Learning to Learn Group Alignment: A Self-Tuning Credo Framework with Multiagent Teams

David Radke*
University of Waterloo
Waterloo, Canada
dtradke@uwaterloo.ca

Kyle Tilbury*
University of Waterloo
Waterloo, Canada
ktilbury@uwaterloo.ca

ABSTRACT

Mixed incentives among a population with multiagent teams has been shown to have advantages over a fully cooperative system; however, discovering the best mixture of incentives or team structure is a difficult and dynamic problem. We propose a framework where individual learning agents self-regulate their configuration of incentives through various parts of their reward function. This work extends previous work by giving agents the ability to dynamically update their group alignment during learning and by allowing teammates to have different group alignment. Our model builds on ideas from hierarchical reinforcement learning and meta-learning to learn the configuration of a reward function that supports the development of a behavioral policy. We provide preliminary results in a commonly studied multiagent environment and find that agents can achieve better global outcomes by self-tuning their respective group alignment parameters.

1 INTRODUCTION

Cooperation and teamwork are central to the success of many human endeavours. Recently, there has been increasing support for the study of cooperation and teams being central to the development of artificial intelligence (AI) and multiagent systems (MAS) [2, 3]. Similarly to humans, intelligent agents cooperating and working in teams can enhance their capabilities beyond those of a single agent. However, recent work has shown that agents defined to be fully cooperative can be sub-optimal; agents that are not fully aligned with their teammates can achieve more globally favorable results [5, 13].

This paper extends a recently proposed model, *credo* [14]. Credo regulates how an individual learning agent optimizes for multiple objectives in the presence of teams. Specifically, credo represents how much the agent optimizes for the goals of different groups they belong to: their individual goals, the goals of any teams they belong to, and the goals of the entire system. In previous experiments within multiagent reinforcement learning (MARL) environments, the credo model showed that the best global outcomes for a population of agents were achieved when agents in a larger group were somewhat selfish or when agents were mostly aligned with a smaller sub-team, robust to some amount of selfishness. While credo was predetermined and fixed in these past experiments, the results motivate the key research question this paper aims to address: can giving agents the ability to dynamically tune their credo allow them to learn favorable group alignments automatically?

*These authors contributed equally to this work.

Proc. of the Adaptive and Learning Agents Workshop (ALA 2023), Cruz, Hayes, Wang, Yates (eds.), May 29-30, 2023, London, UK, https://alaworkshop2023.github.io/. 2023.

In this paper, we conceptualize and provide a preliminary approach that enables agents to self-tune their credo. We provide theoretical foundations as to the motivation behind self-tuning credo in the context of different team structures and group alignment. Further, we detail our framework that endows agents with the ability to tune their own credo. The framework borrows implementation concepts from hierarchical reinforcement learning (HRL) and meta-learning. Conceptually, each agent has a credo-tuning policy and a behavioral policy to maintain the decentralized nature of individual learning agents. The values of an agent's credo ultimately shapes their reward function, and thus, the optimization landscape of their behavioral policy. The dual-layer structure is similar to high and low-level policies in HRL, while the credo-tuning policy learning to shape the optimization landscape for the behavioral policy reflects that of a meta-learning problem.

We present preliminary results in a widely studied MARL environment, the Cleanup Gridworld Game [23], and outline future plans for evaluation. We show that, when starting from a known sub-optimal group alignment (i.e., sub-optimal credo), agents that tune their respective credos with our framework move to a better group alignment and learn a more globally favorable joint policy. While favorable team structures and group alignments have been explored in our preliminary testing environment, we describe our plans to test our framework in an environment where these are not known. The goal of this work is to enable agents to optimize their behaviors towards the various groups they belong to in any environment – enabling agents to learn better joint policies while eliminating the need for researchers and practitioners to engineer specific team structures and credo parameters. With this paper we make the following contributions:

- We provide theoretical motivation behind dynamically tuning credo (Section 4.1).
- We conceptualize an agent framework to allow agents to self-regulate their own individual credo (Section 4.2).
- We present preliminary results demonstrating the efficacy of our framework (Section 5.4) and outline future work (Section 6).

2 PRELIMINARIES

We model our base environment as a stochastic game $\mathcal{G} = \langle \mathcal{N}, S, \{A\}_{i \in N}, \{R\}_{i \in N}, P, \gamma, \Sigma \rangle$. $\mathcal{N}$ is our set of all agents that learn online from experience (with size $N \in \mathbb{N}$) and S is the state space, observable by all agents, where s_i is a single state observed by agent i. $A = A_1 \times \ldots \times A_N$ is the joint action space for all agents where A_i is the action space of agent i. $R = R_1 \times \ldots \times R_N$ is the joint reward space for all agents where R_i is the reward function of agent i defined as $R_i : S \times A \times S \mapsto \mathbb{R}$, a real-numbered reward for taking an action in an initial state and resulting in the next state. $P : S \times A \mapsto \Delta(S)$ represents the transition function which maps a state and joint action into a next state with some probability and γ represents the discount factor so that $0 \leq \gamma < 1$. Σ represents the policy space of all agents, and the policy of agent i is represented as $\pi_i : S \mapsto A_i$ which specifies an action that the agent should take in an observed state.[1]

We use "common interest" to refer to when agents share their reward and a *team* is a set of individual agents that can have some degree of common interest for team-level goals. Given a population, multiple teams with different preferences and interests that are not in zero-sum competition may co-exist. The collection of all teams is referred to as a team *structure*. We denote the set of all teams as $\mathcal{T}$, the teams agent i belongs to as $\mathcal{T}_i$, and a specific team as $T_i \in \mathcal{T}_i$.

