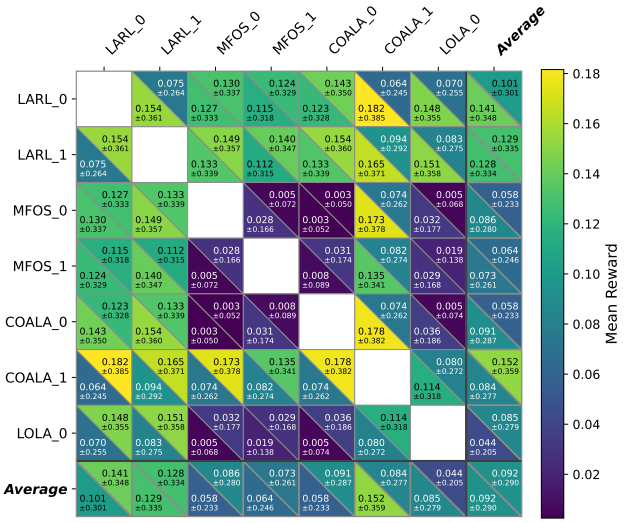


Figure 7: Cross-play results in Cleanup. LARL achieves the strongest performance across all coplayers. LARL_1 is trained by population-based LARL (Algorithm 2).

6 Conclusion and Discussions

In this work, we address the social dilemma challenge in MARL. We propose a multi-agent decision framework that characterizes agents' learning as actions and represents coplayers' learning process as observations. We model an agent's learning awareness as the information about coplayers available to it, and introduce the learning-awareness order over the agent set. This formulation constitutes a basis for understanding the learning dynamics of existing approaches and designing new learning-aware methods. We then introduce LARL, a model-free, decentralized learning algorithm that performs learning-aware RL by learning from the agent's own experience spanning coplayers' policy updates. Empirical studies show that LARL outperforms existing decentralized approaches over well-known sequential social dilemma tasks. Further, LARL exhibits superior adhoc generalization capabilities in evaluation against unseen coplayers. One outstanding limitation lies in its on-policyness Extending support to off-policy learning points to a promising direction for high sample efficiency.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools to polish author-drafted text, including improving grammar, clarity, and phrasing. We did not use generative AI tools for research ideation or execution, drafting substantive sections of the paper, generating synthetic datasets, or proving mathematical claims. Additionally, we used generative AI tools to assist with sourcing and searching for information, surveying relevant literature, and editing and checking code. We reviewed all AI-assisted work: literature-related outputs were verified against the original sources, writing edits were manually reviewed by the authors, and AI-assisted code changes were inspected and tested for correctness. We take responsibility for the final content of this work, including text, claims, code, and artifacts.

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 Agustin Duque, Tim Cooijmans, and Aaron Courville. Loqa: Learning with opponent q-learning awareness. *arXiv preprint arXiv:2405.01035*, 2024.

Maruan Al-Shedivat, Trapit Bansal, Yura Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations*, 2018.

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

Stefano V Albrecht, Filippos Christianos, and Lukas Schäfer. *Multi-agent reinforcement learning: Foundations and modern approaches*. MIT Press, 2024.

Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. *arXiv preprint arXiv:2210.05492*, 2022.

Tyler Becker and Zachary Sunberg. Bridging the gap between partially observable stochastic games and sparse pomdp methods. *arXiv preprint arXiv:2405.18703*, 2024.

Quentin Bertrand, Juan Agustin Duque, Emilio Calvano, and Gauthier Gidel. Self-play q-learners can provably collude in the iterated prisoner's dilemma. In *International Conference on Machine Learning*, pp. 3952–3975. PMLR, 2025.

Michael Bowling and Peter McCracken. Coordination and adaptation in impromptu teams. In *AAAI*, volume 5, pp. 53–58, 2005.

Bryan Randolph Bruns. Names for games: locating $2\times$ 2 games. *Games*, 6(4):495–520, 2015.

Brandon Cui, Hengyuan Hu, Luis Pineda, and Jakob Foerster. K-level reasoning for zero-shot coordination in hanabi. *Advances in Neural Information Processing Systems*, 34:8215–8228, 2021.

Brandon Cui, Andrei Lupu, Samuel Sokota, Hengyuan Hu, David J Wu, and Jakob Nicolaus Foerster. Adversarial diversity in hanabi. In *The Eleventh International Conference on Learning Representations*, 2023.

Jan De Houwer, Dermot Barnes-Holmes, and Agnes Moors. What is learning? on the nature and merits of a functional definition of learning. *Psychonomic bulletin & review*, 20(4):631–642, 2013.

Michael Dennis, Natasha Jaques, Eugene Vinitsky, Alexandre Bayen, Stuart Russell, Andrew Critch, and Sergey Levine. Emergent complexity and zero-shot transfer via unsupervised environment design. *Advances in neural information processing systems*, 33:13049–13061, 2020.

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.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Juan Agustin Duque, Milad Aghajohari, Tim Cooijmans, Razvan Ciuca, Tianyu Zhang, Gauthier Gidel, and Aaron Courville. Advantage alignment algorithms. *arXiv preprint arXiv:2406.14662*, 2024.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric Xing, and Shimon Whiteson. Dice: The infinitely differentiable monte carlo estimator. In *International Conference on Machine Learning*, pp. 1529–1538. PMLR, 2018.

Jakob N Foerster, Richard Y Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. Learning with opponent-learning awareness. *arXiv preprint arXiv:1709.04326*, 2017.

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

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International conference on machine learning*, pp. 1804–1813. PMLR, 2016.

Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. "other-play" for zero-shot coordination. In *International Conference on Machine Learning*, pp. 4399–4410. PMLR, 2020.

Akbir Khan, Timon Willi, Newton Kwan, Andrea Tacchetti, Chris Lu, Edward Grefenstette, Tim Rocktäschel, and Jakob Foerster. Scaling opponent shaping to high dimensional games. *arXiv preprint arXiv:2312.12568*, 2023.

Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76: 201–264, 2023.

Harold W Kuhn. Extensive games and the problem of information. *Contributions to the Theory of Games*, 2(28):193–216, 1953.

Sheldon J Lachman. Learning is a process: Toward an improved definition of learning. *The Journal of psychology*, 131(5):477–480, 1997.

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.

Xiangyu Liu, Hangtian Jia, Ying Wen, Yujing Hu, Yingfeng Chen, Changjie Fan, Zhipeng Hu, and Yaodong Yang. Towards unifying behavioral and response diversity for open-ended learning in zero-sum games. *Advances in Neural Information Processing Systems*, 34:941–952, 2021.

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.

Keane Lucas and Ross E Allen. Any-play: An intrinsic augmentation for zero-shot coordination. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pp. 853–861, 2022.

Andrei Lupu, Brandon Cui, Hengyuan Hu, and Jakob Foerster. Trajectory diversity for zero-shot coordination. In *International conference on machine learning*, pp. 7204–7213. PMLR, 2021.

