The Impact of Teams in Multiagent Systems

by

David Thomas Radke

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Computer Science

Waterloo, Ontario, Canada, 2023

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Fernando Santos Assistant Professor, Informatics Institute, University of Amsterdam
Supervisor(s):	Kate Larson Professor, David R. Cheriton School of Computer Science, University of Waterloo
	Tim Brecht Professor, David R. Cheriton School of Computer Science, University of Waterloo
Internal Member:	Jesse Hoey Professor, David R. Cheriton School of Computer Science, University of Waterloo
Internal Member:	Robin Cohen Professor, David R. Cheriton School of Computer Science, University of Waterloo
Internal-External Member:	Mark Crowley Associate Professor, Dept. of Electrical and Computer Engineering, University of Waterloo

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Across many domains, the ability to work in teams can magnify a group's abilities beyond the capabilities of any individual. While the science of teamwork is typically studied in organizational psychology (OP) and areas of biology, understanding how multiple agents can work together is an important topic in artificial intelligence (AI) and multiagent systems (MAS). Teams in AI have taken many forms, including ad hoc teamwork [226], hierarchical structures of rule-based agents [233], and teams of multiagent reinforcement learning (MARL) agents [13]. Despite significant evidence in the natural world about the impact of family structure on child development and health [122, 245], the impact of team structure on the policies that individual learning agents develop is not often explicitly studied. In this thesis, we hypothesize that teams can provide significant advantages in guiding the development of policies for individual agents that learn from experience.

We focus on mixed-motive domains, where long-term global welfare is maximized through global cooperation. We present a model of multiagent teams with individual learning agents inspired by OP and early work using teams in AI, and introduce *credo*, a model that defines how agents optimize their behavior for the goals of various groups they belong to: themselves (a group of one), any teams they belong to, and the entire system. We find that teams help agents develop cooperative policies with agents in other teams despite game-theoretic incentives to defect in various settings that are robust to some amount of selfishness. While previous work assumed that a fully cooperative population (all agents share rewards) obtains the best possible performance in mixed-motive domains [266, 71], we show that there exist multiple configurations of team structures and credo parameters that achieve up to 33% more reward than the fully cooperative system. Agents in these scenarios learn more effective joint policies while maintaining high reward equality. Inspired by these results, we derive theoretical underpinnings that characterize settings where teammates may be beneficial, or not beneficial, for learning. We also propose a preliminary credo-regulating agent architecture to autonomously discover favorable learning conditions in challenging settings.

Acknowledgements

This dissertation is not only a product of my time at the University of Waterloo, but also the culmination of many chapters of my life. For that, my gratitude spans beyond the confounds of graduate school to include everyone that has led me to the path I find myself traveling down.

I would like to thank my Mom (Susan), Dad (John), Meredith, and Dan for their love and encouragement throughout all aspects of my life. You all have been the best support system, COVID quarantine group, research group [190], and weekly Zoom group from almost every corner of the world. A special thank you to Jack for always welcoming me home with the most excitement possible.

Thank you to my wonderful partner, co-author [167, 187], and editor, Alexi. Exploring the interesting world of graduate school with you has been the adventure of a lifetime.

Thank you to my advisors, Kate Larson and Tim Brecht, for bringing me to Waterloo and helping me navigate graduate school with the utmost support. You taught me how to tackle large problems, ask hard questions, and pay attention to the smallest details – skills that will stick with me throughout my life.

To my PhD committee, Jesse Hoey, Robin Cohen, Mark Crowley, and Fernando Santos: thank you for your feedback and questions on many aspects of my work. Your own research contributions, and our constructive discussions on mine, has helped drive the work in this dissertation and beyond. Many of your own works have inspired the evolution of my own interests. Thank you to Jim Gemmell and Bill Reeves, both University of Waterloo Alumni, for helping me decide to pursue a PhD at Waterloo. A very special thank you to the late Andrew Price-Smith (APS), a beloved Canadian, hockey fan, and former Colorado College professor. You helped me make graduate school at Waterloo a reality in a pivotal time of my life. Without your support, empathy, and vision, I can confidently say that my graduate school experience would have been completely different.

Thank you to my collaborators, colleagues, and friends: Kyle Tilbury, Alex Pawelczyk, Ben Armstrong, Valerie Platsko, Sriram Subramanian, Kanav Mehra, Nanda Kishore, Marvin Pafla, and Wei Hu. Our somewhat-regularly scheduled drink nights were always a highlight of life in Waterloo; I hope they continue among future members of the group. Thank you to Mike Shaekermann and Ellie Sanoubari (partners in crime) for all of the fun times and late nights through the years.

The sport of ice hockey helped shape my collegiate and graduate school path. Furthermore, many of the ideas in this dissertation stem from my personal experiences being a

defensive partner and teammate for my entire life. I would like to acknowledge and thank all of the teammates I have had throughout my playing career. Many of those teammates have helped me to my highest moments, but more importantly, seen and helped me through my lowest moments. I am happy to say that many of you are close friends for life and I look forward to making more memories with you all in the future.

Thank you to all of the hockey coaches that helped me develop along the trajectory I ended up on. To everyone associated with the University of Waterloo men's hockey program, especially Brian Bourque, for giving me the opportunity to play for the Warriors and supporting me throughout my time in graduate school. Thank you to Gene Reilly for recruiting me to Colorado College and to my junior hockey coaches and management staff who helped me develop into a better person, teaching me valuable lessons I still carry today. A special thank you to the medical and physical therapy staff who helped me physically and mentally rebound from major surgeries at Colorado College and allowed me to see the bright opportunities ahead.

I would like to thank my various professional connections and collaborations for providing me new experiences and insights into how research is conducted within various fields. Thank you to SonyAI for giving me the opportunity to work with an amazing team of brilliant people during my internship in the Fall of 2022. Many thanks to my collaborators at the National Hockey League (NHL) for the various meetings and supporting our research projects, helping me bridge the gap between my interests of computer science and ice hockey. Thank you to the Chicago Blackhawks organization for giving me the opportunity to explore how artificial intelligence can have impact in the world of professional ice hockey.

This research was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC), an Ontario Graduate Scholarship, a Cheriton Scholarship, and the University of Waterloo President's Graduate Scholarship. Thank you to the Vector Institute for providing the compute resources necessary for much of this research.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Observed in both animal and human behavior, the ability to work in teams can magnify a group's abilities beyond those of any individual. The evolution of how cooperative behavior emerged among humans was one of 2005 *Science Magazine*'s top 25 questions facing science over the following 25 years [39]. Cooperation, teamwork, and social relationships are central to the development of human cognition and contribute to the success of many endeavours [146]. However, almost 20 years later, the emergence of cooperation among humans is still not well understood. As artificial intelligence (AI) agents further interact with people, agents will need to understand how and when cooperation is necessary or desirable. As a result, there is growing interest in making the study of cooperation central to the development of AI and multiagent systems (MAS) [38, 39]. In a human context, there is significant evidence demonstrating the importance of family stability and structure on child development and health [122, 245]; however, the impact of social structure on individual learning agents in AI is often overlooked. In this dissertation, we explore the role that *teams* play in the development of individual learning agents' policies to better understand the impact that team structures have on learning in AI.

The science of teamwork and social connection is widely studied across various disciplines in the natural sciences [7, 270]. Researchers in organizational psychology (OP) and biology have studied how teams increase the collective efforts of all sub-groups and individuals within a team [271, 143, 136, 193]. Interdependence theory has formalized the importance of mutuality and the degree of dependence for the performance of interpersonal relationships [108, 246].

Understanding how multiple agents can work together has been an important topic in MAS for many decades. Early work with teams in AI experimented with agents sharing

their beliefs, intentions, and organizing a group into decomposed structures of sub-teams with a shared overall goal [75, 233]. That work showed how decomposing a task into sub-tasks to be solved by sub-teams (within the larger team) could achieve state-of-the-art task completion performance in team settings with simple rule-based agents. More recent work with teams in AI focuses on ad hoc teamwork or teams of multiagent reinforcement learning (MARL) agents. Ad hoc teamwork studies how multiple individual agents can coordinate their behavior in zero-shot settings [226]. MARL focuses on how multiple agents can learn to coordinate their behavior over time [30] and teams are typically explored in cooperative or competitive domains (i.e., one cooperative population or two competing teams). MARL has made significant advances in these domains by designing algorithms to guide individual agents towards learning coordinated policies [195]. However, cooperative and competitive domains offer limited opportunities for agents to learn *how* or *when* to cooperate with others [12] and typically overlooks the possibility of sub-teams with variations in their incentives.

There has been an increasing interest in the study of mixed-motive domains [115], environments that are not purely cooperative or purely competitive and as a result, agents' incentives may align some amount of the time. High levels of long term rewards in mixed-motive domains can be obtained through mutual cooperation; however, agents have the short-term incentive not to exhibit cooperative behavior. If all agents learn non-cooperative behavior, all agents receive poor outcomes. Investigating the impact of group size and structure on system stability has been argued as a way to relate AI research to findings in the natural sciences [163]. We aim to reinvigorate the study of teams in AI using individual reinforcement learning (RL) agents in contexts where agents may have different incentives. We emphasize the exploration of teams in mixed-motive domains to study how team structures (the number and size of teams among a population) influence how agents learn *when*, *how*, and *with whom* to cooperate, each concept being an important area of research in AI. We draw significant inspiration from OP in our design of team structures with individual learning agents. These settings allow us to study the impact of teams and the degree of agents' aligned goals on how agents develop their policies, as well as understand the potential side-effects and pitfalls of poorly constructed team structures. We believe this work is of interest to researchers across Game Theory, MAS, and MARL. We view our biggest contribution as being within MARL, highlighting the impact and influence that population structure has on the policies that individual agents learn. Thus, we argue that researchers and engineers should carefully consider the social structure of populations when designing learning agents within specific environments.

1.1 Thesis Statement

The hypothesis being studied in this dissertation is:

> *Teams can provide significant advantages in guiding the development of policies for individual agents that learn from experience.*

To explore this hypothesis, the remainder of this dissertation investigates the following research questions:

1. **Can organizing a population of agents into teams impact the behavior that is learned by individual agents?**
   (a) Can teams change the underlying game-theoretic properties of mixed-motive domains?
   (b) Can teams promote cooperation in the context of mixed-motive domains?
   (c) Can teams guide agents to develop globally beneficial joint policies despite mixed incentives?
2. **How do various degrees of common interest for shared goals impact the behaviour learned by agents in teams?**
   (a) Do teams impact the amount of common interest necessary for globally favorable results in different domains?
   (b) Do different levels of common interest impact game-theoretic equilibria in the context of team structures?
   (c) Can mixed incentives support agents in learning globally favorable joint policies in the context of teams?
3. **Can we derive theoretical constructs about the impact of team structures on how agents learn?**
   (a) Can teammates help agents explore and identify valuable areas of the state space?
   (b) Do team structures and shared goals impact the ability for agents to perform effective credit assignment?
   (c) Do settings exist where specific team structures may not be beneficial or desired?

1.2 Motivating Examples

We identify several real-world, motivating examples that inspired various stages of this research. These are situations where insights from this dissertation may have some impact if adapted into real-world settings, either by constructing teams or changing incentive structures among a population.

1.2.1 Wildland Fire Fuel Mitigation

Wildfires claim many lives and cost communities billions of dollars every year [183]. California's wildland fuel mitigation defensible space code (CA PRC §4291) describes required practices for removing flammable vegetation around buildings to reduce wildfire risk and maintain a defensible space for fire fighters. The code states that landowners must "maintain a defensible space of 100 feet from each side and from the front and rear of the structure, but not beyond the property line" [62]. However, the code is not able to ensure that neighbors mitigate their properties if a structure is within 100 feet of the property line, since landowners are unable to be held liable for the independent actions of their neighbors (i.e., placing a structure close to the property line).

The current state of California's defensible space code presents a social dilemma that humans currently do not solve. A neighborhood is safer if all landowners mitigate around all structures (regardless of property lines), but mitigating carries a cost for the property owner. One study found that 98% of the 686 buildings damaged in the 2018 Woolsey Fire had insufficient, but technically legal, vegetation mitigation practices [164]. Changing the vegetation mitigation code to require all buildings to be protected would disproportionately impact homeowners since those with more neighboring structures close to their property lines would incur greater mitigation costs. Furthermore, policing mitigation practices is costly due to the high cost of manual surveys and terrestrial vegetation scanning. While we have previously developed a method for less expensive widespread vegetation monitoring using deep learning and remotely sensed data [190], the incentives to mitigate properly are not currently strong enough for individuals to solve this dilemma.

Elinor Ostrom's Nobel Prize winning work in economics recognized that self-monitoring and enforcement are necessary for a group to solve social dilemmas [169]. We believe that teams can improve self-monitoring and enforcement to promote aspects of Ostrom's work. We hypothesize that teams could create incentives to mitigate risky areas of flammable vegetation that are currently considered legal under the California's defensible space code. Furthermore, teams that comprise neighborhoods and communities could better allocate

resources to optimize mitigation efforts and conform to their own localized social norms of behavior.

1.2.2 Team Sports Analytics for Invasion Games

In 2002, the Oakland Athletics adopted an approach to constructing and managing baseball teams, starting a revolution in sports analytics that was popularized by the book and movie *Moneyball* [130], using empirical statistics as a basis for roster management. Two decades later, most baseball teams employ staffs of analysts that support coaching and management decisions with quantitative data [56]. Baseball is classified as a "striking game" [57] due to its episodic and repetitive structure, allowing for relatively easily collected data that lends itself nicely to statistics. However, statistics alone are unable to fully capture the complexity of "invasion games" [57], defined by using a goal or hoop where attacks rely on invading opponent territory and players can interact anywhere on a playing surface (e.g., football (soccer), ice hockey, and basketball). Teams in invasion games rely on sub-groups of players that play together as one cohesive group under various incentive structures; thus, we posit that sports analytics for invasion games are another domain where our research could be useful. Invasion games support the advancement of multiagent research by providing enclosed, structured environments with an abundance of data collection to adapt various findings from our dissertation to the real world.

Multiagent problems in invasion game sports analytics involve several levels of complexity and time horizons. Coaches must identify players that coordinate well together, devise team-based strategies, and construct best responses during a match to outperform opponents. This requires a rich understanding of joint policies, different types of roles within a team, and the impact of group structures on player development. Managing a sports team requires long-horizon planning across multiple seasons, an environment which emphasizes the importance of modeling long-term emergent of behavior, identifying role specialization among a roster, and constructing monetary incentives to promote the emergence of certain behaviors. The contributions of this dissertation touch on all of the above aspects, emphasizing the importance of team structure, personal or group incentives, and joint policies on agents' abilities to effectively work together. In short, we believe that multiagent systems is to invasion games what statistics is to striking games, and that the research in this dissertation is one example of how concepts in these different fields can intersect.

1.2.3 Research Contributions

In this dissertation, we make the following research contributions:

1. We define a model of teams with individual learning agents for multiagent environments inspired by organizational psychology (OP) and early work with teams in AI. We show how individual reinforcement learning agents in teams can learn cooperative behavior in social dilemma environments where they have game-theoretic incentives to not to cooperate (Figures 4.1 and 4.2). In a gridworld environment, agents divided into multiple teams autonomously learn role specialization and global joint policies that achieve up to 33% more reward than the fully cooperative population (Figure 4.3 and Figures 4.5-4.8). This is significant since the fully cooperative system has previously been assumed to maximize reward in mixed-motive environments [266, 71].

2. Using our model of teams, we introduce *credo*, a model that defines how individual learning agents optimize their behavior for the goals of various groups they belong to: themselves (a group of one), any teams they belong to, and the entire system. Our results show that agents with high team-focus learn cooperation and are robust to some degree of selfishness in settings where they have the game-theoretic incentive to not cooperate (Figure 5.8). Team-focused agents learn to not be exploited by selfish agents and learn mutual cooperation with other team-focused agents in other teams (Figure 5.6). In a gridworld environment, we identify two credo parameter scenarios that achieve the highest reward (Figures 5.11, 5.12, and 5.13); when agents have high team-focus and when high system-focus agents are also slightly self-focused. Agents in these scenarios autonomously learn role specialization and efficient global joint policies that significantly outperform the fully cooperative population.

3. We provide theoretical underpinnings to further understand the conditions under which teammates may be beneficial for individual learning agents (Theorem 1), as well as scenarios where too many teammates may create settings where learning is difficult (Theorem 2). To this end, we perform an extensive empirical evaluation showing how our theoretical findings are consistent across multiple learning algorithms and environments (Figures 6.3 to 6.11).

4. We design and implement a self-tuning credo agent to autonomously discover favorable learning conditions for a defined team structure through regulating credo parameters (Figure 6.12 and Algorithm 1). Each agent acts in the environment using a low-level behavioral policy and maintains and updates their individual credo parameters using a high-level credo policy. We perform a preliminary empirical evaluation

and show how self-tuning credo agents are able to modify their credo parameters to shape their reward function and learn a joint policy that achieves more reward than their initialized configuration (Figure 6.14).

1.3 Research Area

This subsection defines the scope of research conducted as part of this dissertation and is also summarized in Figure 1.1. This research lies in the intersection of three areas: multiagent systems, reinforcement learning, and organizational psychology and multiteam systems.

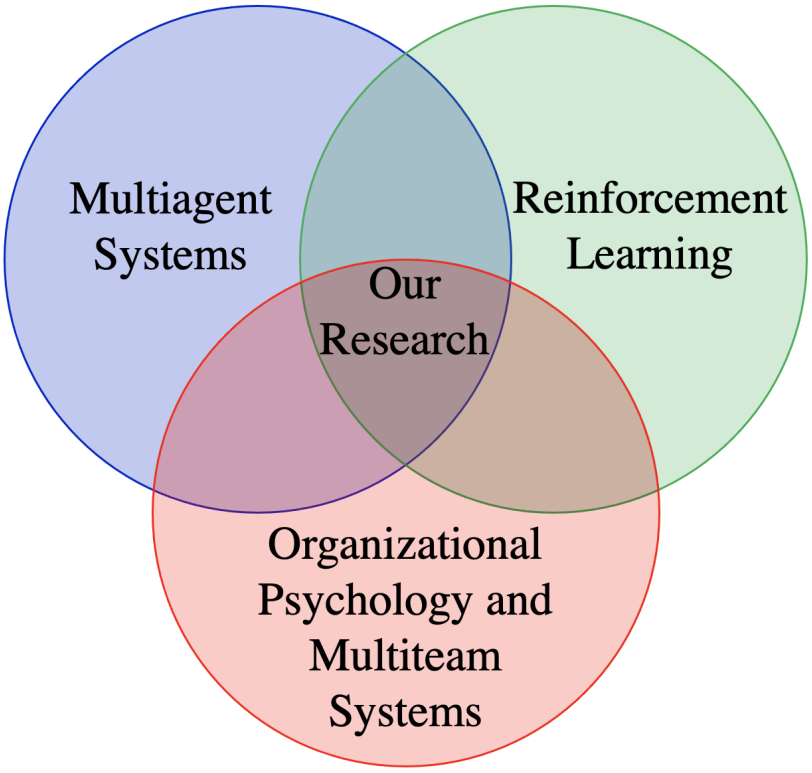

Figure 1.1: The focus of this dissertation as it relates to other fields and research topics.

Multiagent systems (MAS) is an area of research that is concerned with systems containing more than one (intelligent) agent that learns to make decisions [227]. Agents in MAS can either have static behavior or learn from experience. We consider individual agents that learn from their individual experiences using reinforcement learning (RL) [230]. RL is a field of machine learning where an agent has the goal of acting "optimally" in a dynamic environment by maximizing numerical feedback signals (i.e., rewards). For RL agents in MAS environments, the optimal policy depends on both the environment and the policies of all other agents in the system. Learning in MAS using RL has formed its own

sub-field, multiagent RL (MARL) [85]. MARL algorithms can take many forms depending on the dynamics of the underlying environment. For example, agents can work together with aligned goals, be in direct competition with conflicting interests, or have mixed incentives. Limited AI research has been conducted with multiple teams that are not in direct conflict (i.e., not in zero-sum competition). This environment is similar to settings that are studied in organizational psychology (OP) and multiteam systems (MTSs), social science disciplines that focus on how human teams are formed, constructed, and maintained [143]. Our work does not make any direct contributions to the fields of OP or MTSs; however, we aim to utilize relevant ideologies from these fields in MAS using individual RL agents. Thus, we show how team and organizational structures from OP and MTSs impact different settings in MARL and heavily influence how agents learn. Hence, our research lies at the intersection of all three of these fields of research.

1.4 Thesis Overview

The dissertation begins by detailing the related literature that our work is situated within or builds on (Chapter 2). This is followed by a series of chapters that detail our work exploring this dissertation's thesis statement and understanding the impact of teams on how individual agents learn from experience in multiagent systems.

Chapter 3: This chapter presents our model of teams for individual learning agents inspired by early work with teams in AI and OP. Furthermore, we provide an overview of all environments used in our empirical evaluations.

Chapter 4: The work in this chapter was published at the International Joint Conference on Artificial Intelligence (IJCAI) in 2022 [184]. We present an initial analysis of how the implementation of team structures into populations of individual learning agents can help agents achieve globally beneficial outcomes. We perform a game-theoretic analysis to understand how team-shaped reward functions impact the various incentives of agents playing the Iterated Prisoner's Dilemma (IPD) matrix game. Our empirical analysis provides an understanding of how RL agents learn in the context of game-theoretic incentives when divided into various team structures (i.e., different numbers and sizes of teams within the population). Our results show how agents are able to learn favorable policies of global cooperation in the IPD despite game-theoretic incentives suggesting they are better off not cooperating. In the more elaborate Cleanup Gridworld Game social dilemma environment, our results show that the population obtains significantly more reward when agents are divided into several smaller teams with mixed incentives between teams when compared to a fully cooperative population. Global reward equality is also found to remain high in environments of multiple teams with mixed incentives between teams; however, this work assumes agents' goals are fully aligned with those of their teammates.

Chapter 5: The work in this chapter was published at the Autonomous Agents and Multiagent Systems (AAMAS) conference in 2023 [185]. This chapter removes the assumption that agents are fully aligned with their teammates and studies settings where agents may optimize for various goals. We present *credo*, a model that regulates how agents optimize for multiple objectives in the presence of teams. The noun credo, defined as "the aims which guide someone's actions" [224], appropriately describes our model of how agents optimize for various goals. We perform a game-theoretic analysis with teams and credo to show how various incentives are impacted by credo parameters and the environment.