The credo model presented in [14] relaxes the assumption that teammates are fully aligned though common interest to allow settings where agents may only partially optimize for a team's goal. For example, an agent may optimize their policy for the performance of one or multiple teams, while also being somewhat oriented towards it's own personal goals. This is done by decomposing an agent's reward function to be a combination of their individual environmental reward $IR_i = R_i$, the rewards i receives from each team they belong to $TR_i^{T_i} \forall T_i \in \mathcal{T}_i$, and the reward i receives from the system of $|N|$ agents SR_i. $TR_i^{T_i}$ and SR_i can be implemented with any function to aggregate and distribute rewards.

Each agent has a credo vector of parameters where the sum of all parameters is 1, represented $\mathbf{cr}_i = \langle \psi_i, \phi_i^{T_1}, \ldots, \phi_i^{T_{|\mathcal{T}|}}, \omega_i \rangle$, where ψ is the credo parameter for i's individual reward IR_i, $\phi_i^{T_i}$ is the credo parameter for the reward $TR_i^{T_i}$ from team $T_i \in \mathcal{T}_i$, and ω_i is the credo parameter for the reward i receives from the system SR_i. The parameter notation is organized by increasing order of group size, so that $\mathbf{cr}_i = \langle \text{self}, \ldots, \text{teams}, \ldots, \text{system} \rangle$, where $|\text{self}| < |\text{teams}| \leq |\text{system}|$. An agent's credo-based reward $R_i^{\mathbf{cr}}$ is a weighted combination of that agent's credo parameters and reward from that group. Expanded in Section 4.2, in this work we implement functions for $TR_i^{T_i}$ and SR_i specially designed for the self-tuning scenario that maintain consistency with the original implementation in [14].

3 RELATED WORK

Humans have developed with an inherent bias towards teamwork. However, humans are only able to reliably maintain social relationships with a maximum number of individuals, causing them to form smaller groups [4]. Analyzing between-team behaviors is often done in organizational psychology (OP), focusing on the concept of social identification or people's perceptions of their goals [12, 19]. Team members may need to balance tendencies for their own personal goals with the goals of their team or the entire system [1, 25]. Humans are continuously learning; thus, this balance of how humans optimize for goals is likely to be dynamic instead of static.

In AI, the concept of multiple non-conflicting teams within a larger system has been primarily explored for task completion [7, 20], and more recently been used in social dilemma scenarios [13]. Furthermore, agents optimizing their behavior while balancing between personal and group-level goals has been of growing interest to the AI community [5, 11]. One such example of this is ad hoc teamwork which relies on the ability to assess the goals of an individual or group to best optimize a cooperative utility function [10, 17].

A previously proposed model, credo, considers how agents optimize for various goals in the context of multiple non-conflicting teams within a larger system [14]. Credo defines how agents optimize for various groups they belong to, namely themselves, any teams they belong to, and the entire system. While credo showed how groups with mixed motives or some degree of selfishness can significantly outperform fully cooperative populations, all agents' credo parameters were initialized the same and kept constant throughout experiments. Other work has studied concepts of dynamic reward sharing and the emergence of coordination; however, that work relied on random perturbations of reward sharing parameters and did not consider the existence of defined team structures [6].

In this paper, we propose an agent framework where agents are able to self-tune their individual credo parameters for groups they belong to. Our model builds on hierarchical reinforcement learning (HRL) concepts to define multiple policies within a single individual learning agent. Given a population of credo-tuning agents, we hope to develop continuously evolving policies that overcome sub-optimal team definitions, recover favorable joint policies, and preserve cooperation across multiple learning entities.

4 SELF-TUNING CREDO

In this section we provide motivation for allowing agents to self-tune their credo parameters (Section 4.1) and detail our framework that allows agents to do this (Section 4.2).

4.1 Motivation

The main motivation behind allowing agents to self-tune their credo parameters is to recover a more favorable joint policy despite a sub-optimal group environment. For example, Figure 1, adapted from the original credo paper [14], shows a 33% increase in mean population reward in the Cleanup gridworld game when a population of six agents were 80% cooperative and 20% selfish compared to fully cooperative (Scenario 1). Both scenarios highlighted in Figure 1 achieve the highest mean population reward because agents tend to learn better joint policies under certain combinations of team structure and credo definitions.

Defining a team structure that best supports how individual agents learn is a difficult problem. Recent work has used a fully cooperative population (i.e., shared reward function) to compare results with; however, the credo model has shown how a fully aligned population may be sub-optimal. Providing agents with the ability to self-tune their credo parameters allows agents to regulate their internal reward function through group alignment. For example, credo-tuning agents defined in one large cooperative population may discover the benefits of being slightly selfish on their own and converge to Scenario 1 in Figure 1. The pressures to tune credo and recover different reward signals are highly correlated with the size of the reward-sharing group. In this section, we detail how features of group size impact reward signals (Section 4.1.1) and how

[1]We can also allow for randomized policies.

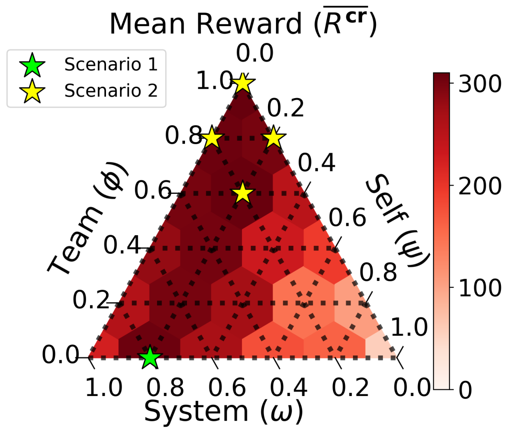

Figure 1: Mean population reward for every credo parameter in the Cleanup environment from [14]. These experiments have $|\mathcal{T}| = 3$ teams of two agents each. Two scenarios achieve the highest reward: when credo has slight self-focus paired with high system-focus (green star) and when team-focus is high (yellow stars).

tuning credo can be leveraged to recover stronger reward signals (Section 4.1.2).