Kevin R McKee, Joel Z Leibo, Charlie Beattie, and Richard Everett. Quantifying the effects of environment and population diversity in multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 36(1):21, 2022.

Alexander Meulemans, Seijin Kobayashi, Johannes von Oswald, Nino Scherrer, Eric Elmoznino, Blake Richards, Guillaume Lajoie, João Sacramento, et al. Multi-agent cooperation through learning-aware policy gradients. *arXiv preprint arXiv:2410.18636*, 2024.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Côme Fiegel, et al. Nash learning from human feedback. In *Forty-first International Conference on Machine Learning*, 2024.

John F Nash Jr. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

Hadi Nekoei, Xutong Zhao, Janarthanan Rajendran, Miao Liu, and Sarath Chandar. Towards few-shot coordination: Revisiting ad-hoc teamplay challenge in the game of hanabi. In *Conference on Lifelong Learning Agents*, pp. 861–877. PMLR, 2023.

Tianwei Ni, Benjamin Eysenbach, and Ruslan Salakhutdinov. Recurrent model-free rl can be a strong baseline for many pomdps. *arXiv preprint arXiv:2110.05038*, 2021.

Tianwei Ni, Michel Ma, Benjamin Eysenbach, and Pierre-Luc Bacon. When do transformers shine in rl? decoupling memory from credit assignment. *Advances in Neural Information Processing Systems*, 36:50429–50452, 2023.

Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, 53(11):13677–13722, 2023.

Vilfredo Pareto. *Manual of political economy: a critical and variorum edition*. OUP Oxford, 2014.

Jack Parker-Holder, Aldo Pacchiano, Krzysztof M Choromanski, and Stephen J Roberts. Effective diversity in population based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:18050–18062, 2020.

David Premack and Guy Woodruff. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526, 1978.

Neil Rabinowitz, Frank Perbet, Francis Song, Chiyuan Zhang, SM Ali Eslami, and Matthew Botvinick. Machine theory of mind. In *International conference on machine learning*, pp. 4218–4227. PMLR, 2018.

Arrasy Rahman, Elliot Fosong, Ignacio Carlucho, and Stefano V Albrecht. Generating teammates for training robust ad hoc teamwork agents via best-response diversity. *arXiv preprint arXiv:2207.14138*, 2022.

Muhammad Rahman, Jiaxun Cui, and Peter Stone. Minimum coverage sets for training robust ad hoc teamwork agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17523–17530, 2024.

Anatol Rapoport and Albert M Chammah. *Prisoner's dilemma: A study in conflict and cooperation*, volume 165. University of Michigan press, 1965.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sara J Shettleworth. Learning and behavioural ecology. 1984.

Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pp. 1504–1509, 2010.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Adaptive Agent Team, Jakob Bauer, Kate Baumli, Satinder Baveja, Feryal Behbahani, Avishkar Bhoopchand, Nathalie Bradley-Schmieg, Michael Chang, Natalie Clay, Adrian Collister, et al. Human-timescale adaptation in an open-ended task space. *arXiv preprint arXiv:2301.07608*, 2023.

Johannes Treutlein, Michael Dennis, Caspar Oesterheld, and Jakob Foerster. A new formalism, method and open issues for zero-shot coordination. In *International Conference on Machine Learning*, pp. 10413–10423. PMLR, 2021.

John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior, 2nd rev. 1947.

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.

Wako Yoshida, Ray J Dolan, and Karl J Friston. Game theory of mind. *PLoS computational biology*, 4(12):e1000254, 2008.

Rui Zhao, Jinming Song, Yufeng Yuan, Haifeng Hu, Yang Gao, Yi Wu, Zhongqian Sun, and Wei Yang. Maximum entropy population-based training for zero-shot human-ai coordination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6145–6153, 2023.

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.

A Appendix

A.1 Related work

Co-player learning awareness. Theory of mind concerns representing others' mental states, including beliefs, desires, and intentions (Premack & Woodruff, 1978; Yoshida et al., 2008; Rabinowitz et al., 2018); learning awareness instead specifically concerns how their behavior changes through learning. They are connected to meta-RL, as we can treat co-players and their learning dynamics as part of the task distribution. These methods differ in their information requirements and assumptions about co-player learning. LOLA (Foerster et al., 2017) differentiates through an anticipated gradient-ascent update of a naive co-player, with subsequent methods modifying its stability, consistency, or update formulation (Letcher et al., 2019; Willi et al., 2022; Zhao et al., 2022). LOQA (Aghajohari et al., 2024) instead models co-player behavior through estimated action values, while Advantage Alignment (Duque et al., 2024) derives related updates using both agents' advantages. Such methods require access to, or estimates of, co-player update rules or value information. Other methods learn to account for co-player adaptation through interaction, without direct access to co-player learning internals. M-FOS (Lu et al., 2022) formulates a meta-game in which policy choices influence co-player learning across episodes, enabling model-free optimization of these long-horizon effects. Subsequent work extends this approach to high-dimensional games (Khan et al., 2023). COALA (Meulemans et al., 2024) further studies how learning awareness can support cooperation among self-interested agents, including populations containing both naive and learning-aware agents. A related line of work, commonly termed "opponent shaping", optimizes an agent's behavior while accounting for its influence on co-players' learning and subsequent behavior (Foerster et al., 2017; Khan et al., 2023).

Meta-RL. Meta-RL enables rapid adaptation in continuous control (Al-Shedivat et al., 2018) and open-ended tasks (Team et al., 2023), suggesting a natural application in multi-agent system on adaptation to teammates' strategies and learning dynamics. Optimization-based methods adapt through explicit parameter updates (Finn et al., 2017; Al-Shedivat et al., 2018), whereas context-based methods encode adaptation in recurrent or sequence models conditioned on interaction histories (Ni et al., 2021; 2023). For example, RL^2 (Duan et al., 2016) adapts across episodes through its recurrent state without test-time parameter updates.

Ad-hoc teamwork. A large body of MARL research studies cooperation (Oroojlooy & Hajinezhad, 2023; Du et al., 2023; Albrecht et al., 2024), with ad-hoc teamwork (AHT) emerging as a prominent setting for real-world applications, in which an agent must collaborate effectively with previously unseen teammates (Stone et al., 2010). Related work has also studied AHT under alternative names, including zero-shot coordination (ZSC) (Hu et al., 2020; Lupu et al., 2021; Treutlein et al., 2021; Cui et al., 2021; Kirk et al., 2023), which often assumes that partners employ the same algorithm (Nekoei et al., 2023), and coordination in impromptu teams (Bowling & McCracken, 2005). A key principle for ad-hoc generalization is exposing agents to sufficiently *diverse* co-players during training. This diversity primarily concerns *behavioral diversity*, which measures differences in the agents' behaviors themselves (e.g., divergences between policies (Parker-Holder et al., 2020; McKee et al., 2022; Zhao et al., 2023), skill latents (Lucas & Allen, 2022), state-action occupancy measures (Liu et al., 2021), or trajectories (Lupu et al., 2021)), and *response diversity*, which measures how differently other agents behave when interacting with them, such as inducing distinct best responses (Liu et al., 2021; Rahman et al., 2022; 2024). Aside from directly optimizing diversity metrics, empirical methods have also developed diverse populations through mechanisms such as fictitious co-play (Munos et al., 2024), curriculum learning (Dennis et al., 2020), adversarial training (Cui et al., 2023), and introducing inductive bias such as symmetry (Hu et al., 2020; Bakhtin et al., 2022).