In the Cleanup Gridworld Game, we discover multiple credo configurations that achieve the highest mean population reward: when agents in a large group are slightly selfish or when agents mostly optimize their behavior for smaller teams even in situations with some amount of selfishness. High reward is achieved by agents learning efficient joint policies through division of labor, whereas the fully cooperative population fails to learn this behavior.

Chapter 6: The work in this chapter was published at IJCAI in 2023 [186] and appears in the Adaptive and Learning Agents (ALA) workshop at AAMAS in 2023 [191]. This chapter studies why, and under which conditions, certain team structures (Chapter 4) and group alignment (Chapter 5) outperform others. We derive theoretical underpinnings for teams along two lines of inquiry which help agents learn initially but eventually lead to diminishing returns. First, we show how teammates help agents identify valuable areas of the state space more easily than if they did not have teammates. Second, we show how the size of a team impacts the ability for agents to perform effective credit assignment and can create settings where it is difficult to learn – the variance in reward converges to zero, resulting in no meaningful information for agents to learn. We support our theoretical findings with an extensive empirical analysis across four different environments with three different types of learning algorithms. Our empirical results with learning agents align with our theory, showing how team reward, role specialization, and learning characteristics are impacted across all environments and algorithms in our evaluation. Motivated by our theoretical and empirical findings, we conduct preliminary work on the design and implementation of self-tuning credo agents that are able to modify their own credo parameters in a decentralized system. These agents are able to receive the benefits of teams while recovering stronger reward signals in settings where learning is challenging. A preliminary evaluation shows how these agents are able to autonomously discover efficient joint policies despite being initialized with known sub-optimal credo parameters given a specific team structure.

Chapter 7: This chapter presents an overview and conclusions about the work in this dissertation. We revisit our motivating examples to discuss how the insights provided in this work might be used as an approach to study interesting problems in those domains. We discuss the broader implications and ethical considerations of our work and make connections between concepts in our work and the natural world. Finally, we present direct short-term and broader long-term possibilities for future work.

Chapter 2

Background and Related Work

This chapter presents important background material and related research for this dissertation. We begin by introducing features of reinforcement learning (RL) in the single agent setting; specifically, Markov Decision Processes (MDPs), value functions, and types of RL algorithms. Next, we introduce Deep RL and present examples of existing Deep RL algorithms. We use agents that learn using various RL and deep RL algorithms throughout the evaluations in Chapters 4, 5, and 6 to understand how teams impact different types of learning algorithms. We then shift to the multiagent setting and introduce stochastic games and existing multiagent RL (MARL) approaches that we use throughout this dissertation. We include background on organizational psychology (OP) and multiteam systems (MTSs) that motivate our research using AI agents, and discuss the history of teamwork in multiagent systems that is related to this dissertation. This chapter lays the groundwork for subsequent chapters. The learning algorithms we use in this research are presented here, but we leave implementation specifics to the following chapters.

2.1 Reinforcement Learning (RL)

Reinforcement learning (RL) is a field of machine learning that deals with an agent learning what to do, by mapping situations to actions with the goal of maximizing a numerical reward signal [230]. The learning agent is not told which actions to take, but must instead discover the actions that yield the most reward through their execution over discrete timesteps $t \in \mathbb{N}$ in an environment. We implement agents using RL algorithms throughout this dissertation; therefore, we provide specific underlying details about how learning works

with various types of RL. In single-agent RL, an agent operates in a sequential decision-making environment modeled as a Markov Decision Process (MDP) [230]. An MDP can be formally defined as follows:

Definition 1. *A Markov Decision Process (MDP) is defined as* $\langle S, A, R, P, \gamma \rangle$ *where* S *is the set of states,* A *is the set of actions the agent can take,* R *is the reward function for the agent,* P *is the transition function that transitions the agent to a new state following an action in a previous state, and* $0 \leq \gamma < 1$ *is a discount factor to discount future rewards more than near-term rewards.*

At each timestep, the agent takes some action $a \in A$ while being in state s. The agent transitions to a new state s' with some probability $P(s, a, s')$ and collects reward $R(s, a) \in \mathbb{R}$. The reward function R represents the goal of the RL problem; thus, the reward $R(s, a)$ defines what are the good and bad state-action transition events for the agent [230]. The behavior of an agent can be represented by a *policy* $\pi : S \to \Delta(A)$, where $\Delta(A)$ denotes the space of all probability distributions over the agent's action space. A trajectory up to time t is defined as a collection of state-action pairs $\{(s_0, a_0), \cdots, (s_{t-1}, a_{t-1})\}$.

An agent's *value function* specifies the long-term value of being in a certain state (i.e., the total amount of discounted future reward an agent can expect to accumulate from that state, or *return* [230]). While the reward $R(s, a)$ provides immediate feedback of taking an action a in state s, the value function accounts for the states that the agent is likely to visit after state s. Thus, a state may yield low reward for any action but have a high value in the case that the following states typically yield high rewards. The agent learns to select actions based on this value estimate of a state instead of the explicit rewards because actions with high value will return the highest reward over the long run [230].

The value function of state s under policy π, $v_\pi(s)$, is the expected return when starting in state s and following π thereafter. This can be formally defined at any timestep t as:

$$v_\pi(s) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S^t = s \right], \tag{2.1}$$

for all $s \in S$. While $v_\pi(s)$ calculates the value of a particular state, we can also calculate the value of a specific action a from a particular state s under policy π. This is denoted by a Q-value, $Q_\pi(s, a)$, and is defined as the expected return from taking action a in state s and following π thereafter, calculated by:

$$Q_\pi(s,a) = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | S^t = s, A^t = a \right], \tag{2.2}$$

The Q-values of all state-action pairs is known as the *action-value function* for policy π [230]. The functions $v_\pi(s)$ and $Q_\pi(s,a)$ can be estimated from experience while an agent is following policy π by maintaining estimates over state values and state-action value combinations.

One policy π is considered to be *better* than another policy π' if it's expected return is greater than or equal to that of π' for all states (i.e., $v_\pi(s) \geq v_{\pi'}(s) \forall s \in S$) [230]. In finite MDPs, where S, A, and R all have a finite number of elements, there always exists at least one policy that is better than or equal to all other policies, called the *optimal* policy. The goal of the learning agent in a finite MDP is to learn this optimal policy π^*. The optimal policy has the optimal state-value function, defined as $v^*(s) \doteq \max_\pi v_\pi(s)$ for all $s \in S$, and the optimal action-value function, defined as $Q^*(s,a) \doteq \max_\pi Q_\pi(s,a)$ for all $s \in S$ and $a \in A$. For any state-action pair, we can write $Q^*(s,a)$ in terms of $v^*(s)$:

$$Q^*(s,a) = \mathbb{E}\left[R^{t+1} + \gamma v^*(S^{t+1}) | S^t = s, A^t = a \right]. \tag{2.3}$$

Convergence to an optimal value function is guaranteed in the single-agent case [230]; thus, the value function $v^*(s)$ and state-action value function $Q^*(s,a)$ are able to be directly solved. Once an agent has v^*, the optimal policy can be extracted by selecting the action with the highest expected discounted future return by solving:

$$v^*(s) = \max_a \mathbb{E}\left[R^{t+1} + \gamma v^*(S^{t+1}) | S^t = s, A^t = a \right]. \tag{2.4}$$

Solving Equation 2.4 requires calculating the expected return for each action in A; however, if the agent already has the optimal state-action value function, it can simply select the action a that maximizes $Q^*(s,a)$ without needing to explicitly solve Equation 2.4 in state s. Thus, the state-action values $Q^*(s,a)$ cache the values of Equation 2.4 to be available regarding each state-action pair at any timestep. While maintaining the state-action function Q^* requires additional cost of modeling state-action pairs instead of just states, the agent does not need to know anything about the environment's dynamics and successor states to select optimal actions and construct π^*. Algorithms that maintain estimates of value and act accordingly are called *model free*, whereas algorithms that model the transition probabilities and build a world model are called *model based*. We use model free algorithms throughout this dissertation.

Another main distinction between types of RL algorithms is *on-policy* or *off-policy* algorithms. On-policy algorithms are those that attempt to evaluate or improve the policy that is being used to make decisions [230]. In control domains like in MDPs, the state-action value $Q_\pi(s, a)$ must be estimated for the current behavior policy π and all states and actions in the environment. One example of an on-policy RL algorithm is SARSA, a temporal difference control algorithm that estimates Q-values using tuples of state, action, reward, next state, and π's next chosen action (i.e., on-policy). Off-policy algorithms evaluate or improve a policy different from the one that is being used to generate the data the policy learns from in the environment [230]. To learn the optimal policy, off-policy algorithms can use two policies: one that learns from data and ultimately becomes the optimal policy (i.e., a *target* policy) and one that is more exploratory and generates various types of state-action pairs to be used for learning (i.e., a *behavioral* policy). Since the data used for training the target policy comes from the behavioral policy, this type of algorithm is considered *off-policy*. We use both on-policy and off-policy methods in this dissertation to analyze the impact of teams with both types of learning algorithms.

2.1.1 Exploration vs. Exploitation

One of the unique underlying features of RL compared to other types of learning is the trade-off between exploration and exploitation of the state space. The goal of the agent is maximize it's sum of discounted future rewards; however, unless the agent is already following π^*, choosing actions that the agent believes returns the highest reward may cause it to forego alternative actions that are more valuable. To discover the actions that result in the most reward, the agent must *explore* the set of actions and experience their outcomes; however, pure exploration does not help the agent obtain high rewards. Thus, the agent must also exploit it's acquired knowledge regarding the value of actions some amount of the time to gain rewards in the environment. Neither pure exploration or pure exploitation can be done by the agent without failing at the task of maximizing rewards [230].

This dilemma has been studied for decades without resulting in any clear solution. Currently, several exploration algorithms are commonly used to assist the agent in balancing the exploration-exploitation trade-off. On-policy and off-policy RL algorithms inherently deal with exploration differently. Off-policy algorithms learn from data collected using different policies that can be very exploratory, whereas on-policy algorithms learn from the data generated by the current policy and being very exploratory can be costly to learning. One of the most common exploration strategies regardless of learning algorithm is ϵ-greedy exploration, where the agent selects $\max_a Q_\pi(s, a)$ with probability ϵ, otherwise it

selects a random action. Other strategies rely on state visitation frequency or exploration entropy to increase an agent's exploration [230]. We use ϵ-greedy exploration in our RL algorithm implementations throughout this dissertation and draw comparisons between the exploration-exploitation trade-off and how an agent's teammates' behaviors promote exploration in Chapter 6.

2.1.2 *Q*-Learning

Q-learning is a popular model-free RL algorithm that maintains a state-action value function $Q_\pi(s, a)$ for every state-action pair (i.e., Q-learning is value-based). We use Q-learning in Chapter 6 to understand how teams impact simple RL agents in domains with small state spaces. An agent using a Q-learning algorithm operates in an MDP and takes an action a in a state s at each timestep t. Q-learning is easily implemented in single-agent settings with discrete state and action spaces, where $Q_\pi(s, a)$ is maintained in a Q-table of size $|S| \times |A|$ and each value for a state and action is called a Q-value. The Q-values are iteratively updated using the following rule:

$$Q_\pi^{t+1}(s, a) \leftarrow Q_\pi^t(s, a) + \alpha \left[(r_t + \gamma \max_a Q_\pi^t(s', a)) - Q_\pi^t(s, a) \right], \tag{2.5}$$

where $\alpha \in [0, 1]$ is the learning rate. The Q-learning update is similar to that of SARSA, the on-policy algorithm discussed in Section 2.1.1; however, Q-learning uses the maximum Q-value for the next state s' instead of the one chosen by the current policy π. Convergence is guaranteed to a fixed point of optimal state-action value function $Q^*(s, a)$ when α satisfies the following rules over timesteps t of learning [98]:

$$\sum_{t=0}^{\infty} \alpha^t = \infty \qquad\qquad \sum_{t=0}^{\infty} (\alpha^2)^t < \infty. \tag{2.6}$$

The policy at this fixed point is considered the optimal policy π^* by selecting the action with the maximal Q-value at each state. It is also common to use a different policy for action selection than the one being constantly updated; thus, agents often use a *target* policy to decide actions making Q-learning an off-policy algorithm as defined in Section 2.1.1.

2.1.3 Policy Gradients

Q-learning optimizes a value function and takes actions that will result in the maximum expected discounted reward – the policy is directly extracted from the value function. Policy gradient (PG) algorithms are another type of RL optimization that updates the policy directly using stochastic gradient descent (SGD). Thus, these algorithms are not limited to a tabular representation of the policy (like Q-learning) and can act in high dimensional action and state spaces. We use policy gradient algorithms throughout our empirical evaluations.

The parameters of a PG policy are updated by:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim\rho^{\pi_\theta}, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q_{\pi_\theta}(s,a)], \tag{2.7}$$

where $J(\pi_\theta)$ represents the expected sum of discounted rewards following policy π_θ (parameterized by θ), $s \sim \rho^{\pi_\theta}$ represents sampling a state s from a distribution generated though actions sampled from policy π_θ ($a \sim \pi_\theta$), and Q_{π_θ} represents the action-value function of the policy π_θ.

2.2 Deep Reinforcement Learning

Due to scaling difficulties in state and action spaces, RL algorithms were typically limited to simple environments. For example, the Q-table in Q-learning scales linearly with the number of states and is polynomial in the number of actions available at each state (since the Q-table has size $|S| \times |A|$). To expand to larger and more difficult environments, RL algorithms needed a way to approximate these spaces.

Deep neural networks are proven to be universal function approximators given enough depth or width of neurons [91]. In the context of RL, deep neural networks have been adopted to approximate the Q-values at any state (removing the tabular constraint) or directly represent the action-taking policy given a state. The space complexity of a neural network is constant in the number of possible states it could experience; thus, function approximation using a neural network has allowed RL to scale to environments with high-dimensional state and action spaces. The integration of deep neural networks into the RL pipeline created deep reinforcement learning (deep RL), and is responsible for many significant breakthroughs in recent years (detailed below). In this dissertation, we implement most of our agents using deep RL algorithms to expand to environments with large

state spaces. This allows us to analyze the impact of teams across more complex learning environments.

2.2.1 Deep Q-Networks (DQN)

One of the simplest deep RL algorithms is the value-based method inspired by Q-learning in Section 2.1.2 – Deep Q-Networks (DQN) [153]. We implement agents using DQN in Chapters 4, 5, and 6 of this dissertation. Similar to Q-learning, a DQN agent operates in an MDP composed of states, actions, and rewards; however, the DQN implementation introduced several key methods that were necessary to address learning instabilities and distribution shifts for DQN to learn effectively. First, they present the idea of *experience replay* – stashing previous experiences in a replay buffer for the network to learn from to reduce changes in the data distribution (i.e., the networks train on a *minibatch* of past experiences at each training iteration, chosen randomly). Second, similar to how Q-learning uses two policies, they introduce the idea of updating the Q-value estimations from the original (*training*) network (i.e., policy π with parameters θ) towards an identical secondary *target* network to stabilize the learning procedure (i.e., policy π^- with parameters θ^-). Both networks take the state as input and estimate the Q-value of all possible actions; however, only the training network trains regularly and the parameters of the target network are replaced with the parameters of the training network at regular intervals (i.e., every C environmental steps). Lastly, the original DQN implementation uses a convolutional neural network (CNN), a specific kind of neural network typically used for images and high-dimensional data. This detail is specific to the original implementation domain [153]; however, the DQN algorithm is general and can be implemented with any type of deep network. In our implementation, we use a multi-layer perceptron (MLP) given one-dimensional states instead of images [73].

During learning, the DQN algorithm minimizes the mean squared error (MSE) between the training network Q-value estimate and the target network Q-value estimate to compute a *loss*. Let K represent the number of samples selected from the replay buffer as a *minibatch* to train on. The loss is calculated by:

$$L(\pi) = \frac{1}{K} \sum (r + \gamma \max_a Q_\pi(s', a) - Q_{\pi^-}(s, a))^2, \tag{2.8}$$

where r is the observed reward for taking action a in state s, γ is the discount factor, and s' is the next state. Since DQN uses the target network for action selection while updating the training network, it is known as an *off-policy* algorithm. It is also considered

a model-free algorithm since it updates the value estimation of actions instead of building an explicit model of the world. The original implementation of DQN showed that it could achieve results comparable to humans on Atari games from only pixel inputs using a CNN deep network (i.e., 210 × 160 pixel images at 60 Hz) [153]. This implementation was a significant achievement in scaling RL to higher-dimensional state and action spaces.

2.2.2 Trust Region Algortihms

Unlike DQN, deep policy gradient algorithms optimize the policy directly and adapt the policy gradient update in Equation 2.7 to update the weights of a neural network. Trust region algorithms are a sub-class of deep RL algorithms that optimize the policy directly to some new policy that is not significantly different from the old policy (i.e., within a trusted region of the policy space).

Trust Region Policy Optimization (TRPO) [212] is one of the most popular trust region algorithms. While DQN is an off-policy value-based RL algorithm, TRPO is a policy-gradient RL algorithm that is considered on-policy. The overall objective of TRPO is to limit the search of parameter updates for some new policy π to a trusted region within the policy space by restricting the length of the update step size α. TRPO theoretically guarantees monotonic improvement on the policy given non-trivial step sizes. The loss of TRPO is similar to the policy gradient loss function in Equation 2.7. We write the TRPO optimization function using the notion of an objective function that is maximized instead of minimizing a loss for easier comparison with the next algorithm. At timestep t, TRPO aims to solve the following constrained optimization problem:

$$\begin{aligned} &\underset{\theta}{\text{maximize}}\ \hat{\mathbb{E}}^t \left[\frac{\pi_\theta(a^t|s^t)}{\pi_{\theta_{old}}(a^t|s^t)} \hat{A}^t \right] \\ &\text{subject to}\ \hat{\mathbb{E}}^t \left[D_{KL}[\pi_{\theta_{old}}(\cdot|s^t)||\pi_\theta(\cdot|s^t)] \right] \leq \delta, \end{aligned} \tag{2.9}$$

where θ_{old} is the vector of policy parameters before the update, θ is the vector of policy parameters after the update, D_{KL} is the Kullback-Leibler (KL) divergence, δ is in $\mathbb{R}^+$, and $\hat{A}^t$ is the advantage function, defined as the difference between a policy's Q-value estimate given a state-action pair (parameterized by θ) and the state-value function estimate for a given state (i.e., $A_\pi(s,a) = Q_\pi(s,a) - V_\pi(s)$). The theory behind TRPO actually suggests using a penalty β on the KL divergence and solving an *unconstrained* optimization problem (instead of using δ); however, choosing a single value of β is challenging in practice. We

revisit the learning features of TRPO in the context of our broader discussion in Chapter 7.

Proximal Policy Optimization (PPO) [214] is an on-policy extension of TRPO that only needs to solve a first-order approximation instead of TRPO's second-order method (due to the KL divergence). We implement agents using PPO across many empirical evaluations in Chapters 4, 5, and 6 in complex environments with high-dimensional state spaces. PPO constrains the ratio between the old and new policies without imposing hard constraints on the optimization problem or using the KL divergence distance. Let $r^t(\theta) = \frac{\pi_\theta(a^t|s^t)}{\pi_{\theta_{old}}(a^t|s^t)}$. TRPO maximizes a "surrogate" objective $\hat{\mathbb{E}}[r^t(\theta)\hat{A}^t]$ with the constraint that the new update is less than δ away from the previous policy. PPO modifies this objective to penalize changes to the policy that move $r^t(\theta)$ away from 1, making the PPO objective:

$$L^{CLIP}(\theta) = \hat{\mathbb{E}}^t \left[\min(r^t(\theta)\hat{A}^t, \text{clip}(r^t(\theta), 1-\epsilon, 1+\epsilon) \right], \tag{2.10}$$

where ϵ is a hyperparameter. The second term in Equation 2.10 modifies the surrogate objective by clipping the policy update so that the next policy π_θ is not significantly different from the previous policy $\pi_{\theta_{old}}$. This clipping removes any incentive for moving r^t outside of the interval $[1-\epsilon, 1+\epsilon]$ to ensure the next policy is within some trusted region. PPO has been shown to be easier to implement and faster to train than TRPO with similar performance in many environments [214]. Thus, PPO has emerged as a popular policy gradient algorithm that is widely used in RL implmentations today.

2.3 Multiagent Reinforcement Learning (MARL)

Sections 2.1 and 2.2 introduced underlying RL algorithms and concepts behind many single-agent RL implementations. While we implement agents using various algorithms presented in Sections 2.1 and 2.2, our work is concerned with settings where there are multiple learning agents within the same environment (i.e., multiagent environments). The theory behind learning with single-agent algorithms typically assumes satisfaction of the Markov property (i.e., environments are assumed to be stationary) which does not hold in multiagent environments. Multiple agents in the same environment make learning in these settings inherently more complicated. Converging to optimal policies in multiagent RL (MARL) relies on both the environment and the strategies or behaviors of all other agents in the system. This implies that the environment dynamics appear to be nonstationary

from the perspective of a single agent (i.e., taking the same action from the same state can result in different outcomes).

In settings where agents are assumed to be cooperative, the single-agent MDP formulation can be extended to a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [217]. While the Dec-POMDP setting offers a framework for decentralized *cooperative* decision making among multiple agents, our work in Chapters 4, 5, and 6 considers domains beyond pure cooperation; specifically, *stochastic games* with mixed incentives. A stochastic game is a strategic generalization of both MDPs and repeated games from game theory that models the interactions of multiple agents. The optimal policy of an agent in a stochastic game depends on the policies of all other agents in the environment.