4.1.1 Reward Signals and Group Size. Consider a scenario with a fully aligned cooperative population of N agents with only behavioral policies (i.e., only one group, the entire system). In this setting, all agents share rewards at every timestep. Thus, if agent i collects a reward of r at time t, all agents receive a reward of $\frac{r}{N}$ (assuming no other agent collects a non-zero reward from the environment). The size of the reward-sharing group has two impacts on the reward function and agents' abilities to perform effective credit assignment.

Probability of non-zero reward approaches one: Starting with three common assumptions in reinforcement learning (RL), assume agents 1) are initialized with random policies, 2) fully explore the state space in the limit, and 3) each have equal probability of collecting a non-zero reward from the environment $P(r)_i$ (i.e., $P(r)_i = P(r)_j$ for any agents i and j). The probability of **any** agent collecting a non-zero reward is: $1 - P(r)^N$. The derivative of this is positive, $f'(1 - P(r)^N) = N \cdot P(r)^{N-1}$; thus, agents in a reward-sharing group are monotonically more likely to receive a non-zero feedback signal at any timestep as the size of that group increases. This probability approaches 1 as $N \longrightarrow \infty$ in the limit.

Variance of non-zero reward approaches zero: Agents receiving non-zero reward for their actions causes them to assign credit to these actions. More positive reward for certain state-action pairs will result in them executing these state-action pairs more often in the future, and vice versa for negative reward. However, while the **probability** of receiving a non-zero reward approaches 1 as N increases, the derivative $f'\left(\frac{r}{N}\right) = -\frac{r}{N^2}$ implies the value of this non-zero reward monotonically approaches 0 as N increases. With a large group, the reward that each agent receives at every timestep will be a function of the expected number of agents that obtain rewards at any timestep. As $N \rightarrow \infty$, the variance of this reward approaches zero. Thus, agents would be unable to perform effective credit assignment if the size of their reward-sharing group is too large.

4.1.2 Recovering a Stronger Reward Signal. The previous subsection describes a scenario where agents lose the ability to perform effective credit assignment if the size of a reward sharing group is too large (assuming agents fully share rewards). The credo model removes the assumption that agents fully share rewards to analyze situations where agents can learn from multiple types of groups they belong to. Thus, regulating credo could allow agents to recover meaningful feedback signals from their actions in environments where credit assignment becomes challenging (i.e., if the reward-sharing group is too large).

Consider again the results from the original credo paper shown in Figure 1. Agents defined in a fully aligned population (one team of six agents) fail to converge to the most efficient joint policy; however, agents are able to recover a better joint policy when agents are 20% selfish (Scenario 1). Agents that can self-regulate their credo parameters may recover better joint policies despite a sub-optimal environment that can impose credit assignment challenges, such as poorly defined team structures or group alignment.

4.2 Self-Tuning Credo Framework

This section details how we extend the credo-based reward function design and our proposed self-tuning credo framework.

4.2.1 Extending Credo. Recall from Section 2 that agent i's credo is defined as a vector of parameters that sum to 1, represented $\mathbf{cr}_i = \langle \psi_i, \phi_i^{T_1}, \ldots, \phi_i^{T_{|\mathcal{T}|}}, \omega_i \rangle$, where ψ is the credo parameter for i's individual reward IR_i, $\phi_i^{T_i}$ is the credo parameter for the reward $TR_i^{T_i}$ from team $T_i \in \mathcal{T}_i$, and ω_i is the credo parameter for the reward i receives from the system SR_i. In this paper, we define agent i's credo-based reward function $R_i^{\mathbf{cr}}$ to be calculated as:

$$R_i^{\mathbf{cr}} = \psi_i IR_i + \sum_{T_i \in \mathcal{T}_i} \frac{\phi_i^{T_i}}{\sum_{j \in T_i} \phi_j^{T_i}} TR_i^{T_i} + \frac{\omega_i}{\sum_{j \in N} \omega_j} SR_i. \quad (1)$$

Different from the original implementation, Equation 1 allocates team and system rewards based on the *ratio* of an agent's credo parameter for that group compared to the sum of credo parameters of other agents in that group. This is necessary modification for the scenario when agents may have different credo parameters for the same group. To maintain consistency with past work, we modify $TR_i^{T_i}$ and SR_i to be the weighted sum of agents' rewards and their credo parameter for that specific group:

$$TR_i^{T_i} = \sum_{j \in T_i} \phi_j^{T_i} R_j(S, A_j, S),$$

$$SR_i = \sum_{j \in N} \omega_j R_j(S, A_j, S).$$

This ensures all rewards that are collected from the environment are re-allocated to the various groups and scaled according to all credo parameters. These modifications are equivalent to the previous credo setting when all agents have the same credo, but expand the reward function dynamics to when teammates may not have the same credo for a team.