Compare LARL to prior experience-based methods. LARL differs from prior experience-based methods, MFOS (Lu et al., 2022), and COALA (Meulemans et al., 2024), in several key aspects. Prior methods leverages a batched coplayer strategy, which corresponds each learning-aware agent's trajectory (called the meta trajectory) to the whole parallel batch of rollouts with naive agents. Hence, at return evaluation of the meta agent, they normalize the future episodes' returns at each inner episode boundary across the batch dimension. COALA additionally normalizes the current episode returns over the batches to balance the contributions from the current and the future. We instead follow a simpler view that the parallel rollouts are all outer episodes. State-action distribution shifts

at each inner episode boundary can be reflected in the trajectories, upon which the learning-aware agent optimizes its policy. Moreover, LARL supports LA2+, which builds a hierarchy of learning-awareness over agents with different LA-orders using a fixed batch size, whereas prior methods only support LA2, i.e., learning-awareness of only naive agents. To support higher LA-orders, their batched coplayer strategy would encounter scalability issue as it requires an additional batch of meta trajectories. Furthermore, our meta-learning approach performs gradient updates sequentially over all groups, while COALA averages the gradients across groups for the same agent.

A.2 EXAMPLES

Example 1 of outer POSG. Consider an exemplary inner POSG $\mathtt{M}$ with episode length $\mathtt{H} = 4$, where agents $\mathcal{N}$ include (1) online actor-critic (Sutton et al., 1998), (2) PPO (Schulman et al., 2017), and (3) LOLA-DICE with one-step look-ahead (Foerster et al., 2018), whose parameter-update intervals are $1, 2, 2$ inner timesteps, respectively. All policies are parameterized by MLPs. In the corresponding outer POSG $\tilde{M}$, the outer **state** $\tilde{s}$ consists of intra-agent and inter-agent components. The former includes all agents' network parameters, optimizer states; the latter includes the inner environment features, transitions, and step counters. An outer **action** $\tilde{a}^i$ is a learner-state-updating function defined by an inner agent's learning algorithm. At scheduled updates, agent 1 updates its actor and critic using one-step TD error; agent 2 estimates advantages and optimizes PPO's clipped surrogate and value losses; agent 3 differentiates its return through one hypothetical policy-gradient update of each coplayer. An agent-specific outer **observation** $\tilde{o}^i$ could take different forms depending on assumptions of observability. Here, each agent observes its own learner state and latest inner transition. Agent 3 additionally accesses coplayer policy parameters and an auxiliary simulator providing joint observations, actions, and rewards for rollouts before and after hypothetical updates.

Example 2 of outer POSG. Consider an exemplary inner POSG $\mathtt{M}$ with episode length $\mathtt{H} = 4$, where agents $\mathcal{N}$ include (0) random policy, (1) online actor-critic (Sutton et al., 1998), and (2) PPO (Schulman et al., 2017) equipped with an RNN, whose update intervals are $0, 1, 4$ inner timesteps respectively, as shown by agents' rollouts in Figure 2. Learnable agents (1~2) are parameterized by neural networks and update parameters with Adam optimizers. In the corresponding outer POSG $\tilde{M}$, the outer **state** $\tilde{s}$ consists of intra-agent and inter-agent components. The former includes all inner agents' network parameters, optimizer states (e.g., momentums), agent 2's RNN hidden states, etc, and the latter includes shared inner environment instances, per-timestep inner transitions, step counters, etc. An outer **action** $\tilde{a}^i$ is a learner-state-updating function defined by an inner agent's learning algorithm. In particular, if the inner timestep coincides with an agent's update interval, the agent's outer action is the update rule of its learning algorithm; otherwise, the outer action is sequence-model-based in-context learning for agent 2. An agent-specific outer **observation** $\tilde{o}^i$ could take different forms depending on assumptions of observability. A natural choice is the corresponding inner agent's transitions $\{(\mathtt{o}^i_\mathtt{t}, \mathtt{a}^i_\mathtt{t}, \mathtt{r}^i_\mathtt{t}, \mathtt{o}^i_{\mathtt{t}+1})\}$ required by its learning algorithm for an update. Such an observation does not provide the outer agents with more information than their inner counterparts already access; however, it allows sparser updates – i.e., one that updates the policy at a lower frequency in terms of inner timesteps – to *naturally* observe histories that reflects the learning dynamics of coplayers with more frequent updates, without explicit modeling of other agents' learning dynamics. For instance, agent 2's episode-length trajectory contains transitions before and after agent 1's each update, hence agent 2's inner PPO update naturally conditions on the non-stationarity introduced by agent 1's learning dynamics.

Example 1 of LA-order. Consider the aforementioned agents in example 1: (1) online actor-critic, (2) PPO, and (3) LOLA-DiCE with one-step look-ahead. Agent 2 conditions its update on transitions collected before and after an update of agent 1, while agent 3 additionally uses coplayer policy parameters and explicitly modeled updates to account for agents 1 and 2's learning. Thus, the agents have LA-orders $1, 2, 3$, respectively. The hierarchy requires the stated information comparisons, and algorithm names alone do not establish it. Restricting the hierarchy to agents $2, 3$ gives LA-orders $1, 2$, illustrating that numerical orders depend on the selected subset.

Example 2 of LA-order. Suppose we have the following agents: (0) fixed random policy, (1) online actor-critic, (2) PPO with an RNN, with (3) PPO with a smaller network, (4) LOLA-DICE (Foerster et al., 2018) with $k = 2$, (5) LOLA-DICE with $k = 3$, (6) AdAlign (Duque et al., 2024), (7)

DQN (Mnih et al., 2013) with update interval 2, and (8) RL^2 (Duan et al., 2016) with a causal transformer and update interval 12. The following ranks require the stated information relationships:

(a) Under Assumption 3.1, the hierarchy $0 \prec 1 \prec (2 \sim 3) \prec 4 \prec 5$ (agent identifiers) gives levels $0, 1, 2, 2, 3, 4$. Agents 2 and 3 share a level despite different network sizes when their information-type comparisons coincide.

(b) Restricting this hierarchy to $[2, 3, 4, 5]$, with comparisons preserved, gives levels $1, 1, 2, 3$.