Definition 2. *A stochastic game is defined by* $\langle \mathcal{N}, S, \{A\}_{i \in N}, \{R\}_{i \in N}, P, \gamma, \Pi \rangle$ *where* $\mathcal{N}$ *is the set of agents with size* $N \in \mathbb{N}$*,* S *is a finite set of states, and* $A = A_1 \times \ldots \times A_N$ *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* $R_i : S \times A \times S \mapsto \mathbb{R}$*.* $P : S \times A \mapsto \Delta(S)$ *represents a transition function which maps a state and joint action into a next state with some probability and* γ *represents the discount factor so that* $0 \leq \gamma < 1$*.* Π *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.*

A common assumption in stochastic games is that all agents are able to observe the finite set of states S (i.e., no agent can be excluded from observing any state in S). At each timestep of the game t, agent i observes the current state $s_i^t \in S$ and takes a local action $a_i^t \in A_i$ (note that agents may still be partially observable). Agents receive a reward r_i^t according to their reward function R_i, the joint state $\mathbf{s}^t$, and the joint action of all agents $\mathbf{a}^t$ at time t (we use bold to denote vector notation, i.e., the joint state or action for all agents). The transition function P transitions the environment to the next joint state $\mathbf{s}^{t+1}$, conditioned on the previous joint state $\mathbf{s}^t$ and joint action $\mathbf{a}^t$. Given a stochastic game, a global *joint policy* is defined as $\boldsymbol{\pi}_N = \{\pi_1, \cdots, \pi_N\}$, where π_i represents the stochastic policy followed by agent i. In Chapters 4, 5, and 6, we define joint policies for a team, a subset of agents, or the entire population following the same logic as above.

Stochastic games operate with a temporal dimension captured through timesteps of the game. One can model each timestep of a stochastic game as a single *stage game* where agents select an action and receive some payoff which depends on the joint action of all agents $\mathbf{a}^t$. One class of stage games are two-player *matrix* games where the payoffs of either agent depend on the actions taken by both agents. The game-theoretic incentives of an

agent playing a stage game can be modeled using the payoffs and their opponent's strategy profile. The reward function of a stochastic game can be defined to have specialized characteristics at the stage game level or across long horizons that define certain types of solution concepts. The formulation of reward functions where the structure of environmental rewards are dependant on the context of other agents' actions is called a *general sum* model. The *Nash Equilibrium* of a general sum stochastic game is known as a stable point in the joint policy space – the point at which no player will unilaterally be incentivized to deviate from their strategy, coinciding with the minimax solution in zero-sum two-player games [159, 201].

Definition 3. *Nash Equilibrium*
A Nash Equilibrium of a game between N players is a strategy profile $A = \{a_1, \ldots, a_i, \ldots, a_N\}$ with the property that no player i can do better by choosing an action different from a_i, given that every other player j will keep selecting a_j.

While the Nash Equilibrium is one solution concept of stochastic games, there may be multiple Nash Equilibia that create coordination challenges or mixed incentives between agents. Another solution concept of a stochastic game is one that maximizes the number of players with *maximal* payoff. This converges to a joint strategy which is desirable if no player can improve their payoff without decreasing the payoff of others. This is defined as a Pareto Optimal outcome.

Definition 4. *Pareto Optimality*
Let $u_i(X)$ represent the payoff for player i under strategy profile X. A strategy profile $A^* = \{a_1^*, \ldots, a_i^*, \ldots, a_N^*\}$ is Pareto Optimal if there is no other strategy profile $A = \{a_1, \ldots, a_i, \ldots, a_N\}$ such that $u_i(A) \geq u_i(A^*) \forall i$.

Agents converging to Nash Equilibria or Pareto Optimal strategies may result in different global outcomes depending on the dynamics of the environment (i.e., Nash Equilibria may lead to unfavorable global results). For example, stochastic games are not limited in the structure of their reward function and can have either competitive, cooperative, or mixed-motive dynamics. Competitive stochastic games are those with conflicting interests between agents, where one agent's payoff is associated with a decrease in others' payoffs, whereas cooperative games correspond with a mutual increase in payoffs between agents. Games with mixed motives are those in which agent interests are, to varying extents, sometimes aligned and sometimes in conflict [39]. Our work mostly considers mixed-motive stochastic games; however, we provide a brief background on existing work in each setting.

2.3.1 Multiagent Work in Competitive Environments

State-of-the-art successes in deep learning have fueled some recent progress in MARL for competitive domains. One class of competitive games are *zero-sum*, where the sum of the global joint reward is equal to zero (i.e., one agent's gain is another agent's loss). Two-player zero-sum board games where MARL has achieved superhuman performance are Chess [223, 119], Go [223], and Shogi [211]. While these advances operate with only a single opponent, the game can technically be considered multiagent due to the existence of an opponent influencing the environment strategically. Competitive settings can also support opportunities for coordination between agents in competitive teams. Two-*team* zero-sum games where teammates are bound together with pure-common interest have been studied, also achieving high performance in environments such as Starcraft [251], Capture the Flag [99], Dota II [18], hide-and-seek [13], Robot Soccer [111], and Honor of Kings [267]. These environments represent games of pure zero-sum competition between teams, where one team's gain is another team's loss. While coordination exists within a team, groups of agents are in direct competition. We highlight this work to show that multiagent teams have been explored in competitive contexts; however, our work focuses on multiagent teams in settings that are not purely competitive.

2.3.2 Multiagent Work in Cooperative Environments

Another type of setting is a fully cooperative domain, where agents aim to coordinate their behavior towards a common goal [37]. Agents in these environments are always assumed to be cooperative; thus, several algorithms have been developed to reduce nonstationarity and improve coordination that often assumes internal access to all agents in the system. This includes algorithms where agents share a centralized critic network for value estimations [135, 66], network parameters [77], or gradient updates [268]. Other approaches have endowed agents with the ability to communicate to reduce uncertainty, in both centralized and decentralized settings [65, 131]. Another approach is to help a group coordinate through the centralized training decentralized execution (CTDE) methodology [166, 114] where value decomposition of a joint reward signal has gained popularity [195, 69, 174]. Value decomposition algorithms divide a group's reward among agents conditioned on their marginal contribution, and have been shown to be effective to help agents learn coordination [195, 69, 174]. This value decomposition function is typically learned using supervised learning during a centralized pre-training phase; thus, these algorithms rely on a cooperative population and a separate centralized pre-training phase which may not always be possible.

Many existing MARL algorithms with cooperative populations suffer from the issue of scalability since they rely on information from all agents. Therefore, another MARL modeling approach is independent RL, where agents learn their own policy and assume other agents are simply a part of the environment [236]. No information sharing is done between agents to allow for independent updates and increased scalability to large environments with many agents. Independent PPO learners have been shown to perform as well as network or parameter-sharing PPO implementations in cooperative environments [269]. Some theoretical work has been done to bound convergence properties with policy gradient algorithms in specific settings [51]; however, convergence with independent learning agents it is not always guaranteed and depends on exploration actions among the population [37].

In Chapters 4, 5, and 6, we model agents as individual RL agents to maintain individuality between their decisions and learning updates. This allows us to study emergent aspects such as individual policy development, role specialization, and the joint policy learned across a population.

2.3.3 Multiagent Work in Mixed-Motive Environments

While competitive and cooperative settings offer environments that are easy to benchmark performance or assess coordination capabilities, they offer no opportunity for agents to learn how to cooperate in settings with mixed incentives. A third class of multiagent games, and the main ones considered in this dissertation, are those with mixed-motives where agent interests are sometimes aligned and sometimes in conflict, often to varying extents [39]. A main challenge is that agents should not only cooperate (as in cooperative domains) to avoid being exploited, but instead understand *when* and *how* to cooperate. Developing agents within the scope of "Cooperative AI" has gained popularity to address the important and challenging problems ahead [38]. Our work is motivated by the focus on Cooperative AI to study how multiagent teams influence game-theoretic incentives and learned behavior in mixed-motive domains.

Mixed-motive domains present scenarios where the Nash Equilibrium and Pareto Optimal outcomes are different. The resulting environmental dynamics put strain on agents' incentives to optimize their individual utilities to the detriment other agents, inducing a social dilemma. Nash Equilibium strategies can result in poor long-term results for both the individual agent and the entire system. Thus, mixed-motive environments offer settings where the long-term rational behavior for the population is not converging to the short-term Nash Equilibrium strategy. A successful population should converge to some alternative strategy rather than acting on their short-term incentives (that are defined by

	Cooperate	Defect
Cooperate	R, R	S, T
Defect	T, S	P, P

Table 2.1: An example of the two-player Prisoner's Dilemma. T, R, P, and S are payoffs for a *row* and a *column* player who simultaneously choose to "cooperate" or "defect", and receive the payoffs according to the result of the joint action. $T > R > P > S$ is required to make this game a social dilemma.

the Nash Equilibrium). A common assumption in the MARL literature is that aligning all agents' reward functions towards a common goal (i.e., turning the environment into a cooperative system) will achieve the highest reward in mixed-motive environments [266, 71]. We also compare our results with the fully cooperative system; however, in contrast with past work, we find that agents in teams can achieve significantly more reward than the aligned population.

We position our work in the context of mixed-motive stochastic game environments with individual learning agents. Although convergence to the Nash Equilibrium is a popular solution concept in multiagent domains with cooperative or competitive dynamics [225], we mostly explore domains where agents' incentives can be mixed across a population or have various degrees of alignment. Therefore, convergence to a Nash Equilibrium in these settings yields poor outcomes. Since many of our evaluation settings contain underlying mixed-motive social dilemmas, we provide deeper background in social dilemma domains and existing methods that allow a population of agents to coordinate their behavior. While we emphasize social dilemma domains here, we note that the entirety of our work in this dissertation is not limited to social dilemmas.

Social Dilemma Environments

For normal-form games in game theory, the behavior of an agent can be generally thought of in the context of "cooperation" (C) and "defection" (D). A social dilemma is a situation in which an individual receives a higher payoff for defecting than cooperating, but all are better off if all cooperate than if all defect [26, 42]. The failure to solve social dilemmas comes from acting in a way which seems individually rational by optimizing short-term payoffs, but leads to agents being worse off than if they chose an action with a potentially lower payoff. Social dilemmas can take many forms, including tragedies of

the commons [170], collective risk dilemmas [150], and the Prisoner's Dilemma [194]. Real-world social dilemmas include scenarios of abstaining from benefit or incurring a cost for the benefit of others, such as maintaining common-pool resources [25], donating money [42], and wildfire fuel mitigation [164]. Wildfire fuel mitigation is one of our motivating examples listed in Chapter 1. Social dilemmas are difficult for AI learning agents to solve without additional infrastructure since solving them requires an emphasis on long-term planning and understanding the benefits of cooperation.

One of the most well studied social dilemmas is the two-player matrix game called the Prisoner's Dilemma, shown in Table 2.1. There exists a *row* and a *column* player, where the row player selects the action corresponding to the rows of the matrix and the column player selects actions corresponding to the columns. The variables T, R, P, and S represent payoff values that both players would receive depending on the joint action of the two agents. The payoff scheme $T > R > P > S$ is required to make this a social dilemma. Agents individually choose to either cooperate (C) or defect (D) and receive the payoffs according to Table 2.1. The joint strategy of row and column players is represented by a tuple (row, column), such as (C, C) for mutual cooperation.

When agents play the Prisoner's Dilemma repeatedly a finite number of times, known as the Iterated Prisoner's Dilemma (IPD), the Nash Equilibrium is (D, D). Both players should defect to avoid the S payoff, which results in P, the second lowest available payoff. The IPD has three Pareto Optimal strategy profiles, namely (C, C), (C, D), and (D, C), and is considered a social dilemma due to existence of some outcome (C, C) which is strictly better for all agents than the unique Nash Equilibrium of (D, D). The RL methods that have achieved success in two-player zero-sum games have subsequently failed to achieve cooperative policies in repeated social dilemmas, ultimately leading to low return [3, 12, 40]. Therefore, solutions which help RL agents solve social dilemmas typically rely on decentralized and centralized coordination mechanisms with various assumptions.

We highlight relevant decentralized and centralized approaches in the next two subsections to help position our work on multiagent teams. Specifically, we position the work in this dissertation between centralized and decentralized systems. The following chapters will show how teams 1) allow for more agency and adaptability than centralized systems through autonomous role specialization, and 2) provide more learning stability than decentralized systems by removing the need for prior definitions or modified social networks.

Method	Summary	Details	Limitations
Punishment norms	Retaliation against defecting agents.	Negative reward [113, 125].	Needs consensus on punishable behavior.
Reputation norms	Labeling based on behavior.	Markovian [264], non-Markovian [205].	Bottom-up reputation may be interpreted differently by agents.
Payments	Pay agents to cooperate.	Reward payments [266], payment homophily [52].	Generates new reward from nothing.
Modify social connections	Change social networks.	Partner selection [4], remove social connections [70].	May not be possible in practice.

Table 2.2: Summary of decentralized systems for social dilemmas.

Decentralized Systems for Social Dilemmas

Research on decentralized systems in social dilemma domains that we discuss here are summarized in Table 2.2. Research focused on decentralized mechanisms in mixed-motive domains to sway behavior has roots in economics [169, 25] and evolutionary game theory (EGT) [207, 206]. This work often highlights social norms, punishment, and emergent institutions that improve welfare as the mechanisms behind emergent cooperation. Social norms are commonly known standards of behaviour of how individuals ought to behave in a given situation and are the underlying mechanisms behind many decentralized systems for cooperation [61, 59, 20]. While the extensive work on decentralized systems has shown breakthroughs by promoting emergent cooperative behavior, our work shows how team structures can allow agents to autonomously learn specialized roles within their teams that benefit the overall community. We find that the structure of teams in the population (Chapters 4 and 6) and agents' alignment with the goals of various groups (Chapter 5), significantly impact how these roles are developed.

EGT simulates generations of agents with simple strategies that are updated through a dynamic learning process, where better performing strategies are more likely to be imitated

by others [222]. Despite its simplicity, EGT is known to closely imitate human group behavior and suggests that defective behavior will overtake any society once introduced into a population; however, various mechanisms can inhibit this result [222]. Specifically, enforcing norms through punishments have been found to lead to cooperation in human studies [24, 59], then in EGT [207, 206], and also in AI [3, 125]. Synthesizing multiple norms together to decrease the size of the set of norms for an agent to track has been the focus of some previous work [140, 154, 64]. However, other work has found that adding *more* rules with decentralized punishment mechanisms improves the ability for MARL agents to learn compliance, even if the additional rules do not improve welfare directly [81, 113].

In the absence of direct punishment, reputation mechanisms have been shown to help agents avoid being exploited by defectors [84], but these require high rates of participation and truthful reporting [106, 197]. One example of a social norm is the process of assigning reputation to agents in settings of indirect reciprocity [264] or as a third party agent observing an interaction [60]. This norm can include Markovian (based on most recent information only) or non-Markovian levels of complexity, though Markovian has been shown to be sufficient through experimentation [205]. However, for reputation to effectively lead to cooperation in settings with multiple interacting agents, the method by which it is assigned must be known by all agents, defined system-wide *a priori*, or be one of few existing mechanisms [221, 248, 264]. Converging to a single reputation norm using a bottom-up approach has been shown to be difficult in MARL due to agents understanding reputation differently [3]. One aspect of teams we highlight in Chapters 4 and 5 is the ability for agents to converge to specific roles in their team. This indicates agents individually arrive at similar representations of the role requirements for a successful group. However, like norms with decentralized populations, convergence to these roles can be difficult with sub-optimal team structures (Chapter 6). While the mechanism behind punishment or norms needs to be explicitly shared for the bottom-up emergence of cooperative behavior, our work shows how teams allow for the emergence of roles and cooperative behavior autonomously from only a modified reward function that can promote globally favorable results.

Giving agents the ability to make costly reward payments to other agents has been shown to increase population productivity in some of the same mixed-motive domains we explore in Chapters 4, 5, and 6 [266]. This allows agents to be strategic in how they make payments and incentivize their peers; however, their work assumes that agents can make reward payments that generate *more* reward in the environment than what the paying agent has previously received (i.e., new rewards are generated from nothing). While that work increases population productivity in the social dilemma environment, it has been shown to induce a second-order social dilemma in how agents make payments [52] (i.e., agents

Method	Summary	Details	Limitations
Population reward access	Agents condition rewards on all other agents.	Global reward sharing [145, 71], inequity aversion [96], group-based reward [55].	Relies on pre-training phase, access to all agents.
Community pleasing	Takes actions based on information from other agents.	Agreed upon action [15], action pleases peers [117].	Relies on truthful reporting, access to all agents' opinions.
Shared learning mechanisms	Shares learning components to align representations.	Shared observations [46], shared critic [135, 66], shared reward network [254], shared gradients [268].	Not possible in real-world domains.
Group composition.	Constructs group dynamics that promote cooperation.	Role assigning [237], group diversity [148, 147, 149].	Knowledge of social networks or agent types.

Table 2.3: Summary of centralized systems for social dilemmas.

have an incentive to let others make payments). Dong et al., [52] implement a model to encourage *homophily*, where similar-behaving agents are encouraged to have similar payment strategies to mitigate the second-order social dilemma around payment strategies. The multiagent teams we study in our work assume teammates fully share rewards in Chapter 4 and can partially share rewards in Chapter 5. Agents can be strategic with how they share rewards in Section 6.6 of Chapter 6. Our work does not create additional reward but instead shares reward amongst agents on a team. Additionally, since the aforementioned implementations did not outperform the fully cooperative population, and many of our settings generated more reward than the fully cooperative population, we can conclude our settings outperform their methods [266, 52].

Finally, the impact of social network architecture has also been found to have an impact on cooperation and group performance in humans [68, 193], EGT simulations [67, 208, 232], and AI [47]. When giving agents the ability to choose their partner agent, EGT

and AI algorithms have been shown to promote cooperation in the Prisoner's Dilemma game [176, 204, 4]. Similarly, giving agents the ability to remove social ties has also led to cooperation [70]; however, completely controlling which agents they interact with may not be possible in practice. A known result is that cooperation is unable to emerge when RL agents play the IPD with randomly selected other agents [4]. Chapters 4 (Section 4.4) and 5 (Section 5.4.1) shows that teams promote cooperation in social dilemmas despite randomly selected opponents, a setting where agents are unable to modify their social network. Our results show certain settings where agents in teams converge to mutual cooperation by adapting their cooperative behavior with teammates to agents that are in other teams, despite not sharing rewards with those agents and game-theoretic incentives to not cooperate.

Centralized Systems for Social Dilemmas

Research on centralized systems for social dilemma domains that we discuss here are summarized in Table 2.3. Centralized systems of cooperation have taken various forms in previous work, such as centralized structures to support learning, global reward sharing with all agents in a population, or forming groups for tasks with specific agent distributions. These frameworks have advantages of shared formal definitions, globally defined environmental settings, and are easier to promote cooperation and behavioral convergence [83, 50, 3]. However, they typically assume control over all agents, require full observability, or share internal optimization networks. While these assumptions hold under some conditions, they might not be applicable to all problems and may lead to a single point of convergence or lack of robustness.

Some systems rely on agents taking actions that their community agrees on [15] or is pleased by [117]. These systems assume agents report their preferences truthfully, have high rates of communication, or have access to other agents' internal mental states which may not be possible in real-world scenarios. To reduce the complexity of problems and support learning, other work involves agents explicitly sharing or transferring policies [1], local observations [46], a critic network [135, 66], or learning gradients [268]. While this assumption can be overcome through opponent modeling and inverse RL [82], or by creating an intrinsic reward signal through social learning [100], these models often rely on observability of other agents' actions and can be susceptible to deception [35].

Methods focused on reward functions have emerged as the main way to build common interest between RL agents in mixed-motive domains. These include defining agents to have some degree of altruism for all other agents [145, 71], be averse to population inequity [96], or condition reward on the population's performance [55]. While these methods have been

shown to be effective, they rely on having access to the reward functions of all agents in a system and condition agents' rewards relative to the population. Similar to our work, Baker [12] studied the impact of noise and uncertainty over agents' social connections with potentially multiple groups of reward-sharing agents. In groups with less reward sharing, they find that agents learn reciprocal equilibria (i.e., to defect against defecting agents) faster when they have more uncertainty over their social relationships (i.e., the degree of shared rewards from that agent). Furthermore, they found a positive relationship between agents forming cooperative coalitions and social relationship uncertainty. That work mostly studies the impact of social connections instead of team and group structures; thus, our work is different in multiple ways. We do not allow agents to observe any features of how rewards are shared and must learn how to behave through experience (i.e., agents do not view how much reward another agent will give them). Furthermore, we do not inject noise or uncertainty into agents' observations or representations and allow them to reliably observe correct team labels or the surrounding environment (depending on the domain) and find agents in teams learn cooperative behavior while not being exploited by defectors in a variety of conditions and environments.

Despite using individual learning agents, social networks and group construction can be leveraged to study emergence of cooperation. Ultimatum games are those where a proposer agents makes proposals to a set of responder agents with the goal of their proposal being accepted. Pre-assigning roles to agents has been shown to increase fairness in multiplayer ultimatum games using knowledge of agents' social networks [237]. Another type of mixed-motive domain are collective risk dilemmas, where agents donate amounts of capital to a collective pool to avoid some globally unfavorable outcome (i.e., paying costs to mitigate climate change). Without the need for prior labels, constructing groups of agents with diverse wealth and risk perceptions has been shown to promote cooperation in collective goods dilemmas [148, 147]. Furthermore, the compositions of a group has been found to impact the policies developed by individual agents, placing emphasis on how diversity levels in groups should be constructed to promote cooperation [147, 149].

Our work builds on the concept of reward sharing to build common interest; however, we only allow teammates to share rewards in Chapter 4 instead of the broader population. We study multiagent teams with individual RL agents that do not share networks, gradients, or observations. Our results in Chapters 4 and 5 show how teams support the learning process of individual agents in a variety of conditions and environments and promote autonomous role specialization instead of pre-defining roles or diversity distributions.

2.4 Organizational Psychology and Multiteam Systems

The thesis of this dissertation is that teams can have significant advantages in guiding the development of policies for individual agents that learn from experience. We hypothesize that teams help agents better navigate challenging multiagent domains even if their interests are only somewhat aligned with other agents (i.e., only the subset of agents on their team or aligned to various degrees). This idea is broadly inspired by human team organization and team-based behavior.