4.2.2 Agent Architecture. An overview of our proposed agent framework is given in Figure 2. The architecture of the agent is a multi-level policy inspired by HRL, where each layer influences the learning problem of the other. The "low-level" policy, π_i, is a typical behavioral policy that takes actions a_i conditioned on an observed state s_i within an environment. At each timestep, rewards are shared with other agents according to the agent's credo parameters $\mathbf{cr}_i$. The "high-level" policy, $\pi_i^{\mathbf{cr}}$, modifies the agent's credo parameters at a slower time scale. Conditioned on the previous credo parameters, $\mathbf{cr}_i$, and the corresponding low-level policy's reward over $E \geq 1$ episodes, $\overline{R_i^E}$, the high-level agent produces updated credo parameters, $\mathbf{cr}_i'$. The top-layer policy operates at a slower time scale than the low-level behavioral policy to allow the low-level policy to gain experience with a particular credo and stabilize learning.

Both policies learn from experience using RL. They both aim to individually maximize their sum of discounted future rewards and neither policy directly observes the other (i.e., they are both individual learning policies). However, each policy directly influences the optimization landscape of the other. The behavior of the low-level policy determines the reward feedback for the high-level policy; if the behavioral agent fails to gain reward, the high-level credo-tuning policy fails to get positive feedback. Concurrently, the credo output of the high-level policy shapes the reward function of the low-level policy for the next set of E episodes.

As mentioned in Section 4.1.1, tuning the amount of shared reward within groups regulates 1) the probability of an agent receiving a non-zero reward from a group with more teammates, and 2) the variance of their reward signal. Thus, the high-level credo policy shapes the influence of these two aspects with respect to all groups referenced in the credo vector to guide the learning process of the low-level behavioral policy (self, any teams, and system).

5 EVALUATION

This section outlines our implementation details, experimental environment and setup, and presents preliminary experimental results.

5.1 Implementation

Low-level Behavioral Policy: We implement the behavioral policies of our agents with Proximal Policy Optimization (PPO) [16]. The PPO implementation in [14] used an older version of the RLlib library (version 0.8.5) which made interconnecting the credo-tuning framework infeasible. Thus, we adapted the same architecture as the agents in [14] to the current version of RLlib (version 2.1.0) to incorporate the credo-tuning agent architecture shown in Figure 2.[2]

High-level Credo Policy: As a preliminary construction, and to reduce sample complexity, we implement the high-level credo policy as a Q-Learning agent with ϵ-greedy exploration ($\epsilon = 20\%$) [24]. Consistent with the original credo paper, we define agents to belong to only one team, making credo vectors with three parameters (i.e., $\mathbf{cr}_i = \langle \psi_i, \phi_i, \omega_i \rangle$). We limit possible agent credos to intervals of 0.2, creating a state space of 21 possible states (shown in Figure 1 from [14]). With three credo parameters, the agent can choose from any of seven actions. The action space consists of either increasing/decreasing any combination of credo parameters (six actions) or doing nothing (one action). For example, if $\mathbf{cr}_i = \langle 0.2, 0.0, 0.8 \rangle$, the agent can take an action to decrease self-focus and increase system focus (by increments of 0.2) to result in $\mathbf{cr}_i' = \langle 0.0, 0.0, 1.0 \rangle$. If the agent chooses an action that would increase any credo parameter above 1.0 or below 0.0, no action is taken and $\mathbf{cr}_i = \mathbf{cr}_i'$. The behavioral policies are updated with $\mathbf{cr}_i'$ for the next E episodes.

[2]https://docs.ray.io/en/latest/rllib/index.html

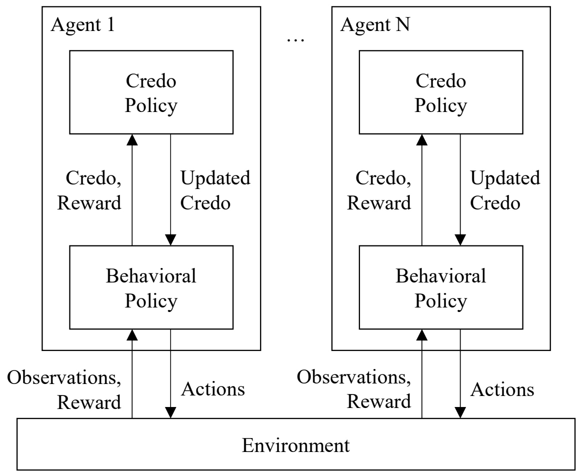

Figure 2: Overview of the proposed credo-tuning agent framework. Each agent has two policies that operate at different time scales: a low-level behavioral policy that acts within an environment and a high-level credo-tuning policy that operates every $E \geq 1$ episodes. The credo-tuning policy shapes the optimization landscape for the behavioral policy while the learned behavior impacts the reward function for the credo-tuning policy.

5.2 Environment

We perform our preliminary evaluation in the Cleanup Gridworld Game [23]. Cleanup is a temporally and spatially extended Markov game representing a sequential social dilemma. We keep the underlying environment unchanged from previous setups [8] except for the team and system reward functions. Agent observability is limited to an egocentric 15×15 pixel window and consuming an apple yields +1 reward. Apple regrowth rate is dependent on the cleanliness of an adjacent river. To be successful in cleanup, agents must learn to balance actions of consuming apples and cleaning the river (which returns no positive reward). Agent rewards are determined by their credo $\mathbf{cr}_i$ which is updated at regular intervals. Consistent with the original credo paper, we set the size of each team to be two agents $|T_i| = 2$, creating $|\mathcal{T}| = 3$ disjoint teams from the population of $N = 6$ agents [14]. Agents are implemented with PPO behavioral policies, Q-learning credo policies, and experiments last for 3.2×10^8 environment steps and credo parameters are updated every 96,000 environment steps.