(c) Agents $[3, 4, 8]$ have levels $1, 2, 3$ if Assumption 3.1 holds with $3 \prec 4 \prec 8$.

(d) Agents $[1, 7]$ are incomparable if $y_{17} \not\subseteq y_{71}$ and $y_{71} \not\subseteq y_{17}$; uniform replay sampling alone does not establish this.

(e) Agents $[4, 6]$ are incomparable if y_{46} and y_{64} are nonempty and disjoint.

Cases (d) and (e) violate totality, so their pairwise comparisons remain defined but the numerical levels above do not.

A.3 Notation and Terminologies

We summarize important notations in Table 4, and terminologies we used in Table 5.

A.4 Problem Setting of Iterated Prisoner's Dilemma

Prisoner's dilemma is a classic social dilemma game, where self-interested naive agents tend to choose the jointly suboptimal defect-defect strategy. Bertrand et al. (2025) demonstrate that self-play Q-learning with optimistic initialization can provably learn the cooperative Pavlov policy (win-stay, lose-switch) in IPD. However, it is unclear how this result generalizes to other algorithmic classes, and performing optimistic initialization in the deep RL regime is challenging. To understand the underlying learning mechanism, we consider the self-interested one-step memory agent that observes its own and coplayer's actions at the past timestep, and optimizes its own expected return via direct gradient ascent.

We consider the two-player IPD, with actions cooperate (C) and defect (D). Both players use the same memory-one cooperation policy,

$$a_k = \sigma(\theta_k), \qquad \sigma(x) = \frac{1}{1+e^{-x}}, \qquad \theta \in \mathbb{R}^5, \quad k \in \{0, 1, 2, 3, 4\}.$$

The entries correspond to the initial decision and previous outcomes CC, CD, DC, DD, respectively, with each player's own action first. For vectors, σ acts componentwise.

In the global outcome order (CC, CD, DC, DD), with player 0's action first, the reward vectors are

$$r^0 = (R, S, T, D)^\top, \qquad r^1 = (R, T, S, D)^\top,$$

where $R, S, T, D \in \mathbb{R}$ and $\top$ denotes transpose.

Assumption A.1 (Game and policies)**.** (i) Actions are sampled independently at the initial decision and conditionally independently given the previous outcome thereafter.

(ii) The rewards satisfy

$$T > R > D > S, \qquad D < M := \frac{S+T}{2} < R.$$

(iii) The discount factor $\gamma \in (0, 1)$ is fixed.

To define each player's partial gradient, write the two policy arguments separately as $a, b \in [0, 1]^5$. For $x, y \in [0, 1]$, define the row vector

$$q(x, y) = \big[xy, x(1-y), (1-x)y, (1-x)(1-y)\big].$$

The first-round outcome distribution and transition matrix are

$$p_0(a, b) = q(a_0, b_0), \qquad P(a, b) = \begin{pmatrix} q(a_1, b_1) \\ q(a_2, b_3) \\ q(a_3, b_2) \\ q(a_4, b_4) \end{pmatrix}.$$

The mixed-outcome rows account for the players' respective perspectives.

The unnormalized discounted state occupancy is

$$d_\gamma(a,b) = \sum_{n=0}^{\infty} \gamma^n p_0(a,b) P(a,b)^n = p_0(a,b)\big(I_4 - \gamma P(a,b)\big)^{-1},$$

where I_4 is the 4×4 identity matrix and $n = 0$ denotes the first-round outcome. Under the shared policy, write $d_\gamma(a) := d_\gamma(a,a)$. Each player's discounted return is

$$J^i(\theta^0,\theta^1) = d_\gamma\big(\sigma(\theta^0),\sigma(\theta^1)\big) r^i, \qquad i \in \{0,1\}, \quad \theta^0,\theta^1 \in \mathbb{R}^5.$$

Both players use the same parameter values throughout learning. Each agent's update uses the partial derivative with respect to its own parameter argument, evaluated at $\theta^0 = \theta^1 = \theta$. The opponent's policy affects the return but is treated as constant when computing this derivative. Symmetry gives the common update direction

$$g(\theta) := \nabla_{\theta^0} J^0(\theta^0,\theta^1)\big|_{\theta^0=\theta^1=\theta} = \nabla_{\theta^1} J^1(\theta^0,\theta^1)\big|_{\theta^0=\theta^1=\theta}.$$

Assumption A.2 (Initialization)**.** Both players initialize their parameters independently from a Gaussian distribution, $\theta^{i,(0)} \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{init}}^2 I_5)$ for $i \in \{0,1\}$, and use exact simultaneous updates:

$$\theta^{i,(t+1)} = \theta^{i,(t)} + \alpha g^i\big(\theta^{0,(t)},\theta^{1,(t)}\big), \qquad a^{i,(t)} = \sigma\big(\theta^{i,(t)}\big).$$

A.5 Algorithms

Algorithm 2: Population-based LARL for LA$[K]$ agents

input : num_train_iterations L, LAk agents' grouping scheduler LambSched_k, Sampler without replacement Rand, population_size_LAk $N_{\text{ppl}k}$, num_sampled_groups_LAk C_k, agents $\bigcup_{k\in[K]} \mathcal{N}_k$, where $|\mathcal{N}_k| = N_{\text{ppl}k}$

output : updated agents $\bigcup_{k\in[K]} \mathcal{N}_k$,

for *train iteration* $l = 1,\dots,L$ **do**
  for LA-*order* $k \in [K]$ **do**
    // Sample (up to) C_k groups of N agents, where each agent's LA order ($\in \{k, k-1\}$) depends on LambSched_k
    updated LAk & LA$(k-1)$ agent combinations $\mathcal{D}_k \leftarrow \emptyset$
    for *group idx* $c = 1,\dots,C_k$ **do**
      at least one LAk agent per group $i_k \leftarrow \text{Rand}(\mathcal{N}_k)$
      agent group $\mathcal{C} \leftarrow \{i_k\}$
      $\lambda_k \leftarrow \text{LambSched}_k(l)$
      // n^{i_k} denotes i_k's agent-in-group idx
      for *agent idx* $n \in [N] \setminus \{n^{i_k}\}$ **do**
        if $\lambda_k < \text{Uniform}[0,1]$ **then**
          $\mathcal{C} \leftarrow \mathcal{C} \cup \text{Rand}(\mathcal{N}_k^n \setminus \{i_k\})$
        else
          $\mathcal{C} \leftarrow \mathcal{C} \cup \text{Rand}(\mathcal{N}_{(k-1)}^n)$
      if $\mathcal{C} \in \mathcal{D}_k$ **then continue**
      $\mathcal{D}_k \leftarrow \mathcal{D}_k \cup \{\mathcal{C}\}$
      rollout and update $\mathcal{C}$ policies (Algorithm 3)