Organizational psychology (OP) has mainly focused on studying the dynamics within a group of human agents; however, this approach has been argued to not completely capture the mechanisms behind successful organizations since most tasks are completed by the combination of multiple coordinating teams [45]. Such organization can be seen in disaster response, companies, the military, and sports teams. Mathieu et al.[143] introduces the concept of "teams-of-teams" as an organizational structure for human teams, naming the structure multiteam systems (MTSs). MTSs are composed of two or more teams that interface directly and interdependently to accomplish collective goals [247], though they may operate under different contextual demands, authority structures, protocols, and norms [270]. Studies of MTSs typically involve multiple teams of human subjects given simulated tasks to study team coordination [270, 271, 142], structure and boundary status [97], component team differences [136], goal type [49], leadership [244, 121, 209], and shared cognitive, motivational, and cohesive emergent states [34, 104, 139, 44, 87]. One topic of particular interest is studying the abilities and impacts of people to balance personal or team goals with the overall system goal, resulting in a mixed-motive social dilemma [262, 244, 161].

The MTSs community has relied heavily on user studies which often result in domain-specific findings and are sometimes inconsistent which can cause confusion about the impact of team structures on peoples' abilities and development. For example, studies of the benefits of strong within- versus between-team transition and action processes (i.e., taking actions/making decisions across multiple teams or within one team) have found conflicting results [41, 156, 274, 21], at least until between-team coordination limits system autonomy with too much centralization [138]. Another example of inconsistency is the advantage of centralized leadership teams [41, 171] or decentralized planning among team boundary-spanning agents [120, 252, 175]. Consistency among subjects with fewer inherent biases could provide rich insights into the benefits of various aspects of team structure.

In our work, we build on the shared concepts of MAS and MTSs to study the concept

of multiple teams within a larger system of agents with various incentives and goals. We believe incorporating multiteam systems-style research into MAS could help provide more stable findings on the advantages of team structures. The dilemma of balancing related self, team, and system-wide goals creates a social dilemma and a challenging landscape for artificial learning agents to navigate. The thesis of this dissertation is heavily motivated by MTSs. We hypothesize that organizational structures that increase joint human productivity will have similar advantages in MAS. In the next section we detail the early groundwork on teams in AI for task completion domains. We believe modern learning algorithms enable this research to expand further into studying interesting problems comparable to those explored in MTSs.

2.5 Teamwork in Multiagent Systems

While this dissertation studies features of teams with learning agents, the general idea of teamwork and agents working together is not new to MAS. Early work with teams in AI identified models which highlight the importance of joint intentions, sharing plans, and communication for agents to work together. Despite this early work, research using multiple component teams within a population has lacked in recent MARL literature.

Cooperative game theory, coalition structure generation (CSG), and Team Forming can be used to sort a population into sub-groups for some desired goal [265, 192, 215]. In the context of teams, CSG has been used to create sub-teams working towards specific tasks; however, CSG often relies on pure common-interest scenarios, stationary policies or abilities, and requires full control over agents' social networks [116]. In sport domains, coalitions of players with high utility have been found using cooperative game theory methods like the Shapley value [265, 133]. In our work, we are concerned with understanding the development of policies with learning agents divided into teams instead of algorithms that create teams of agents with pre-determined skills.

Agents working to coordinate their behavior has been an important area of research in MAS for several decades. We now present the progression of early work on teams in AI with rule-based agents. These frameworks focus on helping agents coordinate their actions instead of supporting learning; however, they provide valuable groundwork on which we construct our model of teams. Early work by Pollack defines a mental state model for making collaborative plans among two agents [178, 179]. This framework relies on agents having mutual belief that both of their actions would achieve some mutually desired and achievable goal. Extending Pollack's work, Grosz and Sidner [76] construct SharedPlans, a model that includes more multiagent actions and mutual agent beliefs,

similar to shared mental models in human teamwork [58, 63]. Eventually, SharedPlans was extended to incorporate dialogue for communication between agents, however several important limitations remained. First, SharedPlans assumed the impact of joint actions were a linear combination of individual actions, ignoring the idea that joint actions could have stronger impact when agents work together [102, 103]. Second, the model assumed complete prior knowledge of a plan to complete a task [129, 110]. As a result, SharedPlans alone was not complex enough for agents to overcome real problems that required long-term planning and adaptation.

Informed by studies of human collaboration [14], Grosz and Kraus proposed a model to share the internal intentions of agents, not just plans [75]. In their model, agents balanced their intentions for their own success along with the success of the entire group. They proposed a tree structure hierarchy of tasks, where agents perform sub-tasks along branches of the tree towards the final goal positioned at the root. However, this work was primarily domain dependent and only explored using simple rule-based agents. Tambe [233] leveraged the same tree hierarchy of plans to construct a Shell for Teamwork model (STEAM), a general model of teamwork where tasks can be completed by sub-teams of multiple agents within the larger system. STEAM introduced sub-team goals to reduce the complexity of teamwork overheads and uses formal logic and joint intentions to communicate between agents. This idea of sub-teams extended beyond their previous work [234] and other models that allocated role-plans for individuals [110], which made STEAM the state-of-the-art team model with rule-based agents [235]. Extensions to STEAM have added lightweight agent wrappers [182] and increased team lifespan by designing the system to value team persistence over potentially destructive short term plans [235]. However, rule-based agents limited the scope for what these models of teams could achieve since these team hierarchy frameworks were not designed with the intention of supporting learning agents. We construct our team model with multiple levels of goals similar to the team hierarchy used for SharedPlans and STEAM; however, we use more complex RL agents to analyze how teams and team structures influence the policies that learning agents develop over time.

Although STEAM presented a multi-team structure within a broader system, more recent research with teams has mostly overlooked this architecture. Recent work on ad hoc teamwork has maintained the concept of valuable communication [137, 152] and work with learning agents has focused on the balance between individual and group preferences [55, 254]. However, existing algorithms with teams typically assume the existence of a single team with few agents [218, 226, 2, 196, 255]. Whenever work has incorporated multiple teams, the domains are typically competitive [203, 202] or has viewed emergent sub-groups as polarization [206]. One exception is Hu et al.'s work [95], where the local convergence of an action within sub-groups leads to diverse communities, a positive result in their

setting. However, that work only experiments with cooperative matrix games and removes between-community social connections, whereas our work considers social dilemmas (both matrix games and more complex dilemmas) and we do not remove any social connections and allow for any agents to interact.

Fueled by recent successes in deep learning, self play, and natural group selection, Leibo et al. [123] conceptually propose the idea of *adaptive units*. Their proposed definition of an adaptive unit is a group of agents that is composed of sub-units, which itself could be composed of other sub-units, similar to how STEAM has sub-teams composed of agents. They suggest that the interaction between multiple adaptive units forms an *autocurricula*, a self-generated sequence of challenging environments based on the co-evolution of behaviours. Co-evolution of sub-units within the same environment allows for *exploration by exploitation* since the underlying dynamics of the environment shift as interacting agents exploit and optimize their current behavior. Leibo et al. do not expand on their theoretical proposal in that paper [123]; however, Malthusian RL uses these concepts to design an algorithm similar to evolutionary pressures of population size by changing the size of sub-populations based on their performance [124].

While the ideas behind adaptive units and autocurricula are similar to sub-teams, a key distinction from the work done in this dissertation is how adaptive units have been implemented. Malthusian RL [124] changes the size of adaptive units (number of agents) in subsequent episodes of an experiment based on the relative performance of all units in an experiment over previous episodes (executing in parallel simulations). That work showed convergence properties of population sizes across multiple domains, emphasizing how group size has an impact on the policies that are learned. While our work also emphasizes the impact of team size and structure, our work is different in several areas. First, we keep the number of agents in each experiment to be constant. We then modify the number and size of teams (Chapter 4) and how agents optimize their behavior for the goals of themselves, their teams, and the entire system (Chapter 5). We find that teams and various settings of goal alignment promote agents to autonomously discover different distributions of role specializations. Instead, since Malthusian RL uses a shared policy network for agents in the same group, they found that biasing agents towards different roles was necessary to achieve role specialization in some settings. We emphasize that prior role distributions may not be known in different environments; thus, defining prior biases for different roles may result in sub-optimal role distributions. Finally, groups in Malthusian RL rely on the performance of other populations in concurrent simulations instead of groups needing to coordinate within the same environment as done in our work in Chapters 4, 5, and 6.

Our research builds on the prior work presented in this chapter. We adapt existing findings regarding the structure of teams and groups in OP and MTSs to MARL populations

and analyze how features of teams impact learning with individual learning agents. Next, we provide details of our general model of teams and present specific evaluation domains we consider throughout this dissertation.

Chapter 3

Models and Environments

Each chapter of this dissertation utilizes similar notation, frameworks, team models, and environments. This chapter provides more details about the concepts that are used throughout the remaining chapters. We first define our model of multiagent teams. This model is used throughout the dissertation; however, some aspects of the model will change depending on the specific problem addressed in each chapter. We provide specific extensions or modifications to this model in subsequent chapters as appropriate. Lastly, we then present all of the environments used in the empirical evaluations throughout this dissertation.

3.1 A Model for Multiagent Teams

We model our base environment as a stochastic game $\mathcal{G} = \langle \mathcal{N}, S, \{A\}_{i \in N}, \{R\}_{i \in N}, P, \gamma, \Pi \rangle$, a repeated game with probabilistic transitions played by one or more players. $\mathcal{N}$ is our population set of $N \in \mathbb{N}$ agents that learn online from experience and S is the state space, observable by all agents, where s_i is agent i's observation of the environment state. $A = A_1 \times \ldots \times A_N$ 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 $0 \leq \gamma < 1$. Π 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]

Our teams model consists of a stochastic game with teams $\langle \mathcal{G}, \mathcal{T} \rangle$, where $\mathcal{T}$ is a partition of the population of agents into disjoint teams, $\mathcal{T} = \{T_i | T_i \subseteq N, \cup T = N, T_i \cap T_j = \emptyset \forall i, j \in N\}$. We define the term *team structure* as follows:

Definition 5. *A* team structure *is the global composition of $\mathcal{T}$, such as the number of teams and number of agents in each team.*

Consistent with the original groundwork on multiagent teams [233, 76], we define a team of agents as being bounded together through common interest. To be consistent with recent MARL work, we model common interest through reward sharing and assume agents have identical (deterministic) reward functions (i.e., $R_i = R_j$ for all $i, j \in N$) [145, 96]. We define a new reward function for agents in a team as $TR_i : S \times A \times S \mapsto \mathbb{R}$ so that the reward for $i \in T_i$ depends on their own behavior and that of their teammates. Any deterministic function can be implemented to define TR_i so long as every agent in a team gets some amount of the team's reward. In our analysis and experiments, we use:

$$TR_i = \frac{\sum_{j \in T_i} R_j(S, A, S')}{|T_i|}, \tag{3.1}$$

where teammates share their rewards equally to be consistent with past work [254, 13]. While there exist several mechanisms for how teammates share rewards, our model of teams makes the assumption that rewards are shared between teammates instantaneously. Other popular methods include agents explicitly choosing how and when to share rewards [266] or a centralized planner allocating a team's reward according to marginal contribution to team success [195]. In Chapter 5, we relax the assumption that teammates fully share rewards and in Chapter 6 we experiment with agents that learn how to share rewards among groups in the population.

We define agents to be independent learners and learn a policy π_i based on their individual experiences. As is standard in many MARL problems, each agent is trained to independently maximize their own rewards. In particular, at time t each agent i observes the state s_i^t and selects some action a_i^t which together with all agents forms a joint action $\mathbf{a}^t$. This action results in an environment transition from joint state $\mathbf{s}^t$ to joint state $\mathbf{s}^{t+1}$, according to the transition function P, and provides each agent i with reward $R_i^t(\mathbf{s}^t, \mathbf{a}^t, \mathbf{s}^{t+1})$. Agents seek to maximize their sum of discounted future rewards, $V_i = \sum_{t=0}^{\infty} \gamma^t R_i^t$. In later

[1] We can also allow for randomized policies.

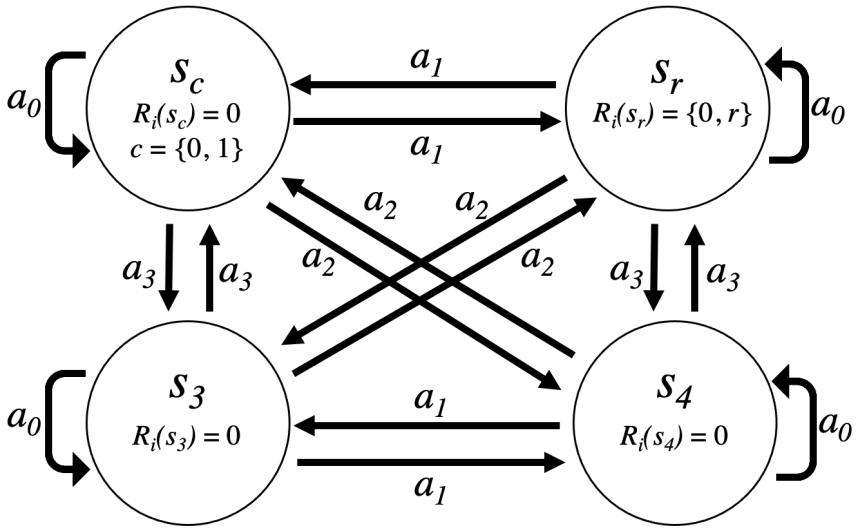

Figure 3.1: **4-States:** Environment diagram.

chapters of the dissertation, we replace R_i with various shared reward functions. For example, an agent that fully optimizes their behavior for a team (Chapter 4) replaces R_i with TR_i, reconfiguring the learning problem so that agents must simultaneously learn what individual behavior maximizes their team's expected discounted future reward. We also experiment with settings where R_i is replaced with a mixture of different reward components for different groups (Chapter 5). We provide specific details about these reward functions in the appropriate chapters.

3.2 Specific Environments Used in Evaluations

Our specific environments range from few-state settings to elaborate and complex gridworld domains with underlying social dilemma or hunter-gatherer dynamics. In this section, we present the environments used in our evaluations throughout this dissertation.

3.2.1 4-States

4-States is a simple, partially observable stochastic game shown in Figure 3.1. This environment can support any number of agents ($N \geq 1$) divided into any number of teams and the state transitions and rewards depend on the joint action of all N agents. There exists

four physical states that agents individually observe: s_c, s_r, s_3, and s_4. At each timestep, agents can take one of four possible actions: stay at their current state (a_0) or move to another state (a_1, a_2, or a_3). The "c" in s_c corresponds with a binary signal (loosely, visiting s_c can be thought of as *reward-causing* depending on this signal) and the "r" in s_r refers to a reward state. States s_3 and s_4 are extra states added to help us assess the performance of agents' joint policies and return a reward of 0. To assist our discussion of the reward dynamics in 4-States, we carefully define the difference between the two types of *states* in this setting:

- s_i^t is the physical state within the environment that agent i is located in and observes at time t.
- $\mathbf{s}^t$ is the joint state of all agents, or the state of the environment, at time t (i.e., the collection of s_i^t for all $i \in N$)

In our analysis, we distinguish between the direct reward an agent receives from the environment when transitioning into their own observed state s_i^{t+1}, $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_i^{t+1})$, and the team reward, TR_i (note the condition on the agent's own observed state s_i^{t+1} in R_i^t). We condition on the individual next state s_i^{t+1} instead of the joint state when defining R_i^t to support a deeper analysis of behavioral dynamics in the 4-State environment (specifically in Chapter 6); however, references to reward for other environments use the joint state (i.e., $R_i^t(\mathbf{s}^t, \mathbf{a}^t, \mathbf{s}^{t+1})$). There is never an environmental reward given to agent i for individually transitioning to s_c, thus $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_c) = 0$. However, any agent (regardless of team affiliation) visiting s_c changes a binary signal c that allows reward to be collected at s_r. Thus, the possible rewards (depending on c) given to any agent transitioning to s_r are $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_r) = \{0, r\}$, where $r > 0$. We assume agents in a team share rewards. When agent i individually transitions to s_r, their reward (before sharing with their team) is $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_r) = 0$ if $c = 0$, and their reward is $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_r) = r$ if $c = 1$. Once the reward is consumed at s_r, c has to be reset by visiting s_c again. Thus, visiting s_c causes reward to be obtained elsewhere in the environment (i.e., when visiting s_r). Explained further in Chapter 6, visiting s_c can be thought of as a *reward-causing state-action* pair. The two additional states, s_3 and s_4, do not impact the reward dynamics of the environment but are included to understand how well agents learn the underlying environment (i.e., they should learn to avoid visiting these states).

This environment's reward function emphasizes the importance of coordination and social responsibility. Visiting s_c requires an agent deciding to fulfill a role of social responsibility and visit a state that will not return an explicit reward; however, an optimal

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

Table 3.1: An example of the Prisoner's Dilemma with the costs (c) and benefits (b) of cooperating ($b > c > 0$). Mutual defection is the unique Nash Equilibrium when playing the Prisoner's Dilemma with agents that do not share rewards.

	Cooperate	Defect
Cooperate	$b-c$, $b-c$	$\frac{b-c}{2}$, $\frac{b-c}{2}$
Defect	$\frac{b-c}{2}$, $\frac{b-c}{2}$	0, 0

Table 3.2: An example of the Prisoner's Dilemma when agents are teammates. Mutual cooperation is the unique Nash Equilibrium when playing the Prisoner's Dilemma with a teammate and sharing rewards.

solution is when exactly one agent visits s_c at each timestep (and other agents are in s_r to collect the reward). Thus, coordination is crucial to achieve a good joint policy in this environment.

Chapter 6 shows how the size of a team has a significant impact on how agents coordinate in this environment. Specifically, if a team is large, agents form joint policies that achieve sub-optimal results by either having too many agents in s_c or visiting s_3 and s_4 too often (which provides nothing to the underlying reward structure).

3.2.2 Iterated Prisoner's Dilemma (IPD)

The Prisoner's Dilemma is a decades-old matrix game analyzed in game theory that represents a social dilemma [194]. In the one-shot Prisoner's Dilemma, two agents interact and each must decide to either cooperate with (C) or defect on (D) each other. We assume there is a cost (c) and a benefit (b) to cooperating where $b > c > 0$ (the payoff matrix is shown in Table 3.1 for one *row* agent and one *column* agent). If an agent cooperates, it incurs the cost c. If both agents cooperate, they both also benefit, each receiving a reward of $b - c$. If one agent cooperates but the other defects, then we assume that the cooperating agent incurs the cost c, but the defecting agent reaps the benefit b (e.g., by

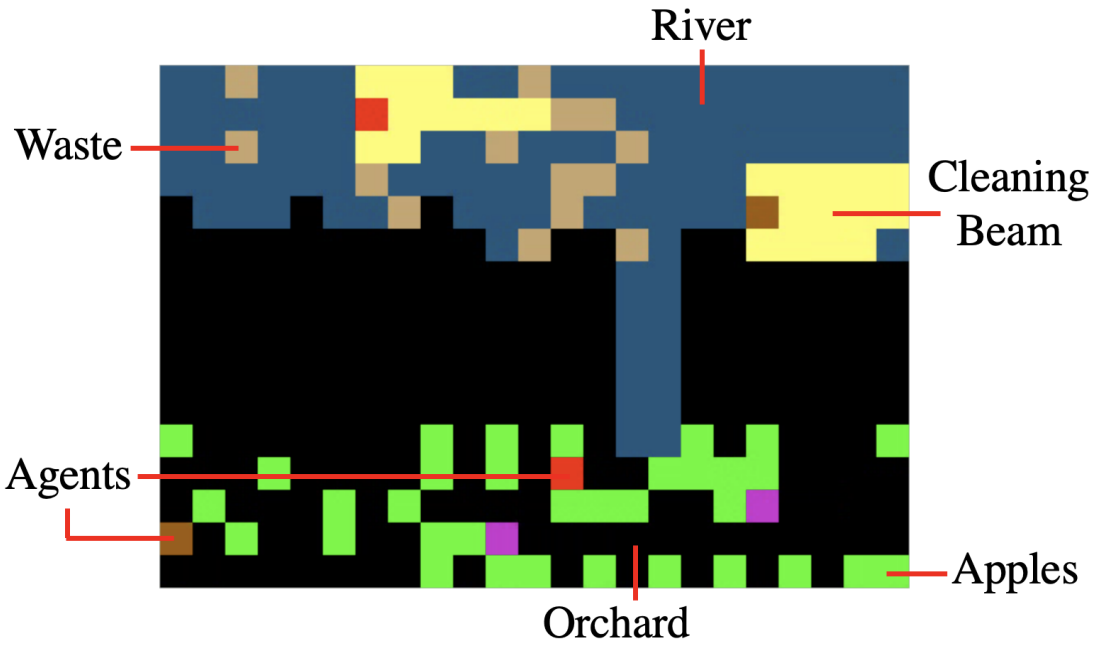

Figure 3.2: **Cleanup:** Cleanup environment with six agents in three teams of two agents each. Agents are represented as squares (i.e., two red, two purple, and two dark brown.

stealing the contribution of the cooperator). If neither cooperate, neither benefit nor incur a cost, leading to a reward of zero for both. The unique Nash Equilibrium is obtained when both agents defect, represented by (D, D). Mutual cooperation does not form an equilibrium, since if one agent cooperates, the other agent is strictly better off defecting and receiving b, instead of $b - c$.

In the Iterated Prisoner's Dilemma (IPD), this game is repeatedly played which adds a temporal component and allows agents to learn a policy over time. Instead of just two agents, we work with a population of agents that are divided into teams *a priori*. At each timestep, agents are randomly paired with another agent, a *counterpart*, that may or may not be a teammate. Agents observe the team identity of their counterpart at time t as s_i^t, though additional identity information is not shared. Agents must decide to cooperate with or defect on their counterpart as a_i^t. Their payoff for the interaction is their team's reward, based on their own and other teammates' interactions. Thus, their direct payoffs for their own interaction (that they contribute to the team reward) come from Table 3.1 if their counterpart **is not** a teammate and their payoffs come from Table 3.2 if their counterpart **is** a teammate (when teammates equally share rewards by Equation 3.1). Agents update

their strategies (i.e., learn) using their direct observation, what action they chose, and their team's reward. Since the only information shared is the team the counterpart belongs to, the strategies of all agents on a team ultimately affects how agents learn to play any member of that team.

We use the IPD for empirical evaluations in Chapters 4, 5, and 6. Our results show multiple settings where agents in teams that have the game-theoretic incentive to defect learn mutual cooperation with agents in other teams (despite not sharing any rewards with these agents). These results are robust to multiple scenarios where agents may not fully optimize for their team's reward and may instead be somewhat selfish.

