A ENVIRONMENT CONFIGURATIONS

A.1 Iterated Prisoner's Dilemma (IPD)

A.1.1 Matching Algorithm. At each instance of the IPD, agent i is given a counterpart, j to play the game in Table 1. Agent pairings are assigned subject to ν, the probability that the counterpart is a teammate. In our analysis, we experiment with three different environments with $|\mathcal{T}| = 5$ teams of five agents each. When $\nu = 0.06$ agents are four times more likely to receive a counterpart from another team. When $\nu = 0.2$, a counterpart from any of the five teams has equal likelihood. When $\nu = 0.5$, agents are four times more likely to receive a counterpart from their own team. Each episode, we construct N pairings by matching each agent $i \in N$ to a counterpart $j \in N \setminus \{i\}$ sampled subject to ν. Each agent observes the team their counterpart belongs to through a numerical signal $s_i \in S$, but not their actual individual identity.

	Cooperate	Defect
Cooperate	$b-c, b-c$	$-c, b$
Defect	$b, -c$	0, 0

Table 1: An example of the Prisoner's Dilemma with the costs (c) and benefits (b) of cooperating ($b > c > 0$).

A.1.2 IPD RL Algorithm. For each round of the IPD, agent i is paired with another agent j chosen randomly from the population, subject to the probability of being paired with a teammate ν. The two agents play one iteration of the game shown in Table 1. Each agent observes the team their counterpart belongs to instead their actual identity. In particular, for a pair of agents, i and j, their states s_i and s_j are defined as $s_i = T_j$ and $s_j = T_i$. Given each s, the two agents simultaneously choose actions a_i and a_j, either to cooperate or defect. They do not observe the action of their counterpart, but instead receive rewards $R_i^{\mathbf{cr}}$ and $R_j^{\mathbf{cr}}$ based on every interaction and their credo. Each i (and also j with their information) stores the tuple $\langle s_i, a_i, R_i^{\mathbf{cr}} \rangle$ in their replay buffer to train their policy after each episode using Deep Q-Learning [1] by sampling a random batch of 32 interactions. Each agent's internal neural network consists of an input layer of size $|\mathcal{T}| = 5$, two hidden layers of 200 nodes each with hypobolic tangent activation functions, and a two-action output layer with a linear activation function. Our agents use a learning rate of 1×10^{-4} and discount factor of 0.99 with ϵ-exploration.

A.2 Cleanup Markov Game

A.2.1 Cleanup RL Algorithm. The typical environmental setup for Cleanup implemented in past work uses five agents. However, five agents are unable to create multiple teams with the same size. Therefore, we instantiate $N = 6$ agents to create $|\mathcal{T}| = 3$ teams of two agents each. We deploy the default Proximal Policy Optimization (PPO) [2] learning algorithm from the Cleanup repository [3]. PPO is a policy gradient algorithm which constrains the space of policy updates to avoid large policy updates for smoother training and has been previously shown as a good algorithm for agents to learn in Cleanup. Agents are only able to observe the a 15×15 box centered at their location and update their policies using the environment's default batch size of at least 16,000 timesteps and a maximum of 100,000 timesteps. Each episode executes for 1,000 timesteps and we run experiments for 1.6×10^8 timesteps. The learning rate decreases linearly from 1.2×10^{-3} to 1.2×10^{-5} over the first 2×10^7 timesteps and remains static afterwards.

A.2.2 Environment Parameters. An example of Cleanup with three teams is shown in Figure 1. The social dilemma dynamics of Cleanup rely on the existence of waste and apples. The functions which govern the creation of these features can be easily modified; however, we evaluate our model of teams primarily using the default parameters of the environment which include a waste regeneration rate, waste regeneration threshold, and apple generation rate. Once less than 40% of the river (aquifer) grid-cells contain waste, waste regenerates at each clean cell with a probability of 50% at each timestep. Below this waste regeneration, apples spawn at each location in the orchard with a linear probability ranging from 0% when waste is 40% of the river up to 5% when waste makes up 0% of the river.