B Additional Results

B.1 Detailed description of tasks

Cleanup Introduced by Meulemans et al. (2024), this task is a simplified 2-player version of the original Cleanup task (Agapiou et al., 2022). Agents gain rewards by collecting apples in a grid. The apples' growth rate is inversely related to the cleanliness of a nearby river, which accumulates

Algorithm 3: Rollout and update $\mathcal{C}$

input : inner environment M, agents $\mathcal{C}$, LAk agent's number of inner episodes in one outer episode T_k
output : updated agents $\mathcal{C}$
// $\mathcal{C}_{k'} = \{i \in \mathcal{C} : \text{LA}^i = k'\}$
for *inner episode* $e = 1, \ldots, T_k$ **do**
  rollout policies $\text{M}(\mathcal{C})$
  if $e \bmod T_{(k-1)} = 0$ **then**
    update $\mathcal{C}_{(k-1)}$ every $T_{(k-1)}$ inner episodes
update $\mathcal{C}_k$ every T_k inner episodes

pollution constantly. As cleaning the pollution requires a physical distance to the apple orchard, to maintain continual apple growth, agents need to make effortful contributions, trading off between collecting and cleaning. Showing similar incentives to the Chicken Game, this task models the *tragedy of the commons* (Hardin, 1968) dilemma.

Coins First introduced by Lerer & Peysakhovich (2017), this task shares similar properties to the Prisoner's Dilemma. Two agents in red and blue colors move around a wrap-around grid and collect red or blue coins. An agent receives a reward of +1 for collecting any coin. If an agent collects a coin matching its coplayer's color, the coplayer receives a reward of -2. Hence if both agents collect coins greedily without regard to color, their expected reward are both 0.

Coop-mining This task is a simplified 2-player version of the original task (Agapiou et al., 2022). Two agents mine gold or iron ore, which are randomly spawned in empty spaces. Iron ore can be mined individually and gives the miner a reward of +1, while gold ore requires group coordination and gives both miners a reward of +8. Greedy agents would tend to mine iron ore more. This task shares common properties to the Stag Hunt.

B.2 Cooperation results

Table 6 summarizes the mean episode rewards at the final iteration for Figure 4.

In Coins, all methods except LARL fluctuate around 0. Figure 9.(a) also shows these methods collect coins of different colors with similar probabilities, suggesting that they fail to learn a cooperative strategy. Most methods reach steady convergence in Coop-Mining and ISH. This observation suggests that the Stag Hunt domain may not pose a significant learning challenge. Although both joint policies CC and DD are Nash Equilibria, agents tend to achieve the Pareto-dominated CC.

B.3 LA agent exploits naive coplayer

Figure 8 shows the mean episode rewards for learning-aware agents and their respective naive coplayers, plotted in solid and dashed lines, respectively.

In Coins, on the other hand, despite large performance gap, Figure 9.(b) shows that all agents collect coins regardless of color, indicating that the learning-aware agents learn the exploiting policy that collects more coins rather than coins of their own color. All methods fail to exploit their coplayers in Coop-Mining and ISH, espeically in Coop-Mining, where the learning-aware agents and their coplayers achieve almost identical performance. This phenomenon corroborates the claim that ISH may not pose a significant challenge.

B.4 Importance of LA order

We additionally include the same experiments on IPD, with quantitative results illstrated in Figure 14 and Figure 13, and qualitative results shown in Figure 16 and Figure 15. These empirical evidence further supports the our claims on the importance of the LA-order.

Similarly, we consider three cooperation settings corresponding to the three exploitation settings, named (coop1), (coop2), and (coop3), respectively. Figure 12 presents the corresponding mean episode rewards and mean clean frequency in Cleanup. Likewise, $\text{LA}\tilde{3}$ in (coop2) exhibit the same

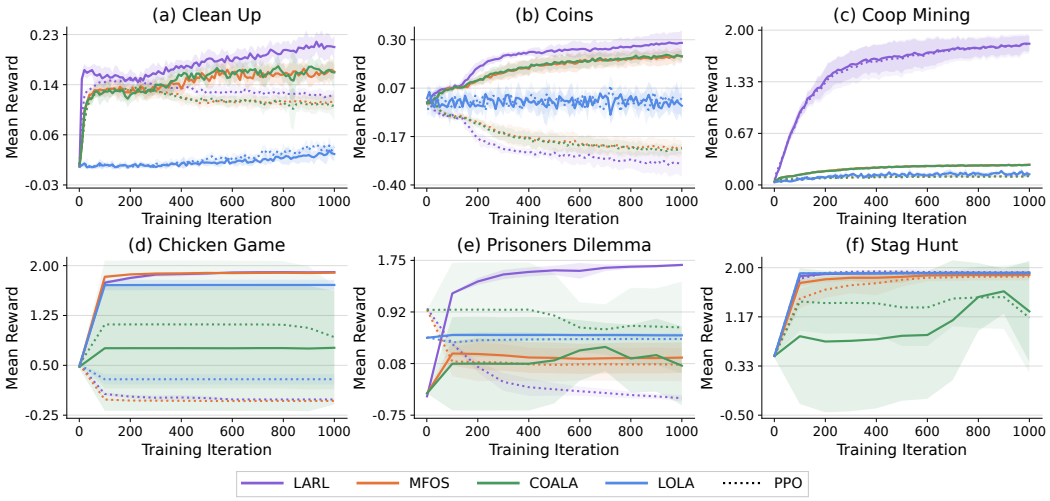


Figure 8: Experience-based learning-aware methods exploit their naive coplayer. LARL outperforms other learning-aware methods in state-based tasks.

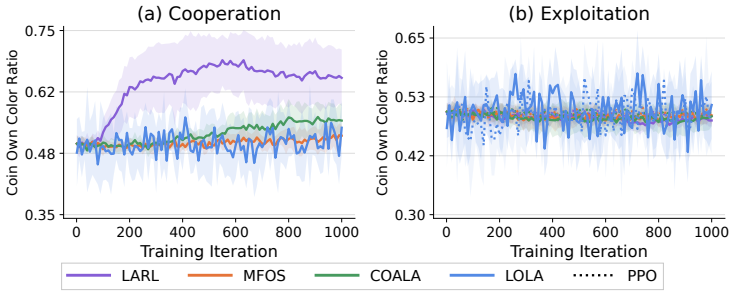


Figure 9: Probability of collecting coins of an agent's own color. (a) corresponds to Figure 4.(b); (b) corresponds to Figure 8.(b). LARL tends to collect coins matching its color more frequently.