3.2.3 Cleanup Gridworld Game

The Cleanup Gridworld Game (or simply Cleanup) [249] is a temporally and spatially extended Markov game that represents a social dilemma. This domain allows us to examine a more complex environment than the IPD while maintaining aspects of an underlying social dilemma. Specifically, agents must learn a cooperative policy through movement and decision actions instead of choosing an explicit *cooperation* action like in the IPD. Successful groups in Cleanup require some agents to perform *active provision* [96], taking actions that supply something of use [224]. Active provision in Cleanup is when agents choose actions with no associated environmental reward, but these actions are necessary for agents to achieve rewards (i.e., these actions are *reward-causing*). Cleanup is a widely used simulated environment in previous MARL research studying the emergence of coordination, cooperation, and pro-social policies among a population of learning agents [96, 100, 266, 52].

Figure 3.2 shows a timestep of an experiment in the Cleanup environment with six agents. The environment visually represents features (and agents) as squares in Figure 3.2. The gridworld contains a river on one side and an apple orchard on the other side. Agents are represented by colors, and we define teammates to share the same color. Figure 3.2 shows a scenario with three teams of two agents each; thus, two agents are red squares, two are purple squares, and two are dark brown squares. Apples only grow in the apple orchard and appear as green squares, whereas light brown waste only spawns in the river. Agents clean the river using a "cleaning beam" action that is represented by a collection of yellow squares.

At each timestep, agents choose among nine actions: five movement (up, down, left, right, or stay), two turning (left or right), and a cleaning or punishing beam. An agent's observability is limited to an egocentric window of 15 $\times$ 15 pixels. Waste accumulates in

the river with some probability at each timestep which must be cleaned by the agents. Once a cleanliness threshold is reached, apples spawn in the orchard proportional to the overall cleanliness of the river. Agents receive a reward of +1 for consuming apples by moving on top of them. The dilemma exists in agents needing to take non-rewarding actions to clean the river to spawn new apples versus staying in the orchard and enjoying the fruits of another's labor. Agents have the incentive to stay in the orchard; however, if all agents attempt this free-riding policy, no apples grow and none get any reward. A successful group in Cleanup will balance the temptation to free-ride with the public obligation to clean the river.

In Chapters 4, 5, and 6 when teammates in Cleanup share some amount of their rewards, rewards are distributed to teammates at the timestep they are collected instead of teammates needing to distribute rewards themselves. Future work could incorporate methods where agents must learn to share with their teammates in subsequent timesteps; however, our model of teams implements instantaneous reward sharing. This is analogous to a team achieving some team-level goal.

In Chapter 4, we find that teams promote autonomous role specialization and the emergence of efficient global joint policies. These joint policies generate 33% more mean population reward than the fully cooperative population (i.e., when all agents in the population share rewards). In Chapter 5, our results show that this joint policy can also emerge in settings when agents are slightly selfish and in Chapter 6, we show how a team's ability to generate high rewards is significantly influenced by the size of the team.

3.2.4 Neural MMO (NMMO)

Neural MMO (NMMO) [228] is a large, customizable, and partially observable multiagent environment that supports foraging and exploration. NMMO is different than Cleanup since it simulates hunter-gatherer societies instead of supporting an underlying social dilemma and has gained popularity among the MARL community to understand how agents operate in large environments [168, 79]. Figure 3.3 shows the NMMO environment with no agents. There is no standard NMMO implementation; thus, we configure a map with 1024×1024 pixels bounded by lava tiles to enclose the agents within the environment. Within the lava boundary, NMMO has squares of grass for agents to move freely on, stones as obstacles, and water and forest (food) squares that regenerate over time. The map is randomly initialized; however, we spatially separate water and forest to encourage exploration. Otherwise, there would be water and forest tiles randomly interspersed and agents would not be forced to explore different areas of the map. An agent's observability

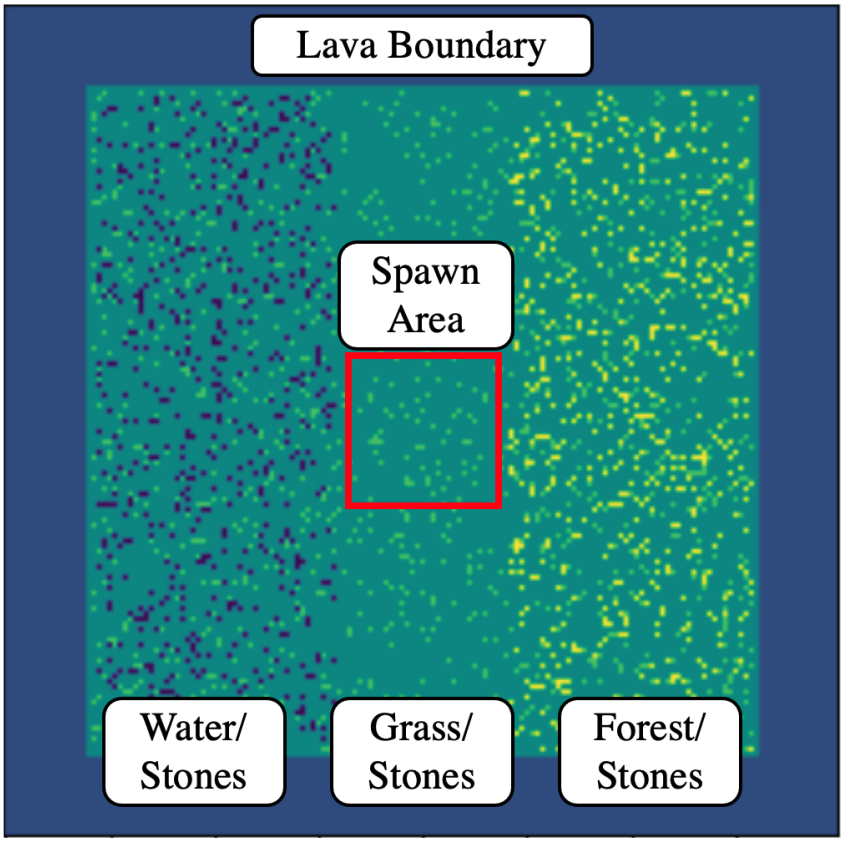

Figure 3.3: **NMMO:** Environment layout.

is limited to an egocentric window of 15 × 15 pixels. We configure the environment so that agents always spawn in the center of the map at the beginning of each episode (i.e., not in the water or forest area) and agents can take movement and combat actions.

Agents maintain a stash of consumable resources (food and water). Collecting any forest (food) or water tile increases an agent's personal inventory for that resource by +0.1 up to a maximum amount of 1.0. At each timestep, an agent's inventory for both food and water deplete by a rate of -0.02 down to a minimum of 0.0 each. When in teams, teammates share water and food resources amongst themselves.

There is no standard reward function in NMMO. To simulate the dynamics of a hunter-gatherer society with multiple tasks, we reward agents based on their inventory of resources. We reward agents for positive increases to their lowest resource: $\min(I)^t - \min(I)^{t-1}$ when $\min(I)^t > \min(I)^{t-1}$, where I is the inventory of food and water. Thus, agents must learn

to maintain both food and water to receive reward. We remove agent "death" by starvation since this feature is not relevant to our study and it's removal maintains consistent episode lengths and learning steps in each experiment. We use NMMO in Chapter 6 where teammates share their rewards according to Equation 3.1. Similar to Cleanup, rewards are distributed to teammates at the timestep they are collected instead of teammates needing to distribute rewards themselves.

In Chapter 6, we find that a single agent in NMMO is unable to learn the reward dynamics of maintaining both food and water resources. When agents are in a team, they divide labor to collect either food or water, similar to the dynamics of hunter-gatherer societies. However, we find that the ability for teams to form an effective joint policy depends on the size of the team.

Chapter 4

The Benefits of Teams in Multiagent Learning

This chapter first introduces the idea of multiagent teams in social dilemma domains with individual learning agents. We find that forming teams within a population helps agents develop cooperative pro-social policies despite incentives to not cooperate. Furthermore, agents are able to better coordinate and learn emergent roles within their teams to achieve higher rewards compared to when the interests of all agents are aligned. This is done through agents autonomously discovering a more efficient global joint policy when defined in certain team structures compared to the fully cooperative population.

4.1 Introduction

Observed in both animal and human behavior, the ability to work in teams can magnify a group's abilities beyond the capability of any individual. In multiagent research with AI agents, a large area of research focuses on the process of *forming* teams of agents with defined abilities towards a known goal [265, 192, 215]. With teams of learning agents, multiagent reinforcement learning (MARL) has achieved impressive results in competitive two-team zero-sum settings such as capture the flag [99], hide-and-seek [13], and Robot Soccer (RoboCup) [111].

While Team Forming algorithms have important and widespread applications, they do not analyze how team dynamics and structures impact the development of learning agents' policies such as in MARL contexts. In the two-team zero-sum domains with MARL teams,

the structure of teams are typically defined (i.e., hide-and-seek is always 2 vs. 2), meaning the potential impact of team structures are typically overlooked. Furthermore, when agents are deployed into the real world, they will be faced with problems that are not zero-sum [12]. Therefore, there is growing interest in exploring how agents can learn cooperation in mixed-motive domains, such as Sequential Social Dilemmas (SSDs) [125]. This chapter presents an analysis of the impact and benefit of teams and different team structures on the learning process for individual agents that learn in the context of mixed-motive domains.

Inspired by group structures in organizational psychology (OP) and early models of teams from the AI literature for task completion (highlighted in Chapter 2), we implement the general model of multiagent teams presented in Chapter 3 and evaluate it in the context of social dilemmas. It is well documented that individual RL agents fail to learn cooperation in social dilemmas while agents with common interest have more success [4, 13]. Our teams model is situated between these two extremes, where teammates are bound by common interest but mixed-motives exist between non-teammates. We show in the Iterated Prisoner's Dilemma (IPD) and Cleanup Gridworld Game (both presented in Chapter 3) that teams improve how agents learn and develop pro-social policies. This chapter makes the following contributions:

- We implement a model of teams inspired by early work in multiagent systems and OP (presented in Chapter 3).
- In Section 4.2 we discuss the theoretical ramifications of our model in the context of social dilemmas regarding game-theoretic incentives under different environmental conditions.
- Through an extensive empirical evaluation, Section 4.3 shows how our model of teams helps agents develop globally beneficial pro-social behavior despite short-term incentives to not cooperate. As a result, agents in teams achieve higher rewards in complex domains than the fully cooperative system by autonomously learning more efficient combinations of roles.

4.2 Equilibrium Analysis with Teams

Our environment in this context is the N-agent stochastic game presented in Chapter 3. In this section, we perform an equilibrium analysis in the context of the IPD environment to understand how teams impact the game-theoretic incentives for behavior in this repeated

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

Table 4.1: An example of the Prisoner's Dilemma with the costs (c) and benefits (b) of cooperating ($b > c > 0$).

game. Recall that in our implementation of the IPD, a population of agents is divided into teams *a priori*. At each timestep t:

1. Agents are randomly paired with another agent, a *counterpart*, that may or may not be a teammate.
2. Agents are informed as to what team their counterpart belongs to through a numerical signal s_i, though additional identity information is not shared.
3. Agents must decide to cooperate with (C) or defect on (D) their counterpart, a_i^t.
4. Agents receive their team reward, TR_i^t, based on their own and their teammates' interactions.
5. Agents update their strategies (i.e., learn) using their own direct observation s_i^t, what action they chose a_i^t, and their team reward TR_i^t.

Since only the team information of the counterpart is shared, the strategies of all agents in team T_i ultimately affects how agents learn to play any member of T_i. We are interested in understanding how the introduction of teams may help or hinder cooperation. As a first step towards addressing this question, we investigate the impact of teams on the *stage game* of the IPD. To provide a clear comparison with the standard IPD, we take an ex-ante approach, where agents are aware of their imminent interaction and the existence of other teams but not the actual team membership of their counterpart.

Assume a pair of agents, i and 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 $(1-\nu)$. Also assume agent j is playing some strategy summarized by the probability that agent j selects action C conditioned on if they are a teammate or non-teammate. Let $\sigma_{T_i} = (\sigma_{ji}, 1-\sigma_{ji})$ be the strategy profile for agent j, where σ_{ji} is the probability that j selects action C if $j \in T_i$ (a teammate) and $\sigma_{T_j} = (\sigma_{jj}, 1-\sigma_{jj})$ be

	Cooperate	Defect
Cooperate	$b-c,\ b-c$	$\frac{b-c}{2},\ \frac{b-c}{2}$
Defect	$\frac{b-c}{2},\ \frac{b-c}{2}$	$0,\ 0$

Table 4.2: An example of the Prisonner's Dilemma when agents are teammates with full common interest. (C, C) is the unique Nash Equilibrium.

the strategy profile when $j \in T_j$ (is not a teammate). The expected utility of i choosing to cooperate (C) or defect (D) can be derived using Table 4.1, Table 4.2 (same tables as in Chapter 3, repeated here for the reader), ν, and the strategy profile of j, σ_{T_i} or σ_{T_j} (we denote strategy profile as σ_T below when referencing both σ_{T_i} or σ_{T_j}, such as in the expected utility to cooperate $\mathbb{E}(C, \sigma_T)$).

If agent i decides to cooperate, it's expected utility, subject to agent j's strategy, is calculated by:

$$\mathbb{E}(C, \sigma_T) = \nu \left[\sigma_{ji}(b-c) + (1-\sigma_{ji})\frac{b-c}{2} \right] + (1-\nu)\left[\sigma_{jj}(b-c) + (1-\sigma_{jj}) - c\right] \tag{4.1}$$

$$\mathbb{E}(C, \sigma_T) = \nu \left[\frac{2\sigma_{ji}(b-c)}{2} + \frac{b-c}{2} - \frac{\sigma_{ji}(b-c)}{2} \right] + (1-\nu)\left[\sigma_{jj}b - \sigma_{jj}c - c + \sigma_{jj}c\right] \tag{4.2}$$

$$\mathbb{E}(C, \sigma_T) = \nu \left[\frac{\sigma_{ji}b - \sigma_{ji}c}{2} + \frac{b-c}{2} \right] + (1-\nu)\left[\sigma_{jj}b - c\right] \tag{4.3}$$

$$\mathbb{E}(C, \sigma_T) = \nu \left[\frac{(b-c)(\sigma_{ji}+1)}{2} \right] + (1-\nu)\left[\sigma_{jj}b - c\right] \tag{4.4}$$

$$\mathbb{E}(C, \sigma_T) = \frac{\nu(b-c)(\sigma_{ji}+1)}{2} + (1-\nu)(\sigma_{jj}b - c) \tag{4.5}$$

If agent i decides to defect, it's expected utility, subject to agent j's strategy, is:

$$\mathbb{E}(D, \sigma_T) = \nu \left[\sigma_{ji}\frac{(b-c)}{2} \right] + (1-\nu)\left[\sigma_{jj}b\right] \tag{4.6}$$

$$\mathbb{E}(D, \sigma_T) = \frac{\nu\sigma_{ji}(b-c)}{2} + (1-\nu)\sigma_{jj}b \tag{4.7}$$

We determine the conditions under which agent i has incentive to cooperate as when $\mathbb{E}(C, \sigma_T) \geq \mathbb{E}(D, \sigma_T)$. We calculate this scenario by substituting $\mathbb{E}(C, \sigma_T)$ and $\mathbb{E}(D, \sigma_T)$ from above:

$$\frac{\nu(b-c)(\sigma_{ji}+1)}{2} + (1-\nu)(\sigma_{jj}b - c) \geq \frac{\nu\sigma_{ji}(b-c)}{2} + (1-\nu)\sigma_{jj}b \tag{4.8}$$

$$\frac{\nu(b-c)(\sigma_{ji}+1)}{2} - c + \nu c \geq \frac{\nu\sigma_{ji}(b-c)}{2} \tag{4.9}$$

$$\frac{\nu(b-c)}{2} - c + \nu c \geq 0 \tag{4.10}$$

$$\nu(b-c) + 2\nu c \geq 2c \tag{4.11}$$

$$\nu b + \nu c \geq 2c \tag{4.12}$$

$$\nu \geq \frac{2c}{b+c} \tag{4.13}$$

The above derivation calculates the point at which agents have incentives to cooperate in our environment. In the regular IPD without teams, agents have no common interest making (D, D) the unique Nash Equilibrium and (C, C), (C, D), and (D, C) the three Pareto Optimal strategies. Since teammates share rewards, the degree of common interest is ultimately determined by the amount they interact with their team, ν (i.e., more teammate-teammate interactions means a higher degree of population common interest). Therefore if Equation 4.13 is satisfied, the game-theoretic properties of the IPD transform so that (C, C) is the unique Nash Equilibrium and Pareto Optimal strategy. We further analyze these incentive dynamics in each of our evaluation domains in the next section; however, we find that teams often lead to agents learning cooperation in settings where they have incentives to defect.

4.3 Empirical Evaluation Configuration

In this section, we present the setup and results of experiments in the IPD [194] (Section 4.4) and Cleanup [249] (Section 4.5) environments using MARL agents. While our teams model does not require it, we assume that for all $T_i, T_j \in \mathcal{T}$, $|T_i| = |T_j|$ (i.e., given a team model, the teams are the same size). This avoids complications that might arise with agent interactions if teams were significantly different sizes and to be consistent across our domains. Alternative interaction mechanisms and teams of different sizes are left for future work. We use the notation $|\mathcal{T}|/|T_i|$ to indicate the total number of teams and the size of each team. For example, $1/N$ indicates one team of N agents (fully cooperative) and $N/1$ represents N teams of one agent (fully mixed-motive). Of course, many scenarios may fall between these two extremes. Since fully mixed-motive has agents working as individuals (i.e., no teams, or N teams of one), it serves as a benchmark against which we can compare the performance of team structures.

4.4 IPD Evaluation

In the IPD, each experiment lasts 1.0×10^6 episodes where $N = 30$ agents learn using Deep Q-Networks [153]. An episode is defined by a set of agent interactions where each agent is paired with another agent and plays an instance of the Prisoner's Dilemma. Agent pairings are assigned using a uniform random distribution over each team so agents are unable to explicitly modify who they interact with, known as a challenging scenario for cooperation to arise without additional infrastructure [4]. We define a counterpart as having equal probability of being in any team (i.e., $p(s_i^t = T_i) = p(s_i^t = T_j) \forall T_i, T_j \in \mathcal{T}$). Each experiment is repeated five times to study variance in results. In Appendix A.1, we prove how this configuration ensures that each agent has the same number of expected interactions to learn from.

Population Reward Results

In our first set of experiments, we explore the degree to which team structures support cooperation. We fix the cost (c) at 1, and let the benefit (b) be 2, 5, or 10. To capture the behavior of agents after they have converged to a policy, the top graph of Figure 4.1 shows the normalized average global reward of the last 25% of the episodes using individual learning RL agents. We normalize the average global reward of each experiment in the interval $[0 - c, 0 + b]$ and calculate 95% confidence intervals to compare different cost and

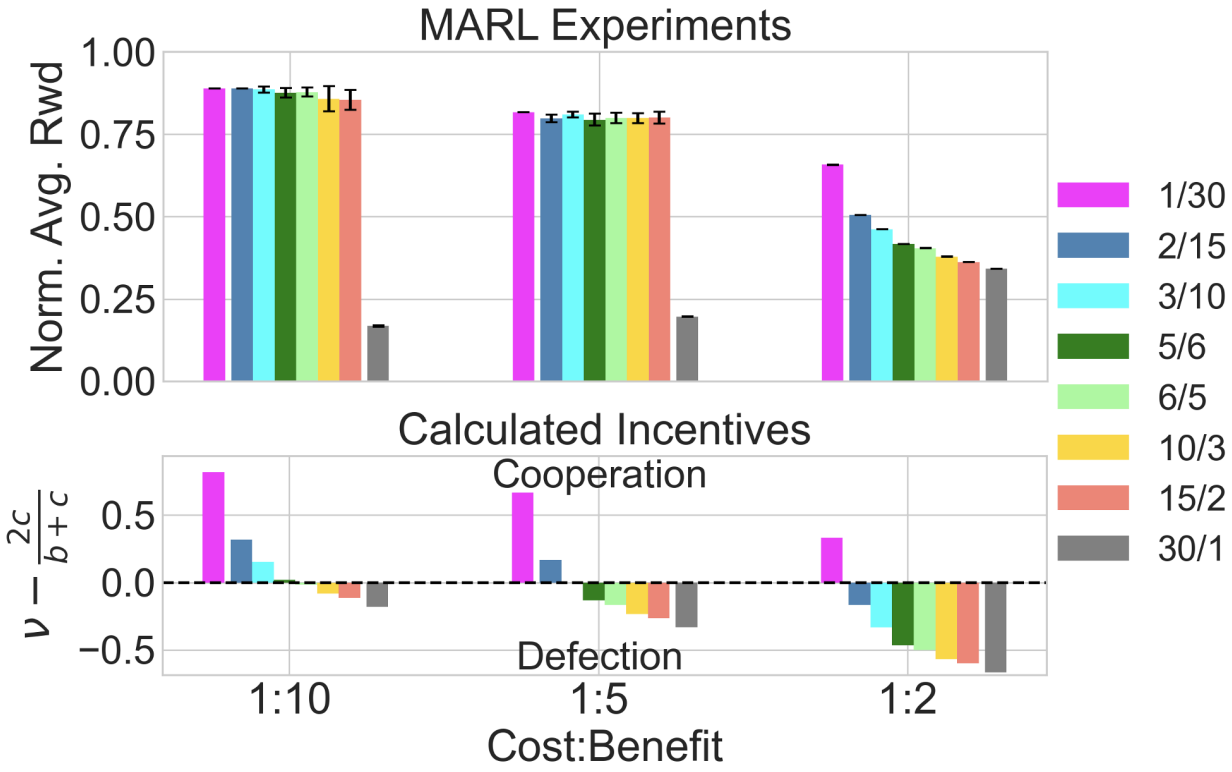

Figure 4.1: **IPD:** The top graph shows the normalized average population reward of MARL experiments with three different cost:benefit ratios when $N = 30$ with 95% confidence intervals. The bottom graph shows incentivized actions from Equation 4.13, where positive (or zero) is cooperation and negative is defection being incentivized. Team structures are labeled $|\mathcal{T}|/|T_i|$ and bookended with fully cooperative (1/30) and fully mixed-motive (30/1). When $b \in \{5, 10\}$, every team structure besides the individualistic case (30/1) achieves about as much reward as 1/30 without requiring a fully cooperative the population.