5.3 Experiment

We design an experiment to evaluate if credo-tuning agents can overcome a sub-optimal initialization to recover a joint policy that achieves higher mean population rewards (Scenario 1 or 2 in Figure 1). We initialize agents to be fully system-focused (i.e., $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$). The low-level behavioral policy trains, and the high-level credo policy updates $\mathbf{cr}_i$, every 96 episodes (rollouts of six workers with 16 environment copies each). This is equivalent to initializing agents with credo parameters in the bottom left corner of Figure 1; however, agents' credo policies are now able to adjust the agent's credo parameters.

This setting directly evaluates our discussion in Section 4.1.1. One of the key findings in previous work is that some amount of mixed incentives can achieve more favorable global outcomes than a fully cooperative population [5, 14]. In Cleanup, agents that are slightly self-focused or fully team-focused (Scenario 1 and 2 respectively in Figure 1) learn a better global joint policy through division of labor. In these settings, agents divide labor and learn to specialize into roles of four apple picking agents and two river cleaning agents. When agents are fully system-focused, they specialize into the sub-optimal joint policy of three apple pickers and three river cleaners.

We hypothesize that a full system-focused group learns this sub-optimal joint policy due to more a difficult credit assignment problem. Agents in Scenario 1 of Figure 1 learn the best joint policy by recovering slightly stronger reward signals by being 20% self-focused. The design of this experiment evaluates the ability for credo-tuning agents to recover stronger reward signals and converge to a better joint policy. Intuitively, this can be thought of as agents learning to configure their credo parameters such that they converge to high-reward areas of Figure 1.

5.4 Preliminary Results

This section shows preliminary results of the experiment detailed in Section 5.3. In the credo tuning experiment, all agents are initialized in the Cleanup environment with $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$. Each agent's behavioral policy updates every 96,000 environmental timesteps (96 episodes), at which point the high-level credo policy modifies the agent's credo parameters. The behavioral policy never observes the credo parameters but instead experiences changes to their reward function over the next batch of episodes. We compare credo tuning to two configurations where credo remains static. In the static team-focus experiment, agents maintain $\mathbf{cr}_i = \langle 0.0, 1.0, 0.0 \rangle$ for the entire experiment and fully share rewards with their teammates. In the static system-focus experiment, agents maintain $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$ for the entire experiment and share rewards with all agents (i.e., a fully cooperative system). In all experiments, there are six agents that are divided into three disjoint teams of two agents each (i.e., $N = 6$, $|\mathcal{T}| = 3$, and $|T_i| = 2$). We execute four trials of each experiment configuration.

We observe the same patterns with the static experiments as in previous work [13, 14]: full team-focus performs significantly better than full system-focus. However, we found that updating the PPO agents from RLlib 0.8.5 to RLlib 2.1.0 modified their learning curves so that agents learn more gradually (despite no changes to the algorithm configurations). Thus, while our direct learning

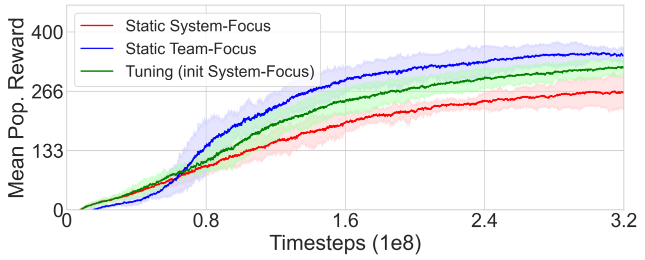

Figure 3: Reward curves in the Cleanup environment for each experiment in our evaluation. Results are the mean across 4 trials for each experiment reported with 95% confidence intervals. The static team-focus environment has been shown to achieve the highest mean population reward in Cleanup with different credos (Figure 1 Scenario 2). This shows that credo-tuning agents that are initialized with system-focus credo can increase their mean population reward to improve towards the level of team-focused agents.

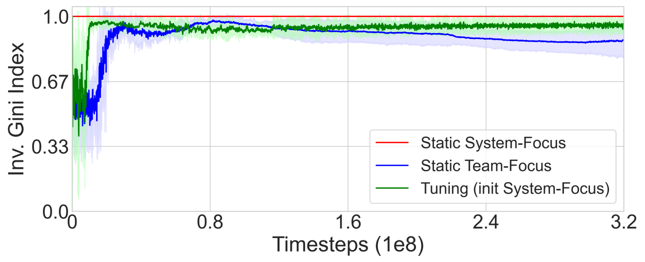

Figure 4: Inverse Gini index curve in the Cleanup environment for each experiment in our evaluation. Results are the mean across 4 trials for each experiment reported with 95% confidence intervals. Static system-focus credo is defined to have full equality and is always 1. This shows that credo-tuning agents converge to slightly higher equality than the static team-focused experiment.

curves are not comparable to past work, the overall result remains consistent and we extend the duration of the experiments from 1.6×10^8 to 3.2×10^8 environment steps.

5.4.1 Reward. Figure 3 shows the mean population reward and 95% confidence intervals obtained by the population of agents in the three different credo scenarios: static system-focus, static team-focus, and credo-tuning agents that were initialized to be system-focused. The y-axis shows mean population reward and the x-axis shows timesteps of the experiment. Consistent with past work, we find that static agents that are fully team-focused (blue) perform significantly better than static system-focused agents (red). This is due to team-focused agents converging to a more efficient division of labor joint policy with two river cleaning agents and four apple picking agents, whereas system-focused agents converge to three agents each cleaning the river or picking apples.