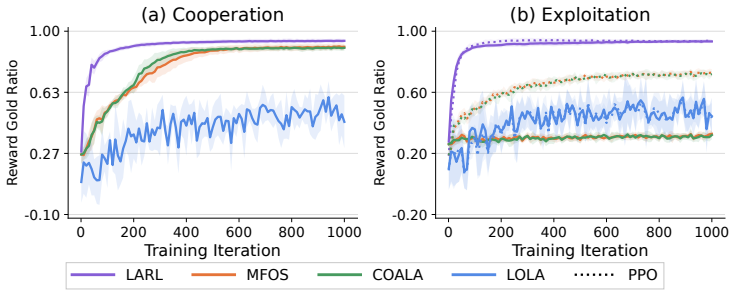


Figure 10: Ratio of reward from mining gold ore.

behavior and performance as LA2 in (expl1), who actively clean the pollution and continually gain rewards from collecting apples. Both LA2 and LA3 in (coop3) tend to actively clean the pollution – with LA2 decreasing its frequency while LA3 increasing it through intra-play. Results from the cooperation settings corroborate the the relativeness of LA-order and mutual influences on learning dynamics.

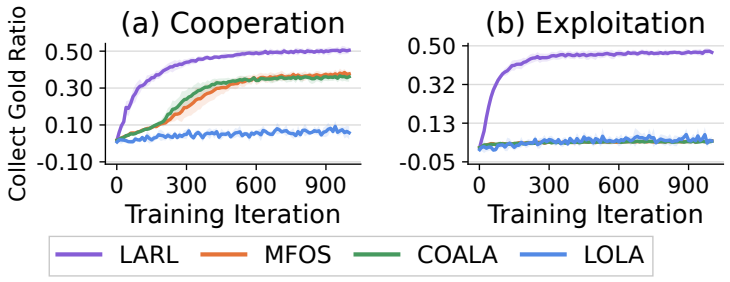


Figure 11: Ratio of collecting gold ore.

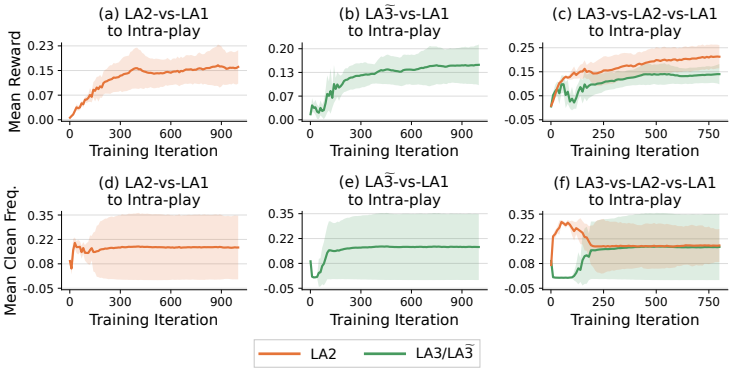


Figure 12: High LA-order agents exploit lower LA-order agents, then high LA-order agents are grouped to achieve cooperation. Similar comparisons as in Figure 6.

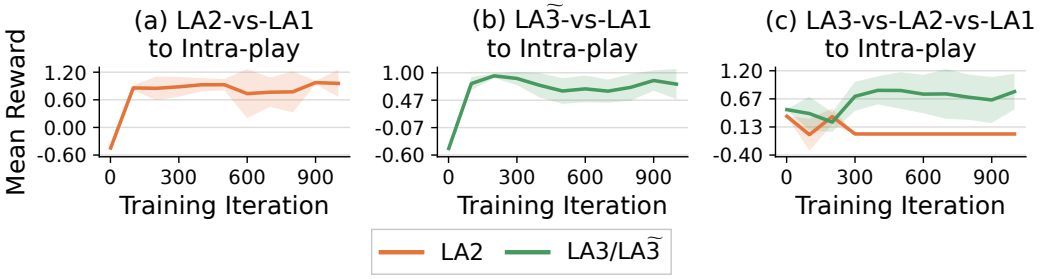


Figure 13: In Prisoner's Dilemma, high LA-order agents exploit lower LA-order agents, then high LA-order agents are grouped to achieve cooperation.

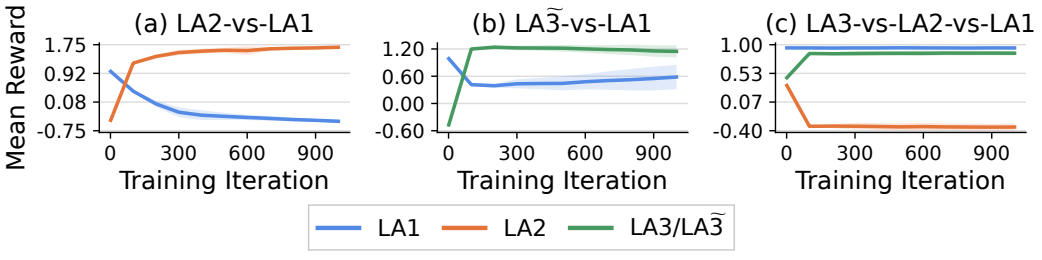


Figure 14: High LA-order agents exploit lower LA-order agents in Prisoner's Dilemma.

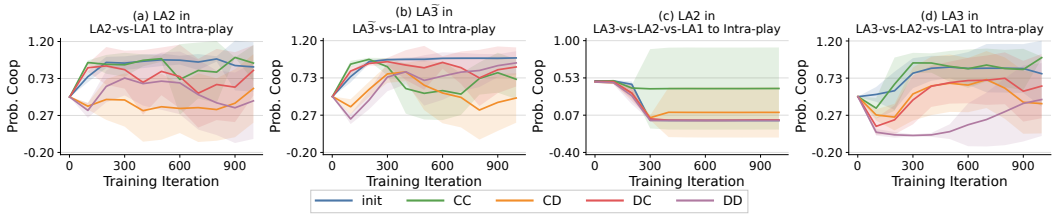


Figure 15: In Prisoner's Dilemma, high LA-order agents exploit lower LA-order agents, then high LA-order agents are grouped to achieve cooperation.

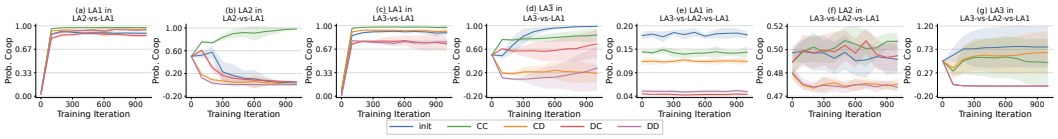


Figure 16: High LA-order agents exploit lower LA-order agents in Prisoner's Dilemma.

B.5 AD-HOC GENERALIZATION

Figure 18 and Figure 17 also confirms LARL strength. These results highlight LARL's outstanding effectiveness in generalization.