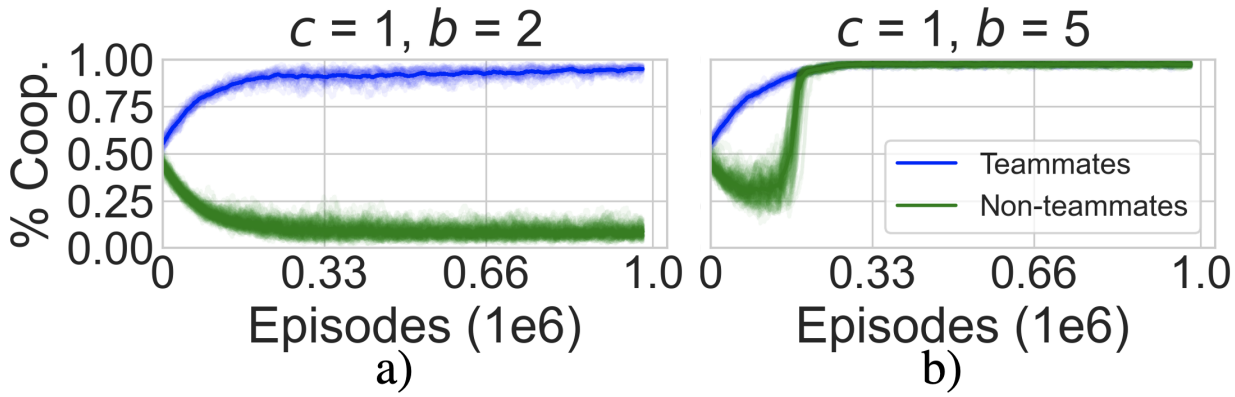

Figure 4.2: **IPD:** The 5/6 team composition showing the percent of cooperation towards teammates and non-teammates when $c = 1$ and $b \in \{2, 5\}$. When benefit is greater, agents develop pro-social policies towards non-teammates despite the incentive to defect.

benefit ratios on the same plot. To show the corresponding incentives of each experiment, we include the bottom graph which displays the calculated action incentive by a modified Equation 4.13, $\nu - \frac{2c}{b+c}$. Each bar in this graph corresponds with the experiment in the top plot so that positive (or zero) bars represent cooperation being incentivized and negative bars represent defection. Cost and benefit ratios are arranged from highest benefit (left) to lowest benefit (right).

Our results show teams always achieve more reward than individual agents (30/1); however, this reward depends on the cost and benefit ratio. When $b = 2$, the experiment results for average population reward a follow trend similar to the incentives of each scenario in the bottom graph. Our main finding in Figure 4.1 is how, when the benefit increases, individual RL agents achieve high average population reward despite the incentive to defect as shown in the bottom graph. When $b \in \{5, 10\}$, every team structure, other than the individualistic 30/1 scenario, achieves basically the same reward as 1/30 even though there exists mixed-motive interactions with other teams. Defection is the incentivized action in seven of 12 (58%) of these experiments that would produce low global reward if agents actually learned defection. Instead, we observe agents develop reciprocally pro-social policies that achieve high rewards in every scenario with teams of multiple agents when $b \in \{5, 10\}$. To analyze how high rewards are achieved in environmental conditions that promote defection, we study agents' behavior over time.

Analyzing Learned Policies

In evolutionary biology, fostering cooperation at various *levels* has been found to depend on the size of the cooperative return [210]. The idea behind cooperation levels comes from the concept that not all cooperative actions are equal, cooperating with certain groups is more significant than cooperating with other groups. Different types of cooperation, or levels of cooperation, have yet to be explicitly explored in MARL; however, teams allow us to identify two levels of cooperation in our IPD environment: cooperation with teammates and cooperation with non-teammates. Figure 4.2 shows the percent of cooperative actions over time with the 5/6 team structure when $b \in \{2, 5\}$. By Equation 5.2, agents have the incentive to defect in both scenarios. The x-axis shows time and the y-axis shows the percent of an agent's actions that are cooperation (2,000 episode sliding window mean, single episode progression).

Both graphs in Figure 4.2 show that agents immediately learn to cooperate with teammates regardless of b. When $b = 2$, agents defect on non-teammates; however, when $b = 5$, agents learn to cooperate with both teammates and non-teammates. We observe similar behavior with every other team structure (not including 30/1) when $b \in \{5, 10\}$. That is, cooperation emerges with teammates and non-teammates despite incentives to defect. While other work requires strong assumptions of agent behavior to foster cooperation, our results indicate teams allow agents to learn an emergent cooperative convention at multiple levels of a system in certain settings.

4.5 Cleanup Gridworld Game Evaluation

We expand our evaluation of teams to the Cleanup Gridworld Game [249]. Instead of distinct "cooperate" and "defect" actions like in the IPD, agents in Cleanup must learn entirely cooperative or defecting policies through their general behavior in the environment (i.e., cleaning the river or picking apples). This added complexity allows us to further analyze how agents develop joint policies, converge to various roles in the environment, and learn to explore the underlying dynamics of the environment.

To allow for multiple teams with the same size in Cleanup, we experiment with $N = 6$ agents (previous work typically uses $N = 5$ [96, 145, 100]). Our agents use the Proximal Policy Optimization (PPO) [214] RL algorithm for 1.6×10^8 environmental timesteps (each episode is 1,000 timesteps). Agent observability is limited to a 15×15 egocentric RGB window. Teammates share the same color and optimize for TR_i calculated at each environmental timestep. Each experiment is repeated for eight trials.

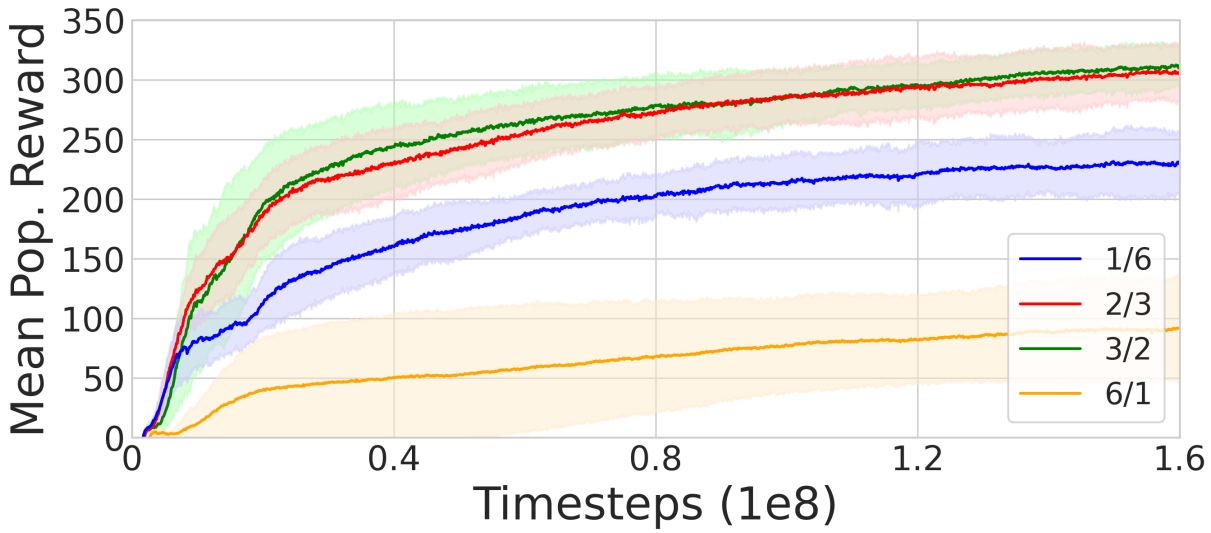

Figure 4.3: **Cleanup:** Mean population reward for each team structure with 95% confidence intervals. 6/1 represents individualistic agents and 1/6 represents a fully cooperative population. Both 2/3 and 3/2 team structures achieve more reward than 1/6 and 6/1.

Mean Population Reward in Cleanup

Figure 4.3 shows the mean population reward for each scenario in Cleanup with 95% confidence intervals. It has been previously assumed that the setting that achieves the most population reward in Cleanup is when agents are fully cooperative and optimize for the collective rewards of the entire group [266, 52, 254, 145], similar to our 1/6 configuration. However, teams introduce a new dynamic to the environment and we find the 2/3 and 3/2 team structures both achieve 33% more reward than 1/6 despite the interests of all agents not being aligned. As expected, the 6/1 scenario fails to achieve significant reward since agents succumb to the incentive to free ride and few apples grow. McKee et al., [145] has shown that only evaluating a system for mean reward masks other dynamics such as high levels of reward inequality among agents.

Reward Equality Among the Population

Mean population reward does not fully investigate the dynamics of why or how team structures achieve this higher reward. It is important to consider potential side effects on population reward equality, such as how the reward is distributed among agents in these

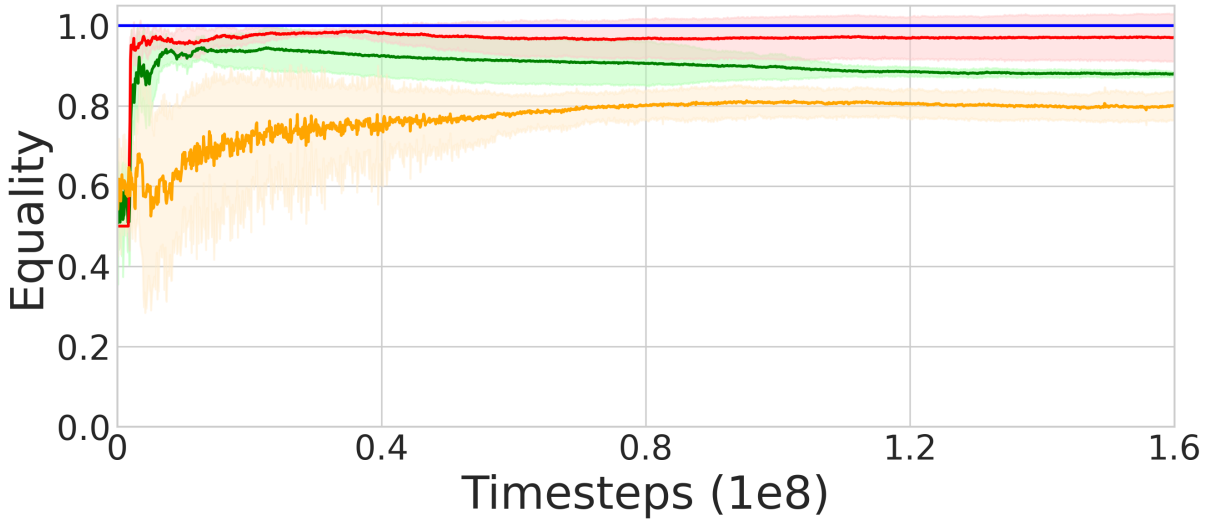

Figure 4.4: **Cleanup:** Inverse Gini index (equality) for each team structure with 95% confidence intervals. Higher values represent more equality. Both 2/3 and 3/2 team structures have high equality despite the interests of all agents not being aligned.

settings. It is important to understand if teams introduce scenarios that lead to high inequality for settings where reward inequality may be detrimental to a system. We model population reward equality as the inverse Gini index, similar to past work [145], calculated as:

$$Equality = 1 - \frac{\sum_{i=0}^{N} \sum_{j=0}^{N} |R_i - R_j|}{2N^2 \overline{R_N}}, \tag{4.14}$$

where $\overline{R_N}$ is the mean population reward. Figure 4.4 shows our results for reward equality over time with 95% confidence intervals where higher values represent more equality. The 1/6 scenario is, by definition, always 1 since there is only one team. Despite earning high reward, both 2/3 and 3/2 team structures also achieve high equality and always have greater equality than 6/1. Success in Cleanup relies on agents coordinating to form an effective joint policy instead of simply choosing an explicit cooperation action (as in the IPD). To further understand how team structures achieve the highest rewards while also maintaining high equality, we analyze agents' policies and division of labor that generates the increase of reward.

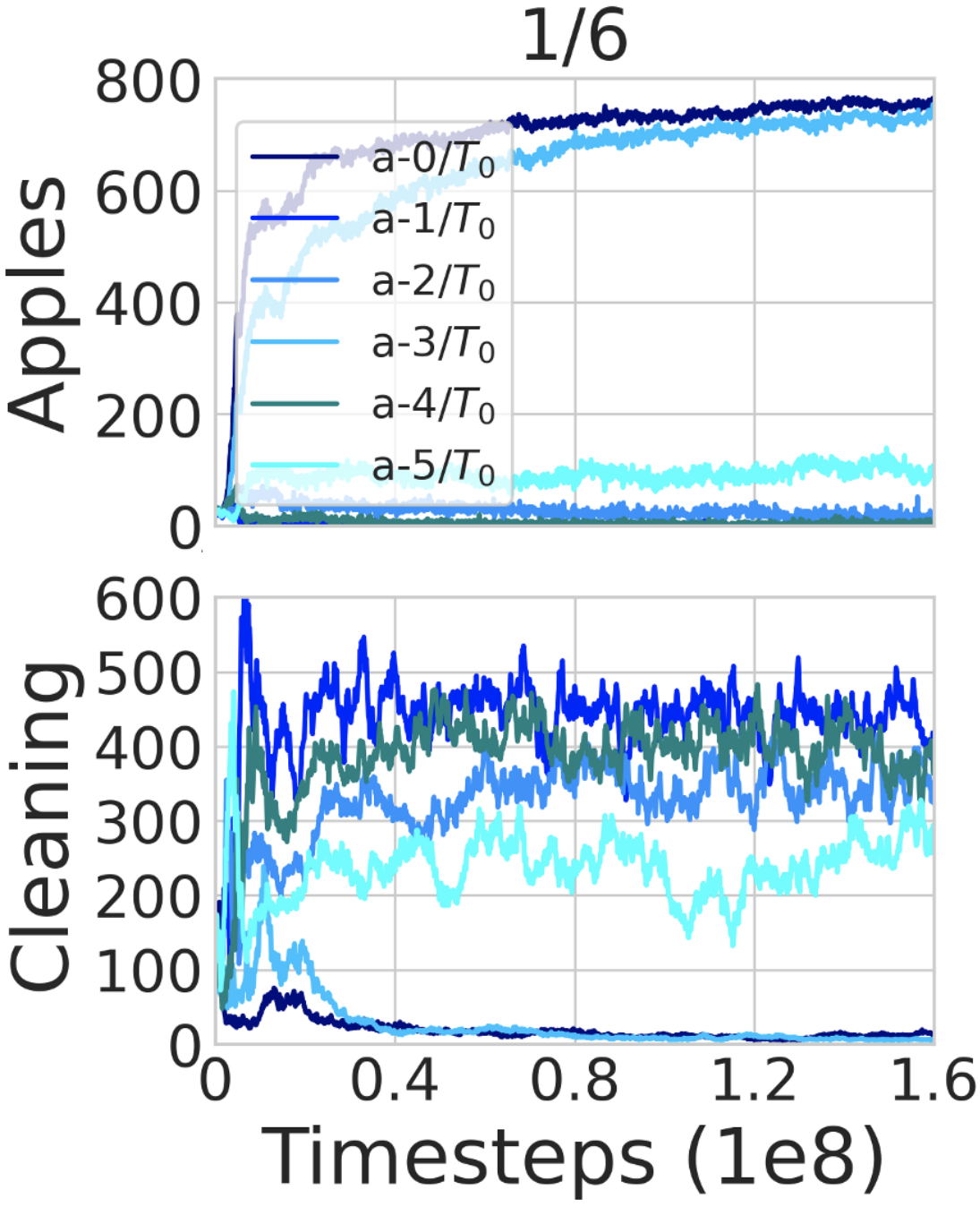

Figure 4.5: **Cleanup:** One team of six agents. Mean number of apples picked (top) and cleaning beams selected (bottom) per-episode.

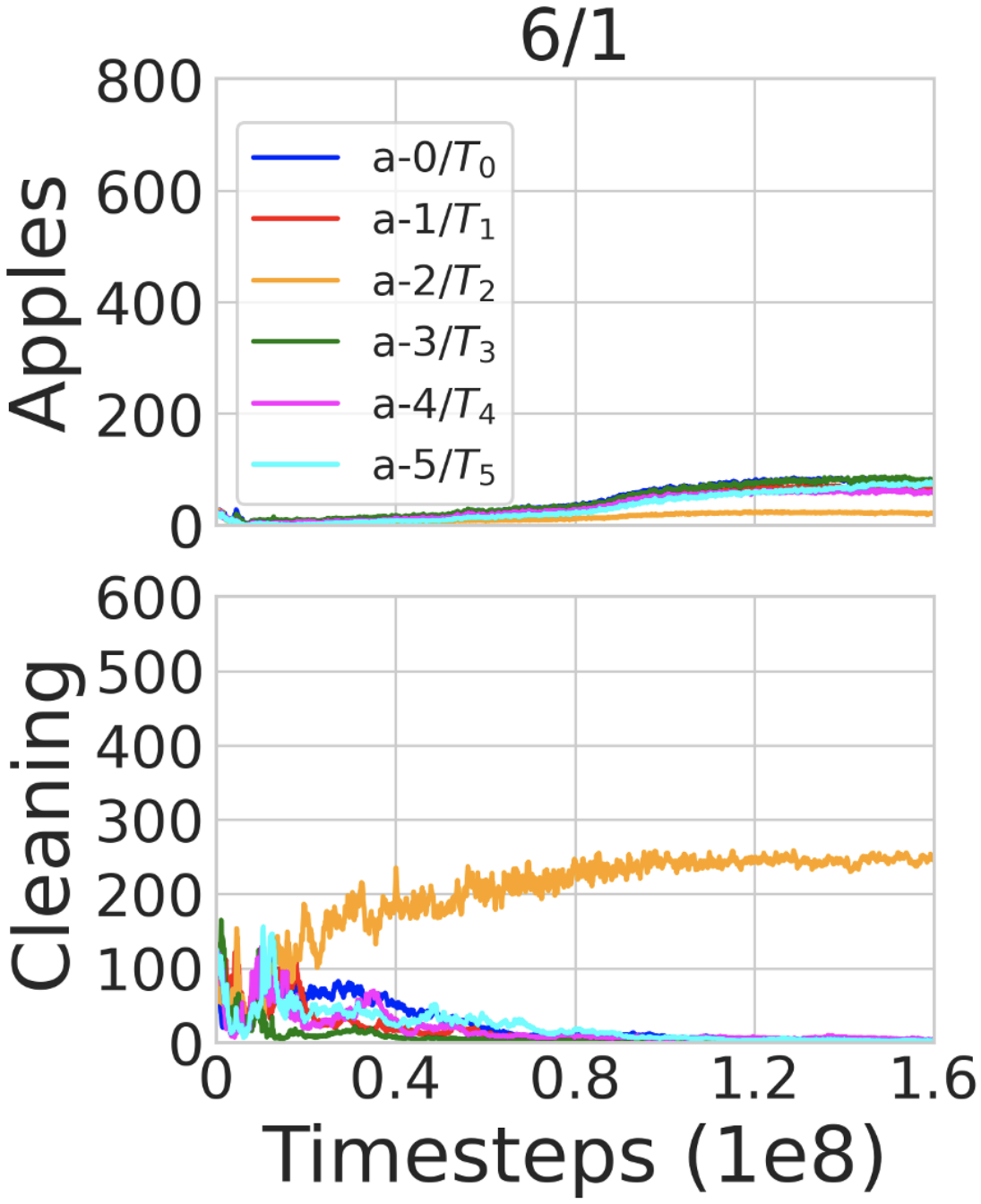

Figure 4.6: **Cleanup:** Six teams of one agent each. Mean number of apples picked (top) and cleaning beams selected (bottom) per-episode.

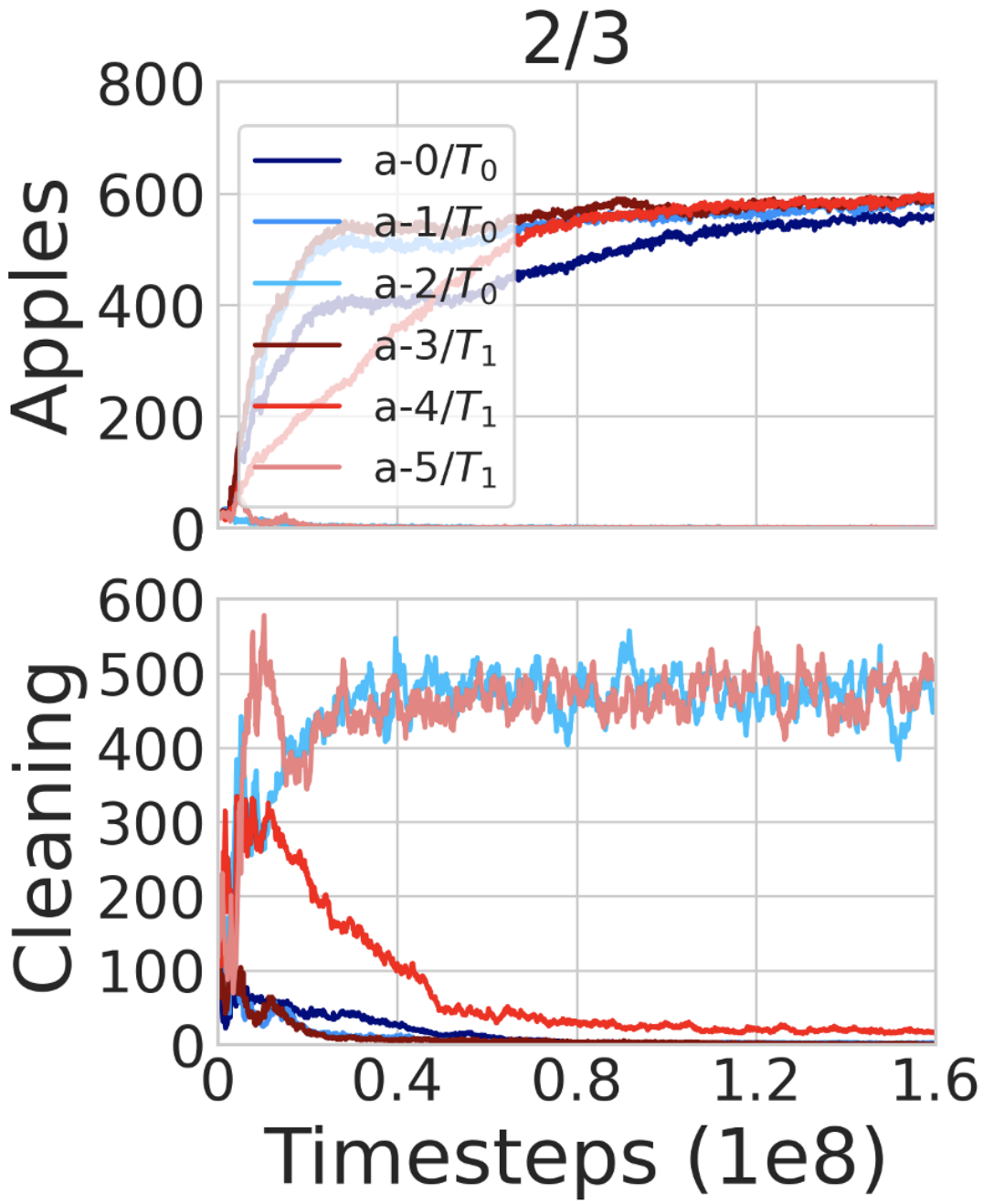

Figure 4.7: **Cleanup:** Two teams of three agents each. Mean number of apples picked (top) and cleaning beams selected (bottom) per-episode.