B EQUILIBRIUM ANALYSIS

Assume a pair of agents, i, j, have been selected to interact at some iteration of the IPD and agent i knows j will be a teammate with probability ν and a non-teammate with probability $(1 - \nu)$. When calculating the expected values of cooperation and defection with different credo, the fully self-focused and system-focused values are simply calculated using Table 1 of the main text. Team-focused credo becomes more complex since it is a mixture of the mixed-motive and common interest game depending on the probability of being paired with a teammate ν. Let $\sigma_{T_i} = (\sigma_{ji}, 1 - \sigma_{ji})$ represent j's strategy profile when $j \in T_i$, where σ_{ji} is the probability for cooperation (C). Likewise, let $\sigma_{T_j} = (\sigma_{jj}, 1 - \sigma_{jj})$ be j's strategy profile when $j \in T_j$, any other team. We show the derivation for team-focused agents and continue with the final equilibrium with credo below.

B.1 Team-Focused Agents

The expected utility of choosing to cooperate (C) or defect (D) for an agent with team-focused credo can be derived based on Table 1, ν, and the strategy of j, σ_T. First we show the derivation for a fully team-focused agent i's expected utility for choosing C subject to j's strategy:

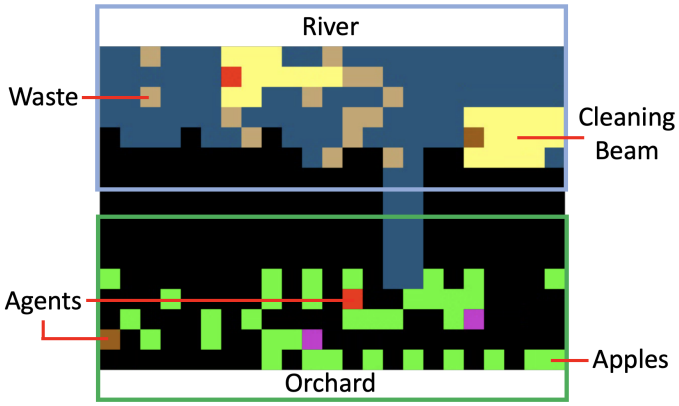

Figure 1: Screenshot of the Cleanup environment with three teams of two agents.

$$\mathbb{E}(C,\sigma)_{\mathcal{T}} = v\left[\sigma_j(b-c) + (1-\sigma_j)\frac{b-c}{2}\right] + (1-v)\left[\sigma_j(b-c) + (1-\sigma_j)(-c)\right] \tag{1}$$

$$= v\left[\frac{2\sigma_j(b-c)}{2} + \frac{b-c}{2} - \frac{\sigma_j(b-c)}{2}\right] + (1-v)\left[\sigma_j b - \sigma_j c - c + \sigma_j c\right] \tag{2}$$

$$= v\left[\frac{\sigma_j b - \sigma_j c}{2} + \frac{b-c}{2}\right] + (1-v)\left[\sigma_j b - c\right] \tag{3}$$

$$= v\left[\frac{(b-c)(\sigma_j+1)}{2}\right] + (1-v)\left[\sigma_j b - c\right] \tag{4}$$

$$= \frac{v(b-c)(\sigma_j+1)}{2} + (1-v)(\sigma_j b - c). \tag{5}$$

Now we show the derivation for a team-focused agent i's expected utility for choosing D subject to j's strategy:

$$\mathbb{E}(D,\sigma)_{\mathcal{T}} = v\left[\sigma_j\frac{(b-c)}{2}\right] + (1-v)\left[\sigma_j b\right] \tag{6}$$

$$= \frac{v\sigma_j(b-c)}{2} + (1-v)\sigma_j b \tag{7}$$

The terms for playing defection with a counterpart who mutually defects is zero, and are therefore omitted above. Next, we show how the final equilibrium is derived using our parameters which define credo.