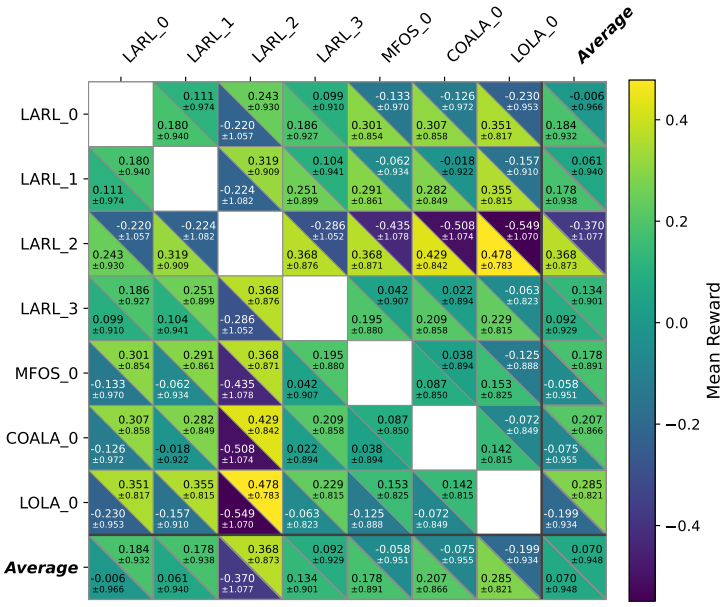


Figure 17: Cross-play results in Coins. LARL achieves the strongest performance across all coplayers. LARL_2 and LARL_3 are trained by population-based LARL Algorithm 2.

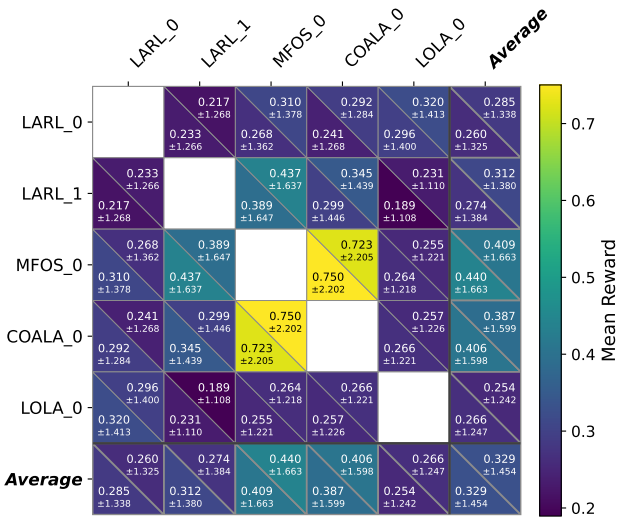


Figure 18: Cross-play results in Coop-Mining. MFOS and COALA achieves the strongest performance, and LARL's results are not significantly different from them. LARL_2 and LARL_3 are trained by population-based LARL Algorithm 2.

Table 4: Notation overview.