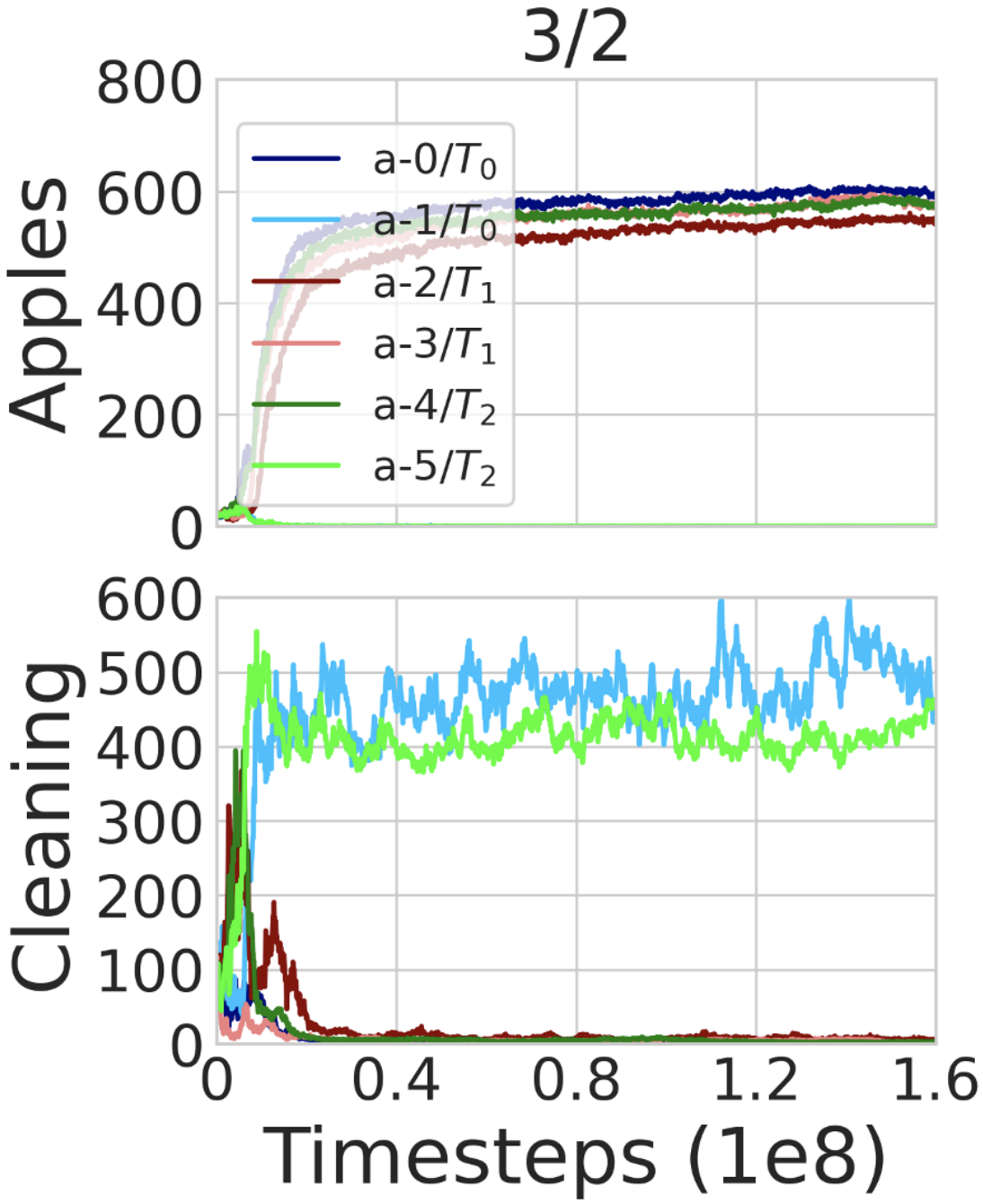

Figure 4.8: **Cleanup:** Three teams of two agents each. Mean number of apples picked (top) and cleaning beams selected (bottom) per-episode.

Division of Labor in Global Joint Policies

While agents in teams consistently learn to divide labor, the same numbered agent does not always learn the same behavior across different trials of our experiments. This makes aggregating multiple trials difficult. Therefore, Figures 4.5 through 4.8 show the mean apples picked (top) and cleaning beams selected (bottom) for each agent in one trial of our evaluation with each team structure (1/6, 6/1, 2/3, and 3/2). The behavior in this trial represents the most common division of labor for each team structure. Agents on the same team in each plot are presented as different shades of the same color (i.e., light red and dark red are teammates). The y-axis shows the number of apples collected or cleaning beam actions taken and the x-axis represents time. Agents rarely punish, thus we omit it from our analysis.

In the 1/6 configuration (Figure 4.5), two agents learn to mostly pick apples while four agents clean the river. While this represents the most common division of labor with 1/6, we do observe two trials where three agents learn to pick apples and three agents learn to clean the river. These strategies achieve high mean reward, but is not the best joint policy observed in our evaluation and consistently achieves less reward than the 2/3 and 3/2 team structures, discussed below. Shown in Figure 4.6, independent agents in 6/1 fail to significantly clean the river; therefore, few apples grow which leads to low rewards. Five agents free-ride on the labor of only one river cleaning agent. Figure 4.7 and Figure 4.8 show the 2/3 and 3/2 team structures respectively. We consistently observe populations of agents in these team structures divide into four apple pickers and two river cleaners. This division of labor joint policy achieves the highest reward in the Cleanup environment out of all joint policies we observed. The 3/2 team structure tends to learn this division slightly quicker (Figure 4.8), although both configurations eventually achieve basically the same reward on average as shown in Figure 4.3.

In summary, our results show how agents in team structures learn better specialization among the population by autonomously learning *roles* within their team. This allows populations in the 2/3 and 3/2 team structures to keep the river clean while most agents collect the spawning apples to collect reward. This causes 2/3 and 3/2 to achieve high mean population reward and equality across teams even though agents on different teams optimize for their own team's reward.

4.6 Discussion

This chapter shows that our model of teams has a significant impact on the development of agents' policies. In the IPD, we show how teams allow agents to immediately identify and cooperate with their teammates. Interestingly, we find that RL agents develop a pro-social convention and adapt this cooperative behavior towards non-teammates with specific team structures depending on the payoff scheme, even if defection has greater expected value. This behavior may be comparable with different levels of cooperation in humans, similar to increasing cooperation from only kin selection to notions of direct reciprocity with other groups [157].

While it was previously assumed that optimizing for signals from all agents (i.e., a fully cooperative population) achieves the highest reward in Cleanup [266, 52, 254, 145], our results indicate that agents optimizing for only a subset of the population (i.e., a team) and maintaining mixed-motives within the population achieves higher reward. Agent specialization in Cleanup is first identified by McKee et al. [145]. However, that work views specialization into a specific *role* of river cleaning agent or apple picking agent as a negative result that causes high labor inequality. We argue that the context of teams should change how role specialization is viewed in MARL. In the literature on Team Forming and Coalition Structure Generation, teams are often constructed to explicitly fill necessary roles [7]. We view role specialization as the agents autonomously learning these roles with only the feedback of their team's reward. This reinforces our hypothesis that teams can help improve how MARL agents learn to coordinate, and may be of specific interest to the emergent behavior community.

However, certain side effects may occur among teams depending on the defined team structure. While our 3/2 team structure achieves high reward in Cleanup, there is higher inequality than 2/3. To achieve the four apple picker and two river cleaner joint policy, one team (T_1 (red) in Figure 4.8) must free-ride on the labor of the other two teams. In practice, systems should consider potential side effects if slight inequality is detrimental to its welfare in the long-run, despite short-term stability. Furthermore, while we explore teams of AI agents, teams may also consist of humans or hybrid populations of both AI and humans. Exploring alternative team reward functions may lead to interesting results and future research, particularly in the context of hybrid teams.

4.7 Conclusions

This chapter provides an initial analysis into how team structures impact the development of individual learning agents' policies. While teammates in this setting share rewards and have common interest, mixed-motives are preserved between teams. Our results show that teams help agents develop pro-social policies in social dilemma domains despite game-theoretic incentives not to cooperate. In Cleanup, this leads to more globally productive joint policies than a fully cooperative population (1/6 team structure). This is significant considering that prior work assumes the fully cooperative population will achieve the best results in mixed-motive domains and has often compared their methods to fully cooperative outcomes. Instead, we find that a fully cooperative population may be sub-optimal and may not achieve the highest reward. In the next chapter, we relax the assumption that teammates fully share rewards and explore the impact of different degrees of mixed-motives among teammates.

Chapter 5

The Impact of Credo on Multiagent Learning

Chapter 4 explored the idea of how team structures impact the development of agents' policies in mixed-motive domains. While teammates in that setting had common interest by fully sharing rewards, mixed-motives were maintained between teams. A main finding is how different team structures help support the development of pro-social policies that can discover more efficient global joint policies than a fully cooperative population. In this chapter, we relax the assumption that teammates fully share rewards and explore settings where agents can partially optimize their behavior for various goals. We introduce a model to define how agents can optimize for different goals in the context of teams and analyze it's impact on the policies that individual learning agents develop.

5.1 Introduction

Humans have evolved with the inherent ability to cooperate and organize into teams. Some hypothesize that this has significantly supported our path to achieving higher intelligence [193, 240]. People tend to organize themselves into "teams-of-teams" within a larger system that are not in zero-sum competition, improving self identification and clarity of goals within a smaller group [143]. Today, these teams are present at various levels of complexity in order to survive, compete in sports, or complete tasks.

Wayne Gretzky, a former ice hockey player known as *The Great One*, describes a successful team as requiring "each and every [player] helping each other and pulling in the same direction". This statement, however, raises a number of questions.

- Do players on successful teams only optimize for the goals of their team?
- Is this strategy the best way for teams to achieve success?
- If not, under which conditions does optimizing for an alternative goal help or hinder overall success?
- Can incentives for individual goals actually promote behavior that is beneficial to the team?

In this chapter, we analyze how the performance and benefits of teams are impacted when learning agents may have different preferences by which they optimize their behavior. In multiagent reinforcement learning (MARL), agents learning to cooperate are often defined to have common interest through sharing exogenous rewards [3, 12]; however, purely pro-social agents may not be possible in practice. For example, consider scenarios where agents are designed by different manufacturers or hybrid AI/human populations interact. Agents in these settings may have some self-interest for personal goals. Therefore, it is important to understand how and when cooperation can be supported in systems where agents may partially optimize for multiple objectives.

In this chapter we introduce agent *credo*, a model which regulates how agents optimize for multiple objectives in the presence of teams. The noun credo, defined as "the aims which guide someone's actions" [224], describes our model of how agents optimize for goals. To be consistent with the analysis in the previous chapter, we analyze credo in mixed-motive social dilemmas popular in recent MARL research on cooperation [125, 249]. A common assumption made in past MARL literature is that aligning all agents' reward functions in mixed-motive environments is the strategy that will achieve the highest reward [266, 71]. Chapter 4 disproved this assumption by showing that multiple teams of fully aligned teammates achieves significantly more reward than the fully cooperative system. In this chapter, we discover multiple situations in which, despite some selfish preferences among agents, certain credo configurations with a defined team structure also significantly outperform the fully cooperative population. This chapter makes the following contributions:

- In Section 5.2, we augment the environment definitions in Chapter 3 and formally define the model credo in the context of multiagent teams.
- In Section 5.3, we study how the incentive structures of social dilemmas depend on the interaction between agents' credo and environmental variables.

- With learning agents, in Section 5.4 we show how different configurations of credo can lead to over 30% higher rewards than a fully cooperative population if agents partially optimize for personal or team-based goals.

5.2 Model of *Credo* with Multiagent Teams

The model of multiagent teams with individual learning agents used in this chapter is consistent with the model detailed in Chapter 3 and implemented in Chapter 4. In review, a *team* is a subset of agents which have some degree of common interest for team-level goals. Given a population, multiple teams with different preferences and interests may co-exist that are not in zero-sum competition. Consistent with Chapter 4, we refer to the number and size of all teams as a team structure and denote the set of all teams as $\mathcal{T}$, the teams agent i belongs to as $\mathcal{T}_i$, and a specific team as $T_i \in \mathcal{T}_i$.

This chapter augments agents' reward functions by introducing *credo*: a model to regulate how much an agent optimizes for different reward components it has access to. We relax the modelling assumption that teammates are bound through full common interest [184, 99, 13, 36] to study how different credos impact a system of learning agents. For example, an agent may optimize their policy for the performance of one or multiple teams, while also being somewhat oriented towards its own personal goals. We represent these guiding principles by decomposing the reward any agent i may receive from the environment into three components:

- IR_i: agents' individual exogenous rewards R_i.
- $TR_i^{T_i} \forall T_i \in \mathcal{T}_i$: the rewards i receives from each team for which they are a member.
- SR_i: the reward i receives from the system of N agents.

$TR_i^{T_i}$ and SR_i can be implemented with any function to aggregate and distribute rewards amongst multiple agents so long as agents are able to receive some amount of reward from these functions.

We define credo to be a vector of *parameters*, $\mathbf{cr}_i$, where the sum of all parameters is 1. The credo of an agent is represented by

$$\mathbf{cr}_i = \langle \psi_i, \phi_i^{T_1}, \ldots, \phi_i^{T_{|\mathcal{T}|}}, \omega_i \rangle,$$

where ψ_i 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 $|\text{self}| < |\text{teams}| \leq |\text{system}|$. Agent i's credo-based reward function $R_i^{\mathbf{cr}}$ is calculated as:

$$R_i^{\mathbf{cr}} = \psi_i IR_i + \sum_{T_i \in \mathcal{T}_i} \phi_i^{T_i} TR_i^{T_i} + \omega_i SR_i, \tag{5.1}$$

The environment in our analysis consists of a stochastic game with a model of team structure $\langle \mathcal{G}, \mathcal{T} \rangle$. Being consistent with Chapter 4, we continue to analyze the setting when agents belong to exactly one team. Formally, $\mathcal{T}$ is a partition of the population into disjoint teams, $\mathcal{T} = \{T_i | T_i \subseteq N, \cup T = N, T_i \cap T_j = \emptyset \forall i, j\}$. This team structure simplifies the credo vector for each agent to be $\mathbf{cr}_i = \langle \psi_i, \phi_i, \omega_i \rangle$, where ϕ_i is the credo parameter for i's team and we drop the team-specific superscript.

Any deterministic function can be used to calculate IR_i, $TR_i^{T_i}$ for any T_i, or SR_i in our model so long as any agent in a team or system receives reward for being part of the team or system (agents are part of the system by default). We implement functions to be consistent with past work by defining agents in a particular group to share rewards equally [254, 13, 99]. $IR_i = R_i$, the agent's normal individual reward function. Their team reward is defined as $TR_i^{T_i} : S \times A_i \times S \mapsto \mathbb{R}$, so that:

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

where teammates share their rewards equally, consistent with Chapter 4. The system-wide reward is defined as $SR_i : S \times A_i \times S \mapsto \mathbb{R}$ so that:

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

the mean reward of all N agents in the system. The final credo-based reward for agent i, $R_i^{\mathbf{cr}}$, is calculated using Equation 5.1 with these functions.

As is standard in many MARL problems, agents are trained to independently maximize their rewards. In particular, at time t each agent i selects some action a_i which together form a joint action $\mathbf{a}^t$. This action results in a transition from joint state $\mathbf{s}^t$ to joint state $\mathbf{s}^{t+1}$, according to the transition function P, and provides each agent i with reward

$R_i^t(\mathbf{s}^t, \mathbf{a}^t, \mathbf{s}^{t+1})$. Agents seek to maximize their sum of discounted future rewards, $V_i = \sum_{t=0}^{\infty} \gamma^t R_i^t$. Our model replaces R_i with $R_i^{\mathbf{cr}}$ at every timestep, reconfiguring the learning problem so agents must learn behavior that maximizes their sum of discounted future **credo-based** rewards according to the team structure and environment. This creates various dimensions of incentives that can impact the policies that agents learn through experience.

5.3 Equilibrium Analysis with Credo

We are interested in understanding the conditions under which credo may help or hinder cooperation. Thus, as a first step we investigate the impact of credo on the *stage game* of the IPD with teams. To provide a clear comparison with the standard IPD, similar to in Chapter 4, we take an ex-ante approach where agents are aware of their imminent interaction and the existence of other teams, but not the actual team membership of their counterpart.

Assume a pair of agents, i and 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 $(1-\nu)$. 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.

For the sake of our analysis, we make the assumption that all agents have the same credo. We calculate the expected values of cooperation and defection in situations where agents are fully self-focused ($\mathbf{cr}_i = \langle 1.0, 0.0, 0.0 \rangle$), team-focused ($\mathbf{cr}_i = \langle 0.0, 1.0, 0.0 \rangle$), and system-focused ($\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$). These values are then weighted by agents' credo parameters to consider all mixtures of possible parameters. We then calculate the conditions in which agent i has the incentive to cooperate as when the expected value of cooperation based on credo is greater than the expected value of defection. We include the full derivation in Appendix B.1. After algebraic simplification, we determine agent i is better off cooperating whenever:

$$\phi_i \left(\nu - \frac{2c}{b+c} \right) + \omega_i \left(\frac{b-c}{2} \right) \geq \psi_i c. \tag{5.2}$$

Note that this is independent of the strategy profile of their counterpart, σ_T (we remove the team notation from the subscript since the counterpart could be from T_i or T_j). Whenever

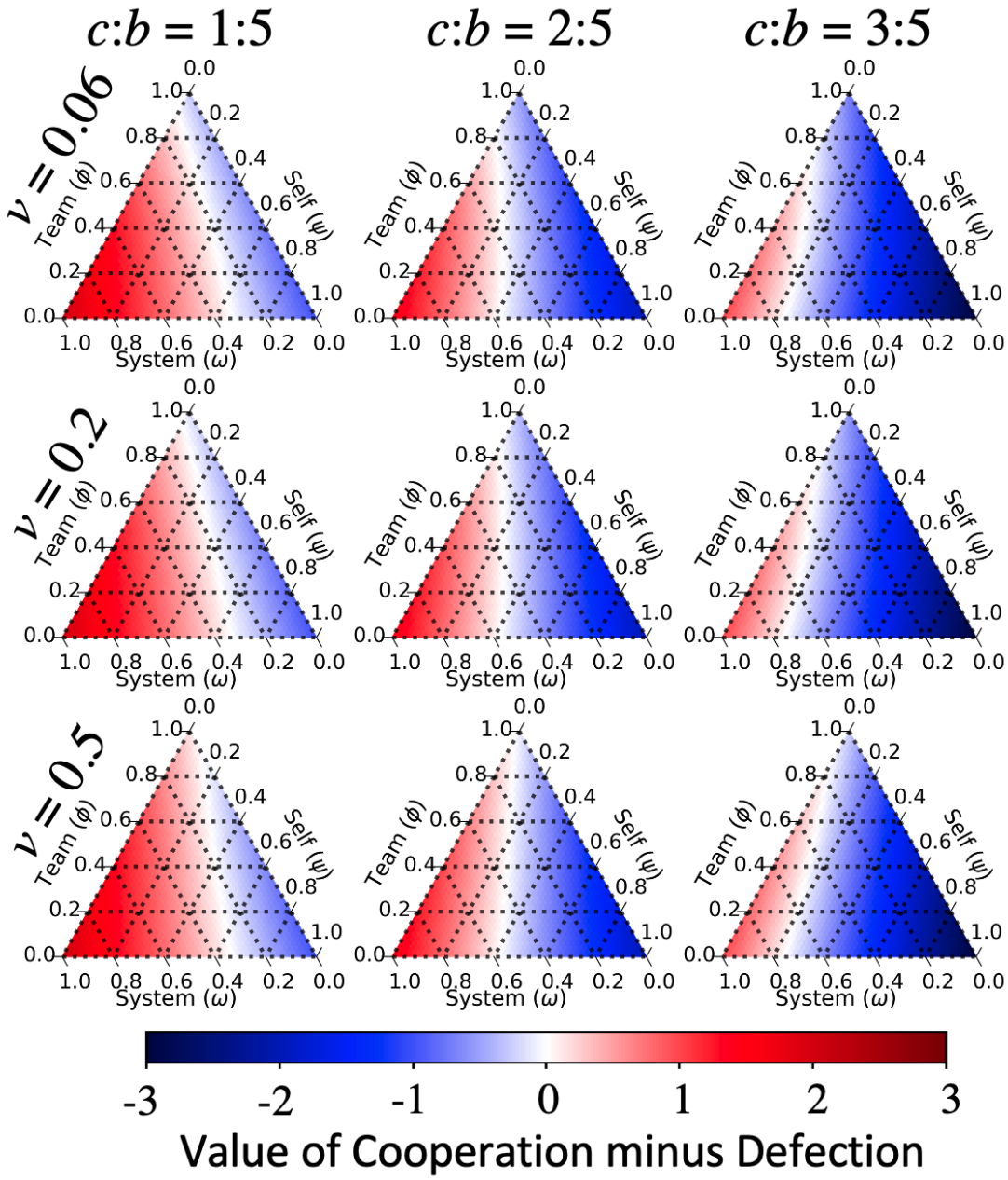

Figure 5.1: Impact of teammate pairing probability ν and the cost of cooperation c (benefit $b = 5$) on action incentives with credo. Red corresponds with cooperation being incentivized and blue corresponds with defection.

cooperation is the dominant strategy in a stage game, it will be supported in the repeated game.