Recall from Figure 1 that full team-focus credo is one setting that achieves the highest reward in this configuration; thus, we treat full team-focus as an upper-bound result in this domain. The goal

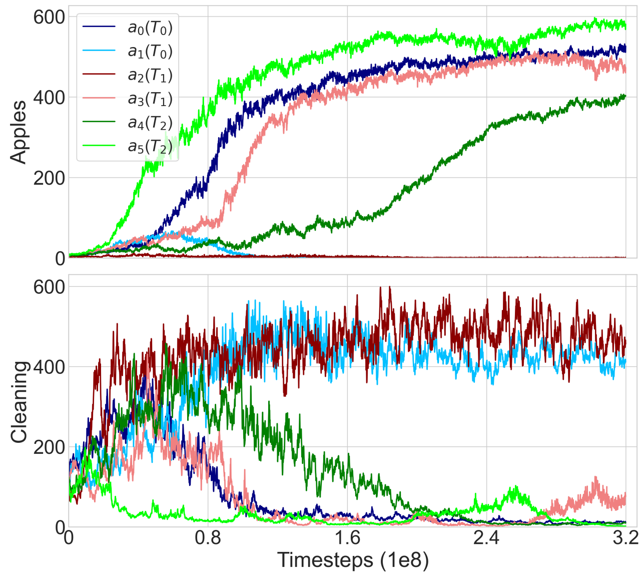

Figure 5: Amount of apples consumed (top) and cleaning beam actions (bottom) by each agent for one trial of the credo-tuning experiment with agents initialized with system-focused credo (green line in Figures 3 and 4). Agents are labeled so that $a_0(T_0)$ is agent 0 on team 0. Teammates are colored with different shades of the same color. Whereas system-focused agents converge to a joint policy of three apple pickers and three cleaning agents, credo-tuning agents recover the better joint policy of four apple pickers and two cleaning agents autonomously (same as fully team-focused agents) and generate more reward (Figure 3).

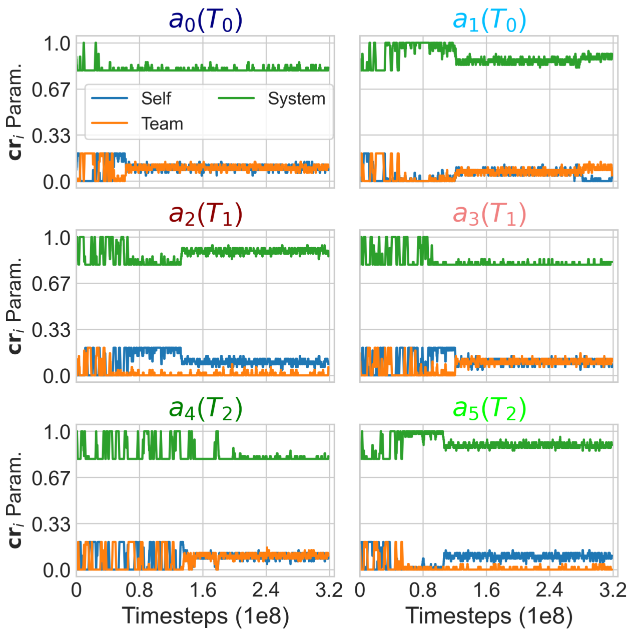

Figure 6: Credos of all six agents over time in the same credo-tuning trial as Figure 5. Each plot shows the credo parameters for a different agent shown in Figure 5. Each y-axis represents credo parameter space and each x-axis represents timesteps. We observe heterogeneous credo parameters emerge across the population; however, a_4 becomes more self- and team-focused as it switches roles to become an apple picking agent.

of credo-tuning agents is not to overtake the team-focus credo, but converge to credo parameters that achieve higher reward than their initialized settings (i.e., fully system-focused credo; red line). The green line in Figure 3 shows the mean population reward and 95% confidence intervals for the credo-tuning agents initialized with full system-focus credos. Through the first 800,000 timesteps of the experiment, the credo-tuning agents (green) learn along the same trajectory as the system-focused agents (red). However, giving agents the ability to modify their credo parameters leads to the population achieving roughly 21% more mean population reward than the system-focus credo by the end of the experiment (320 for credo-tuning agents compared to 264 for static system-focus agents). This shows the ability for credo-tuning agents to achieve more mean population reward despite a known sub-optimal team and credo parameter initialization.

5.4.2 *Reward Equality.*

Since certain roles in the environment do not produce reward and teammates are able to define different credos, it is important to consider population equality to examine if tuning credo leads to significant inequality among the population. We model population reward equality as the inverse Gini index, similar to past work [11, 13]:

$$Equality = 1 - \frac{\sum_{i=0}^{N}\sum_{j=0}^{N}|R_i^{\mathbf{cr}} - R_j^{\mathbf{cr}}|}{2N^2\overline{R^{\mathbf{cr}}}}, \qquad (2)$$

where values closer to 1 represent more equality. Figure 4 shows our equality results, where the y-axis shows the mean inverse Gini index with 95% confidence intervals and the x-axis is the number of timesteps. Since the static system-focus scenario defines agents to fully share rewards, the inverse Gini index is always equal to 1. After some initial learning, we find that the credo-tuning agents converge to a setting where the population has higher mean equality than the static team-focused setting. While this is likely impacted by the credo intialization and is worthy of further exploration, we find that credo-tuning agents discover a setting that achieves high reward while maintaining high equality across the population.