Notation	Meaning	Definition
N	number of agents	
Inner POSG		
$\mathtt{M}$	inner POSG	$\mathtt{M} = (\mathcal{N}, \mathcal{S}, \mathcal{A}, \Omega, \mathcal{O}, \mathtt{P}, \mathtt{R}, \mu, \gamma, \mathtt{H})$
$\mathcal{N}$	Set of N agents	
$\mathcal{S}$	Set of states	
$\mathtt{s}$	A state	$\mathtt{s} \in \mathcal{S}$
$\mathcal{A}^i$	Agent i's set of actions	
$\mathtt{a}^i$	An agent i's action	$\mathtt{a}^i \in \mathcal{A}^i$
$\mathtt{P}$	Markovian transition function	$\mathtt{P}(\mathtt{s}_{\mathtt{t}+1} \vert \mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to [0,1]$
$\mathcal{O}^i$	Agent i's set of observations	
$\mathtt{o}^i$	An agent i's observation	$\mathtt{o}^i \in \mathcal{O}^i$
Ω^i	Agent i's observation function	$\Omega^i(\mathtt{o}^i_\mathtt{t} \vert \mathtt{s}_{\mathtt{t}-1}, \mathtt{a}_{\mathtt{t}-1}, \mathtt{s}_\mathtt{t}) : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \times \mathcal{O}^i \to [0,1]$
$\mathtt{R}^i$	Agent i's reward function	$\mathtt{R}^i(\mathtt{r}^i_\mathtt{t} \vert \mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t}) : \mathcal{S} \times \mathcal{A} \times \mathbb{R} \to [0,1]$
$\mathtt{r}^i$	An agent i's reward	
μ	Initial state distribution	$\mu : \mathcal{S} \to [0,1]$
γ	Discount factor	$\gamma \in [0,1]$
$\mathtt{H}$	Horizon	$\mathtt{H} \in \mathbb{N}^+ \cup \{+\infty\}$
$\mathtt{t}$	Timestep	$\mathtt{t} \in \{0, \dots, \mathtt{H}-1\}$
ν	Occupancy measure	
$\mathtt{G}^i$	Agent i's return	$\mathtt{G}^i = \sum_{\mathtt{t}=0}^{\mathtt{H}} \gamma^\mathtt{t} \mathtt{r}^i(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})$
$\mathcal{H}^i$	Agent i's set of observation-action trajectories	
τ^i	Agent i's observation-action trajectory	$\tau^i_\mathtt{t} \in \mathcal{H}^i_\mathtt{t} = (\mathcal{O}^i_{0:\mathtt{t}}, \mathcal{A}^i_{0:\mathtt{t}-1})$
π^i	Agent i's policy	$\pi^i_\mathtt{t}(\cdot \vert \tau^i_\mathtt{t}) : \mathcal{H}^i \times \mathcal{A}^i \to [0,1]$
Θ^i	Agent i's set of policy parameters	
θ^i	Agent i's policy parameter	$\theta^i \in \Theta^i$
$\mathtt{J}^i$	Agent i's objective function	$\mathtt{J}^i(\theta^i, \theta^{-i}) = \mathbb{E}_{\tau \sim \pi_\theta}[\sum_{\mathtt{t}=0}^{\mathtt{H}} \gamma^\mathtt{t} \mathtt{r}^i(\mathtt{s}_\mathtt{t}, \mathtt{a}_\mathtt{t})]$
Outer POSG		
$\tilde{M}$	outer POSG	$\tilde{M} = (\tilde{\mathcal{N}}, \tilde{\mathcal{S}}, \tilde{\mathcal{A}}, \tilde{\Omega}, \tilde{O}, \tilde{P}, \tilde{R}, \tilde{\mu}, \tilde{\gamma}, \tilde{H})$
$\tilde{\mathcal{N}}$	Set of N outer agents	
$\tilde{\mathcal{I}}$	Subset of outer agents	$\tilde{\mathcal{I}} \subseteq \tilde{\mathcal{N}}$
$\tilde{\mathcal{S}}$	Set of outer states	
$\tilde{s}$	An outer state	$\tilde{s} \in \tilde{\mathcal{S}}$
$\tilde{\mathcal{A}}^i$	Agent i's set of outer actions	
$\tilde{a}^i$	An agent i's outer action	$\tilde{a}^i \in \tilde{\mathcal{A}}^i$
$\tilde{P}$	Markovian transition function	$\tilde{P}(\tilde{s}_{\tilde{t}+1} \vert \tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \tilde{\mathcal{S}} \to [0,1]$
$\tilde{O}^i$	Agent i's set of outer observations	
$\tilde{o}^i$	An agent i's outer observation	$\tilde{o}^i \in \tilde{O}^i$
$\tilde{\Omega}^i$	Agent i's outer observation function	$\tilde{\Omega}^i(\tilde{o}^i_{\tilde{t}} \vert \tilde{s}_{\tilde{t}-1}, \tilde{a}_{\tilde{t}-1}, \tilde{s}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \tilde{\mathcal{S}} \times \tilde{O}^i \to [0,1]$
$\tilde{O}^{i,j}$	Agent i's set of outer observations of agent j	$\tilde{O}^{i,j} \subseteq \tilde{O}^i$
$\tilde{o}^{i,j}$	An agent i's outer observation of agent j	$\tilde{o}^{i,j} \in \tilde{O}^{i,j}$
$\tilde{\Omega}^{i,j}$	Agent i's outer observation function of agent j	$\tilde{\Omega}^{i,j}(\tilde{o}^{i,j}_{\tilde{t}} \vert \tilde{s}_{\tilde{t}-1}, \tilde{a}^j_{\tilde{t}-1}, \tilde{s}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}}^j \times \tilde{\mathcal{S}} \times \tilde{O}^{i,j} \to [0,1]$
$\tilde{R}^i$	Agent i's outer reward function	$\tilde{R}^i(\tilde{r}^i_{\tilde{t}} \vert \tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}}) : \tilde{\mathcal{S}} \times \tilde{\mathcal{A}} \times \mathbb{R} \to [0,1]$
$\tilde{r}^i$	An agent i's outer reward	
$\tilde{\mu}$	Initial outer state distribution	$\tilde{\mu} : \tilde{\mathcal{S}} \to [0,1]$
$\tilde{\gamma}$	Outer discount factor	$\tilde{\gamma} \in [0,1]$
$\tilde{H}$	Outer horizon	$\tilde{H} \in \mathbb{N}^+ \cup \{+\infty\}$
$\tilde{t}$	Outer timestep	$\tilde{t} \in \{0, \dots, \tilde{H}-1\}$
$\tilde{\nu}$	Outer occupancy measure	
$\tilde{G}^i$	Agent i's outer return	$\tilde{G}^i = \sum_{\tilde{t}=0}^{\tilde{H}} \tilde{\gamma}^{\tilde{t}} \tilde{r}^i(\tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}})$
$\tilde{\mathcal{H}}^i$	Agent i's set of outer observation trajectories	
$\tilde{\tau}^i$	Agent i's outer observation trajectory	$\tilde{\tau}^i_{\tilde{t}} \in \tilde{\mathcal{H}}^i_{\tilde{t}} = (\tilde{O}^i_{0:\tilde{t}})$
$\tilde{\mathcal{Y}}$	Agent i's set of outer trajectories of agent j	
$\tilde{y}^{i,j}$	Agent i's outer trajectory of agent j	$\tilde{y}^{i,j}_{\tilde{t}} \coloneqq \{\tilde{o}^{i,j}_{\tilde{t}'}\}_{\tilde{t}' \in [\tilde{t}]}$
$\tilde{\pi}^i$	Agent i's outer policy	$\tilde{\pi}^i_{\tilde{t}}(\cdot \vert \tilde{\tau}^i_{\tilde{t}}) : \tilde{\mathcal{H}}^i \times \tilde{\mathcal{A}}^i \to [0,1]$
$\tilde{\Phi}^i$	Agent i's set of outer policy parameters	
$\tilde{\phi}^i$	Agent i's outer policy parameter	$\tilde{\phi}^i \in \tilde{\Phi}^i$
$\tilde{J}^i$	Agent i's outer objective function	$\tilde{J}^i(\tilde{\phi}^i, \tilde{\phi}^{-i}) = \mathbb{E}_{\tilde{\tau} \sim \tilde{\pi}_{\tilde{\phi}}}[\sum_{\tilde{t}=0}^{\tilde{H}} \tilde{\gamma}^{\tilde{t}} \tilde{r}^i(\tilde{s}_{\tilde{t}}, \tilde{a}_{\tilde{t}})]$

Table 5: Synonymous and related terminologies in reinforcement learning (RL) and game theory (GT), partially adapted from Albrecht et al. (2024). In this work, we mainly use RL terms, except for concepts better-established in GT.

RL term x	GT term x'	Remarks
MDP	one-player SG	stochastic game (SG) or Markov game (MG)
POMDP	one-player POSG	
Dec-MDP	cooperative SG	Cooperative means common-payoff.
Dec-POMDP	cooperative POSG	
environment	game	
reward	payoff, utility	
agent	player	We use "coplayer" to indicate an agent besides the ego agent in the environment. In general-sum games, we avoid commonly-used terms that may imply cooperation (e.g., cooperator, partner, teammate, collaborator) or competition (e.g., competitor, opponent), unless restrictions on rewards are explicitly stated otherwise.
policy	strategy	
deterministic x	pure x'	E.g., deterministic policy/pure strategy, deterministic/pure Nash equilibrium.
stochastic, probabilistic x	mixed x'	E.g., stochastic policy/mixed strategy, probabilistic/mixed Nash equilibrium.
joint x	x' profile	x/x' is a tuple with one element per agent/player, e.g., joint reward/payoff profile, joint policy/strategy profile.

Table 6: Learning-aware methods achieve cooperation. LARL reaches the strongest overall performance. In each task, the best performance is marked in **bold**, and algorithms that are not significantly different from the best performing algorithm are denoted by *.

Approach	Clean Up	Coins	Coop Mining	Chicken Game	Prisoners Dilemma	Stag Hunt
LARL	$\mathbf{0.16 \pm 0.05}$	$\mathbf{0.17 \pm 0.08}$	$\mathbf{1.91 \pm 0.09}$	$\mathbf{0.99 \pm 0.43}$	$\mathbf{0.96 \pm 0.29}$	$2.00 \pm 0.00^\star$
MFOS	$0.12 \pm 0.08^\star$	0.02 ± 0.03	0.76 ± 0.03	$0.98 \pm 0.82^\star$	$0.55 \pm 0.49^\star$	$\mathbf{2.00 \pm 0.00}$
COALA	$0.14 \pm 0.06^\star$	0.04 ± 0.05	0.72 ± 0.09	$0.96 \pm 0.93^\star$	$0.47 \pm 0.48^\star$	$1.60 \pm 0.52^\star$
LOLA	0.04 ± 0.01	0.02 ± 0.06	0.14 ± 0.04	$0.98 \pm 0.04^\star$	$0.90 \pm 0.02^\star$	1.94 ± 0.02