B.2 Equilibrium with Credo

Our credo vector defines how self-focused, team-focused, or system-focused an agent is while it learns in our environment. We can calculate and derive when an agent has the incentive to cooperate in the Prisoner's Dilemma stage-game as:

$$\psi_i\mathbb{E}(C,\sigma)_{\mathcal{I}} + \phi_i\mathbb{E}(C,\sigma)_{\mathcal{T}} + \omega_i\mathbb{E}(C,\sigma)_{\mathcal{S}} \geq \psi_i\mathbb{E}(D,\sigma)_{\mathcal{I}} + \phi_i\mathbb{E}(D,\sigma)_{\mathcal{T}} + \omega_i\mathbb{E}(D,\sigma)_{\mathcal{S}}. \tag{8}$$

Expanding each term with the derivations above and in the main text, we get:

$$\psi_i\left[\sigma_j(b-c) + (1-\sigma_j)(-c)\right] + \phi_i\left[\frac{v(b-c)(\sigma_j+1)}{2} + (1-v)(\sigma_j b - c)\right] + \omega_i\left[\sigma_j(b-c) + (1-\sigma_j)\left(\frac{b-c}{2}\right)\right] \geq \psi_i\left[\sigma_j(b)\right] + \phi_i\left[\frac{v\sigma_j(b-c)}{2} + (1-v)\sigma_j b\right] + \omega_i\left[\sigma_j\left(\frac{b-c}{2}\right)\right]. \tag{9}$$

We expand and simplify:

$$\psi_i\left[\sigma_j b - \sigma_j c - c + \sigma_j c\right] + \phi_i\left[\frac{v(b-c)(\sigma_j+1)}{2} - c + vc\right] + \omega_i\left[\sigma_j(b-c) + \frac{b-c}{2} - \sigma_j\left(\frac{b-c}{2}\right)\right] \geq \psi_i\left[\sigma_j(b)\right] + \phi_i\left[\frac{v\sigma_j(b-c)}{2}\right] + \omega_i\left[\sigma_j\left(\frac{b-c}{2}\right)\right]. \tag{10}$$

We can subtract everything on the right and be left with zero.

$$\psi_i\left[-c\right]+\phi_i\left[\frac{\nu(b-c)}{2}-c+\nu c\right]+\omega_i\left[\sigma_j(b-c)+\frac{b-c}{2}-\sigma_j(b-c)\right]\geq 0 \tag{11}$$

$$\psi_i\left[-c\right]+\phi_i\left[\frac{\nu(b-c)}{2}-c+\nu c\right]+\omega_i\left[\frac{b-c}{2}\right]\geq 0 \tag{12}$$

The self- and system-focus terms are now fully simplified leaving the team-focus derivation remaining. We move $\phi_i\left[2c\right]$ to the other side of the inequality and simplify further.

$$-\psi_i c+\phi_i\left[\nu(b-c)+2\nu c\right]+\omega_i\left[\frac{b-c}{2}\right]\geq \phi_i\left[2c\right] \tag{13}$$

$$-\psi_i c+\phi_i\left[\nu-\frac{2c}{b+c}\right]+\omega_i\left[\frac{b-c}{2}\right]\geq 0 \tag{14}$$

$$\phi_i\left(\nu-\frac{2c}{b+c}\right)+\omega_i\left(\frac{b-c}{2}\right)\geq \psi_i c \tag{15}$$

This last step represents the final derivation shown as Equation 2 of the main text. This equilibrium signifies under which conditions an agent has more incentive to cooperate than to defect.

REFERENCES

[1] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (2015), 529–533.

[2] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. *CoRR* (2017).

[3] Eugene Vinitsky, Natasha Jaques, Joel Leibo, Antonio Castenada, and Edward Hughes. 2019. An Open Source Implementation of Sequential Social Dilemma Games. https://github.com/eugenevinitsky/sequential_social_dilemma_games/issues/182. GitHub repository.