5.4.3 *Division of Labor.*

We now analyze the credo-tuning experiment specifically. Figure 5 shows the amount of apples consumed (top) and cleaning beam actions (bottom) by each credo-tuning agent in one trial where the agents are initialized to be fully system-focused (green line in Figures 3 and 4). Despite being initialized as system-focused, these agents have team membership to one of three teams (T_0, T_1, or T_2) to modify their credo towards. Agents are labeled so that $a_0(T_0)$ represents agent 0 on team 0 and teammates in the plots are colored with different shades of the same color.

Similar to the known result of static system-focused agents shown in past work [14], the agents in the credo-tuning experiment initially specialize into roles of three apple picker and three river cleaning agents. However, the advantage of agents being able to tune their credo causes the $a_4(T_2)$ agent to learn to pick apples in the second half of the experiment. This recovers the global joint policy of four apple picker agents and two river cleaning agents (joint policy of the static team-focused agents) despite agents being initialized with full system-focused credo. This causes an increase in mean population reward from the static system-focused scenario towards the full team-focused scenario. While the mean population reward level of team-focused agents is not quite reached, these agents recover the same global joint policy; thus, while we are unable to make certain claims, perhaps longer training time would see convergence to the reward level of the team-focused population (blue in Figure 3) given this joint policy.

5.4.4 Tuned Credo Parameters. Figure 6 shows how the credo parameters for each agent in the trial shown in Figure 5, modified by each agents' high-level credo policy. Each plot is titled and colored according to the agent's label and color in Figure 5. The y-axis of each plot shows the credo parameter values and the x-axis of each plot shows timesteps of the experiment. The results for each credo parameter are a sliding window mean of every 10 samples; thus, some results appear between two discrete credo steps (such as 0.1 being between 0.0 and 0.2).

Figure 6 shows that two teammates that converge to complimentary roles of one river cleaner and one apple picker, a_0 and a_1 (blue; T_0), maintain periods of non-zero team focus. This allows the agents to share some of the reward gained by their teammate while sharing the majority of their apples through the system-focus reward channel. The other team that divides labor between two roles over the entire experiment, a_2 and a_3 (red; T_1), have heterogeneous credo parameters amongst their team. While the cleaning agent a_2 maintains higher system-focus, the apple picking agent a_3 has slightly higher self-focus to keep some amount of the reward they collect to themselves. The agent that changes roles to become an apple picker, a_4 on T_2, maintains a period of being self- and team-focused, before 1.5×10^8 timesteps. At this time, their teammate ($a_5(T_2)$) develops a credo where they do not share rewards through the team parameter, instead maintaining high system-focus before becoming slightly self-focused. After a period where their teammate is not contributing to the team reward when a_4 is slightly team-focused, the agent switches behaviors to become an apple picking agent. This may indicate why a_4 becomes an apple picker with some amount of self-focus (i.e., increasing their personal reward).

These results show how our framework allows for diverse group alignments to be learned. In turn, these learned heterogeneous alignments lead to agents recovering a globally better joint policy while maintaining high equality.

6 FUTURE WORK

This paper presented an initial framework and evaluation of credo-tuning agents as a preliminary proof of concept. We will expand this work in the two main areas of experimental evaluation and model design.

6.1 Experimental Future Work

We plan on performing a more extensive empirical evaluation of our framework. We plan on designing more specific experimental scenarios such as initializing agents to be fully team-focused, self-focused, or randomly distributed across different credo parameters.These experiments will provide insight into the convergence properties of credo-tuning agents (i.e., if there is one or multiple convergence points in credo parameters for different team structures). Furthermore, allowing agents to belong to multiple teams, and self-regulate their credo for each specific team, is an area of future research that could provide insights into dynamic team membership or multi-group alignment [22]. We also plan on expanding our evaluation to the Neural MMO (NMMO) [18] environment. NMMO is a large, customizable, and partially observable multiagent environment that supports foraging and exploration. Implementing credo-tuning agents in NMMO will help understand the connections between how agents modify their credo in hunter-gatherer-type societies and in scenarios where the most beneficial joint policy may be unknown.

6.2 Model Design

We plan to improve and expand our model design. We implemented the credo policy using Q-Learning with discrete credo step sizes of 0.2 to reduce sample and state/action space complexity for the high-level policy. The state space consisted of a finite set of discrete states and had a finite set of discrete actions. However, recent work in single-agent RL has shown significant advances in continuous control problems where both the state and action spaces consist of high-dimensional continuous values [26]. Other work has shown the ability to decompose the Bellman equation into a vectorized representation to learn the value of different reward parts [9]. We plan on expanding the development of the high-level credo policy to incorporate continuous control designs to learn the value of various reward components (self, team, or system). Furthermore, we plan to expand the action space to directly output credo parameters instead of discrete modifications. Progress in this direction will expand reward function decomposition to the multiagent scenario while allowing for significantly more credo parameter combinations among the population.

7 CONCLUSION

This work presented the design of credo-tuning agent framework influenced by hierarchical reinforcement learning (HRL) and meta-learning. The dual-tiered architecture draws inspiration from the multi-layer optimization problems of HRL; however, the influences on learning dynamics at each layer are more similar to a meta-learning problem. In meta-learning, *learning to learn* is the idea in which an agent learns at two levels, each associated with different time scales [15, 21]. The ability for the behavioral policy to learn effective policies and roles among a group is guided by their reward function's feedback signals, shaped by their credo. The high-level credo policy updates these credo parameters at a slower timescale, changing the low-level behavioral policies optimization landscape and guiding the learning process through different reward components. Dynamically tuning credo allows this meta-learning problem to evolve online while maintaining the decentralized aspects of individual learning agents.

The goal of this work is to allow decentralized agents to self-regulate their credo to overcome sub-optimal initializations of credo or team structures and recover favorable policies. While previous work has shown how team structure has a significant impact on the policies that agents learn, discovering the structure that guides agents towards globally favorable results may be a hard domain dependent problem. Our preliminary results have shown how our multi-tiered learning architecture can allow agents to achieve more globally favorable results despite being initialized in a known sub-optimal configuration. The broader implications of this work allow agents to autonomously recover the learning benefits of teams and group alignment in any environment. This mitigates the burden of researchers or practitioners having to engineer team structures or credo in settings where favorable configurations may be unknown.

REFERENCES

[1] D. R. Carter, R. Asencio, Hayley M. Trainer, Leslie A. DeChurch, R. Kanfer, and S. Zaccaro. 2019. Best Practices for Researchers Working in Multiteam Systems.

[2] Allan Dafoe, Yoram Bachrach, Gillian Hadfield, Eric Horvitz, Kate Larson, and Thore Graepel. 2021. Cooperative AI: machines must learn to find common ground. *Nature* 593 (2021), 33–36.

[3] A. Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, K. Larson, and T. Graepel. 2020. Open Problems in Cooperative AI. *ArXiv* abs/2012.08630 (2020).

[4] Robin IM Dunbar. 1993. Coevolution of neocortical size, group size and language in humans. *Behavioral and brain sciences* 16, 4 (1993), 681–694.

[5] Ishan Durugkar, E. Liebman, and P. Stone. 2020. Balancing Individual Preferences and Shared Objectives in Multiagent Reinforcement Learning. In *IJCAI*.

[6] Ian Gemp, Kevin R McKee, Richard Everett, Edgar A Duéñez-Guzmán, Yoram Bachrach, David Balduzzi, and Andrea Tacchetti. 2022. D3C: Reducing the Price of Anarchy in Multi-Agent Learning. *Proceedings of the 21st International Conference on Autonomous Agents and MultiAgent Systems* (2022).

[7] B. Grosz and S. Kraus. 1996. Collaborative Plans for Complex Group Action. *Artif. Intell.* 86 (1996), 269–357.

[8] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel. 2017. Multi-agent Reinforcement Learning in Sequential Social Dilemmas. In *AAMAS*.

[9] James MacGlashan, Evan Archer, Alisa Devlic, Takuma Seno, Craig Sherstan, Peter R Wurman, and Peter Stone. 2022. Value Function Decomposition for Iterative Design of Reinforcement Learning Agents. *NeurIPS* (2022).

[10] W. Macke, R. Mirsky, and P. Stone. 2021. Expected Value of Communication for Planning in Ad Hoc Teamwork. In *AAAI-21*.

[11] Kevin R. McKee, I. Gemp, Brian McWilliams, Edgar A. Duéñez-Guzmán, Edward Hughes, and Joel Z. Leibo. 2020. Social Diversity and Social Preferences in Mixed-Motive Reinforcement Learning. *AAMAS* (2020).

[12] J. Porck, Fadel K Matta, J. Hollenbeck, Jo K. Oh, Klodiana Lanaj, and S. Lee. 2019. Social Identification in Multiteam Systems: The Role of Depletion and Task Complexity. *Academy of Management Journal* 62 (2019), 1137–1162.

[13] David Radke, Kate Larson, and Tim Brecht. 2022. Exploring the Benefits of Teams in Multiagent Learning. In *IJCAI*.

[14] David Radke, Kate Larson, and Tim Brecht. 2023. The Importance of Credo in Multiagent Learning. *Proceedings of the 22nd International Conference on Autonomous Agents and MultiAgent Systems* (2023).

[15] Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. Meta-learning with memory-augmented neural networks. In *International conference on machine learning*. PMLR, 1842–1850.

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

[17] P. Stone, G. Kaminka, S. Kraus, and J. Rosenschein. 2010. Ad Hoc Autonomous Agent Teams: Collaboration without Pre-Coordination. In *AAAI*.

[18] Joseph Suarez, Yilun Du, Phillip Isola, and Igor Mordatch. 2019. Neural MMO: A massively multiagent game environment for training and evaluating intelligent agents. *arXiv preprint arXiv:1903.00784* (2019).

[19] E. Sundstrom, K. P. D. Meuse, and D. Futrell. 1990. Work teams: Applications and effectiveness. *American Psychologist* 45 (1990), 120–133.

[20] Milind Tambe. 1997. Towards Flexible Teamwork. *Journal of Artificial Intelligence Research* 7 (1997), 83–124.

[21] Sebastian Thrun. 1998. Lifelong learning algorithms. In *Learning to learn*. Springer, 181–209.

[22] Kyle Tilbury and Jesse Hoey. 2022. Identity and Dynamic Teams in Social Dilemmas. *arXiv preprint arXiv:2208.03293* (2022).

[23] 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.

[24] Christopher JCH Watkins and Peter Dayan. 1992. Q-learning. *Machine learning* 8 (1992), 279–292.

[25] Julia Wijnmaalen, H. Voordijk, S. Rietjens, and G. Dewulf. 2019. Intergroup behavior in military multiteam systems. *Human Relations* 72 (2019), 1081 – 1104.

[26] Peter R Wurman, Samuel Barrett, Kenta Kawamoto, James MacGlashan, Kaushik Subramanian, Thomas J Walsh, Roberto Capobianco, Alisa Devlic, Franziska Eckert, Florian Fuchs, et al. 2022. Outracing champion Gran Turismo drivers with deep reinforcement learning. *Nature* 602, 7896 (2022), 223–228.