Figure 5.1 shows the expected reward value of cooperation minus the expected reward value of defection by solving Equation 5.2 with $|\mathcal{T}| = 5$ teams. Each triangle shows the results for the linear combination of agent credo composed of self- (ψ_i; right axis), team- (ϕ_i; left axis), and system-focused (ω_i; bottom axis) parameters for an agent i (increments of 0.02). The colors indicate the expected value that agents would receive if they choose to cooperate; thus, colors correspond with the incentive to defect (blue) or cooperate (red), as computed by subtracting the value of defection from cooperation. White is used when this difference holds with equality.

Each row of plots represents different values of ν, the probability of being paired with a teammate. The remaining probability $1-\nu$ is spread across the $|\mathcal{T}|-1$ teams uniformly. With five teams, these values of ν represent when the chance of a counterpart being from another team is four times more likely than their own team ($\nu = 0.06$), being from any of the five teams has equal probability ($\nu = 0.2$), and being from the same team is four times more likely than another team ($\nu = 0.5$). Each column of plots represents a different cost of cooperation so that $c \in \{1, 2, 3\}$ with the benefit fixed to $b = 5$. For our entire analysis, we increase the cost and fix the benefit since we are interested in the ratio between the cost and benefit of cooperation instead of their absolute values.

We observe less overall incentive to cooperate as the cost c increases (i.e., darker blue and has more area inside the triangles). This pattern resembles findings observed in human behavior, where the amount of cooperation depends on the size of the benefit compared to the cost [210]. Another observation is that defection is incentivized in the presence of any amount of self-focus (right axes), with the exception of one environment ($c = 1$ and $\nu = 0.5$). Even in this scenario, defection becomes quickly incentivized as self-focus increases to $\psi_i = 0.2$. The following empirical experiments show that learning agents are able to develop globally beneficial cooperative behavior in multiple settings where defection is incentivized.

5.4 Empirical Evaluation

The following sections present the setup and results of experiments in the Iterated Prisoner's Dilemma (IPD) and Cleanup gridworld game environments using learning agents. Consistent with Chapter 4, we assume that for all teams $T_i, T_j \in \mathcal{T}$, $|T_i| = |T_j|$ (i.e., given a team model, the teams are the same size). This avoids complications that might arise with

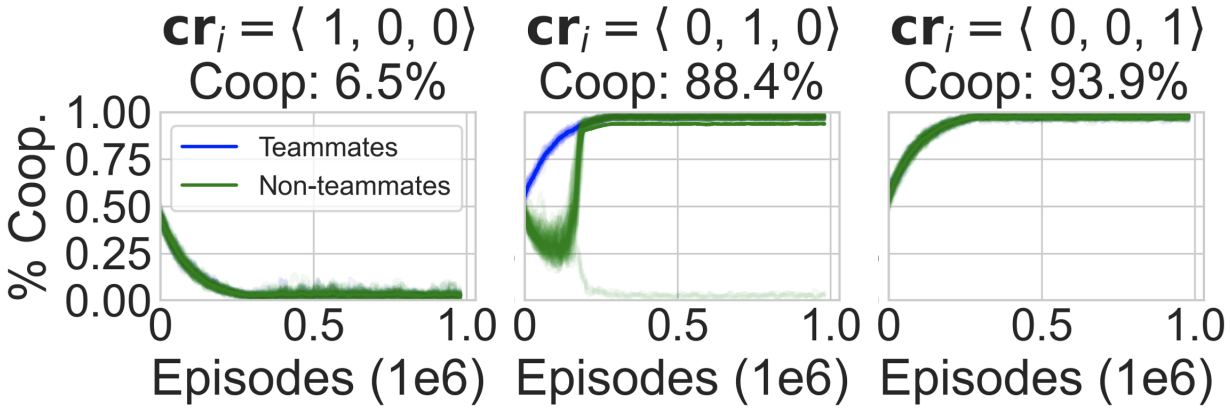

Figure 5.2: **IPD:** Fully self-, team-, and system-focused agents when $c = 1$, $b = 5$, $\nu = 0.2$ in a setting with five teams ($|\mathcal{T}| = 5$) of five agents each ($|T_i| = 5$).

agent interactions if teams were of significantly different sizes and to be consistent across our domains. We initialize $\mathbf{cr}_i$ to be the same for all agents *a priori* and do not change the parameters over the duration of an experiment. Since fully self-focused and system-focused credos have agents working as individuals (i.e., the standard non-team framework) and one full group (i.e., cooperative setting), they serve as benchmarks against which we can compare the performance of other credo with teams.

5.4.1 IPD Evaluation

In the IPD, each experiment lasts 1.0×10^6 episodes. We configure $N = 25$ Deep Q-Network (DQN) [153] agents into five teams ($|\mathcal{T}| = 5$) of equal size (i.e., five agents per-team). While our general team model allows for an arbitrary number of teams of any size, this work is concerned with the relationship between agent credo and environmental conditions.

An episode is defined by a set of agent interactions where each agent is paired with another agent and plays an instance of the Prisoner's Dilemma. Agent pairings are assigned based on ν, the probability of being paired with a teammate, and agents are unable to explicitly modify who they interact with, a challenging scenario for cooperation without additional infrastructure [4]. Each experiment is repeated for five trials. We analyze two types of credo distributions among the population: full-focus and multi-focus credo.

Full-Focus Credo

We start by analyzing how the behavior of agents is impacted by the extremes of full self-focus ($\mathbf{cr}_i = \langle 1.0, 0.0, 0.0 \rangle$), team-focus ($\mathbf{cr}_i = \langle 0.0, 1.0, 0.0 \rangle$), or system-focus ($\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$) credo, denoted as *full-focus* credo.

Figure 5.2 shows our results where the x-axis of each plot shows time and the y-axis shows the percent of actions where agents chose to cooperate, averaged over 2,000 episode windows (sliding window, increments of one episode). We set $c = 1$, $b = 5$, and $\nu = 0.2$ so counterparts have equal probability of being selected from any team (since $|\mathcal{T}| = 5$). Blue represents when the counterpart is a teammate and green when the counterpart is not a teammate.

When all agents are fully self-focused (left), $\mathbf{cr}_i = \langle 1.0, 0.0, 0.0 \rangle \forall i \in N$, they immediately learn defection towards all other agents (blue overlapped by green). When agents are team-focused (middle), $\mathbf{cr}_i = \langle 0.0, 1.0, 0.0 \rangle$, defection is the incentivized behavior as shown in Figure 5.1 (top corner of each triangle plot). However, we find agents quickly identify and cooperate with their teammates and almost every agent simultaneously develops stable pro-social policies with non-teammates despite not sharing rewards (similar to Figure 4.2 in Chapter 4). We hypothesize this is due to a combination of reduced reward variance for actions in specific states and interactions with teammates providing a strong positive feedback signal favoring cooperation. Only one agent over all trials learned defection against non-teammates, likely due to random initialization, although mutual cooperation is sustained among the other agents despite this defecting agent. In the right plot when agents are fully system-focused, agents learn to cooperate with every agent regardless of team (blue overlapped by green). The ϵ-greedy exploration algorithm prevents this percent of cooperation from ever reaching 100%. While other work requires strong assumptions of behavior to steer agents towards cooperation [3], these results indicate that full common interest may not be necessary to promote cooperation across an entire population with teams.

Heterogeneous Environment

Next, we experiment with settings where each of the five teams may have different credos within the same environment (e.g., 1 self-focused team, 3 team-focused teams, and 1 system-focused team). We use the same environmental settings as Figure 5.1, so that $\nu \in \{0.06, 0.2, 0.5\}$, $c \in \{1, 2, 3\}$, and $b = 5$ to understand how credo and environmental parameters impact how agents learn. This set of experiments is designed to understand how teams following different credos learn to interact with each other, and which credos have advantages in certain environments and population distributions. We assign teams *a priori*

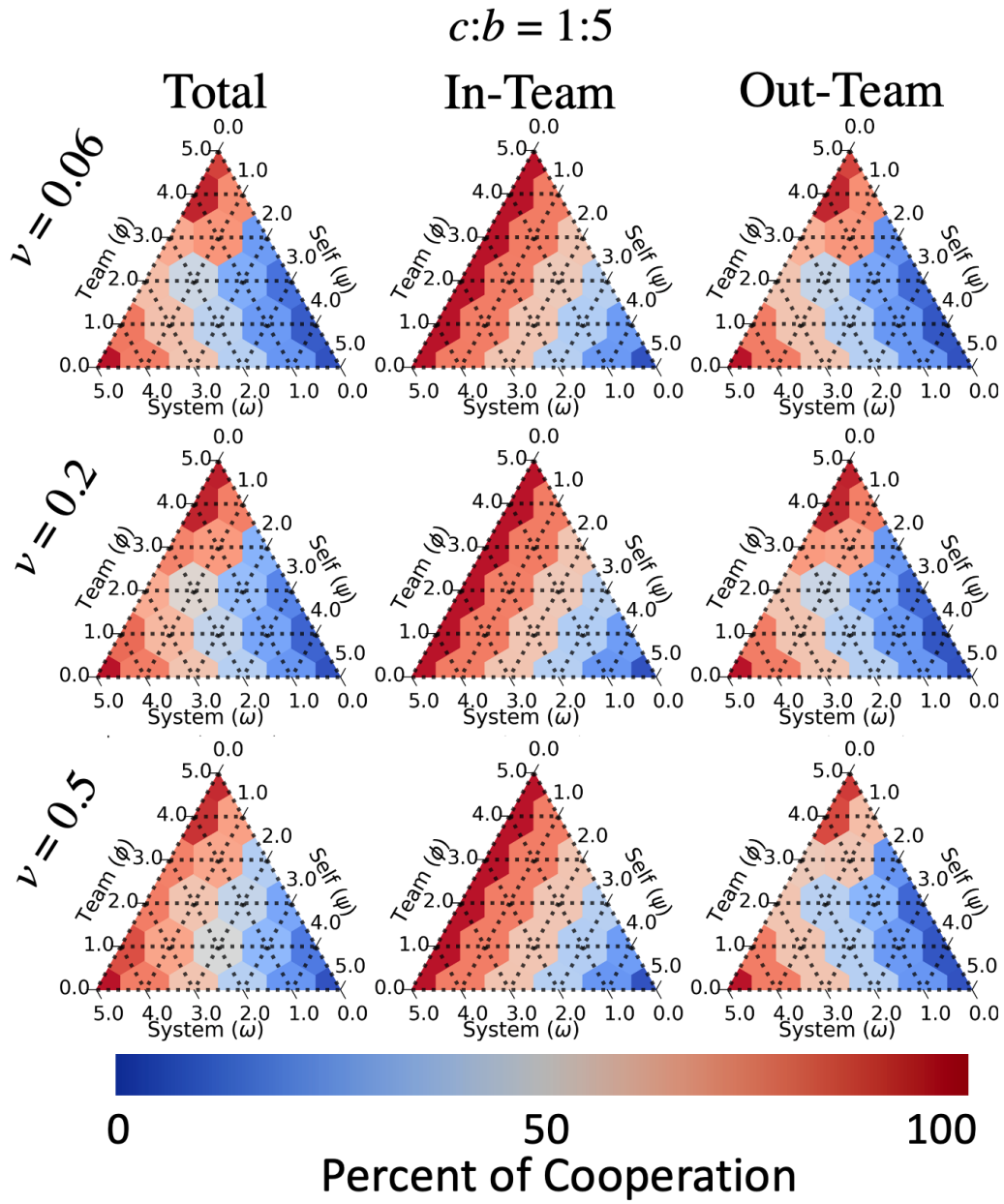

Figure 5.3: **IPD:** Cost is 1 and all agents follow different full-focused credos. Percent of actions that are cooperation (Total), only with teammates (In-Team), and only with non-teammates (Out-Team). We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$ and the benefit is 5.

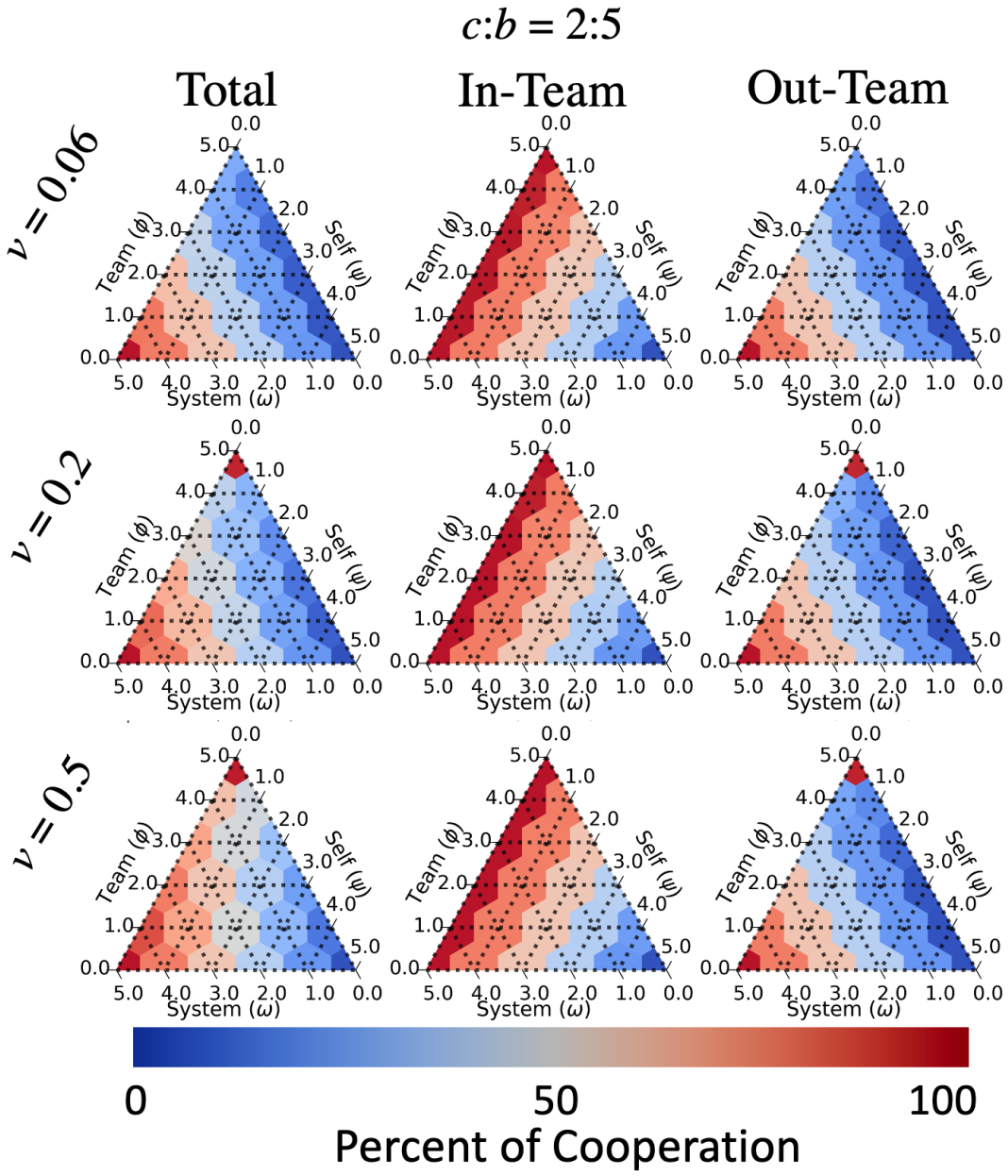

Figure 5.4: **IPD:** Cost is 2 and all agents follow different full-focused credos. Percent of actions that are cooperation (Total), only with teammates (In-Team), and only with non-teammates (Out-Team). We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$ and the benefit is 5.

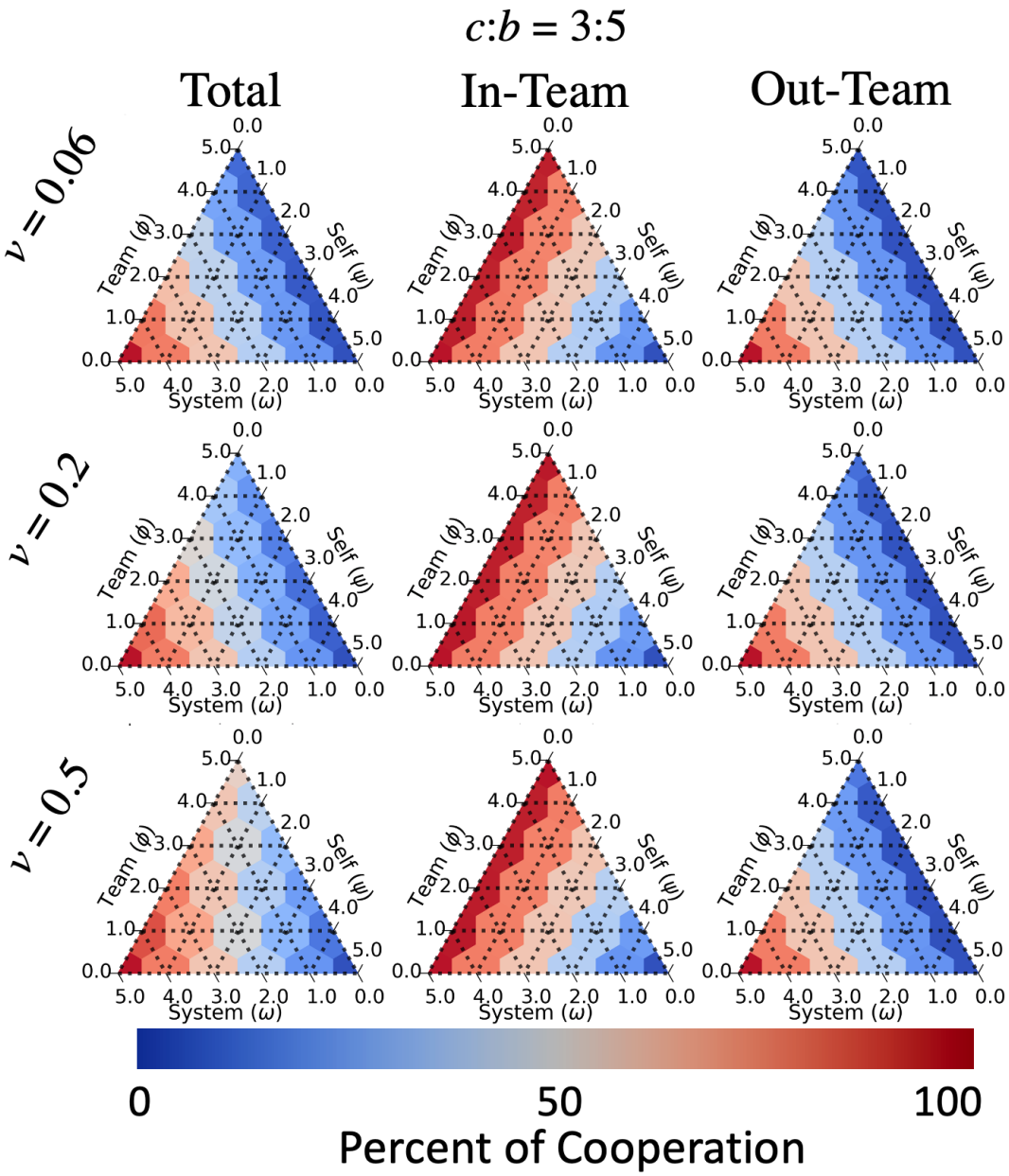

Figure 5.5: **IPD:** Cost is 3 and all agents follow different full-focused credos. Percent of actions that are cooperation (Total), only with teammates (In-Team), and only with non-teammates (Out-Team). We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$ and the benefit is 5.

to follow one of fully self-focused $\mathbf{cr}_i = \langle 1.0, 0.0, 0.0 \rangle$, fully team-focused $\mathbf{cr}_i = \langle 0.0, 1.0, 0.0 \rangle$, or fully system-focused $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$ credos for the duration of that experiment.

We inspect total, in-team, and out-team cooperation in each environment separately. Thus, since we fix the benefit $b = 5$, we plot when cost $c = 1$ (Figure 5.3), $c = 2$ (Figure 5.4), and $c = 3$ (Figure 5.5) in separate figures to allow us to further inspect total cooperation (left plots), cooperation with teammates (In-Team; middle plots), and cooperation with non-teammates (Out-Team; right plots). In the IPD, mutual cooperation yields the highest mean population reward. The hexagonal area around each intersection point is colored according to the global percent of actions which were to cooperate from blue (less cooperation) to red (more cooperation) over the last 25% of the timesteps.

Figures 5.3 to 5.5 show the percent of cooperation that arises when the five teams can follow different credos for different combinations of cost, benefit, and probability of being paired with a teammate ν. Each axis of each triangle plot shows the **number** of teams (out of five) that follow a particular credo (i.e., self-, team-, or system-focused). For example, the left corner of any triangle plot is when agents on all five teams are system-focused, the top corner is when all five teams are team-focused, and the right corner is when agents on all five teams are self-focused. The hexagonal area around each intersection point (within each triangle plot) is colored according to the global mean percent of actions that were cooperation from blue (less cooperation) to red (more cooperation) over the last 25% of timesteps.

We make several key observations in this setting. The amount of in-team cooperation does not change across various values of the probability of being paired with a teammate ν or cost of cooperation c (observed in all Figures 5.3 to 5.5) and is instead dependent on the number of self-focused teams in the environment. Thus, team-focused and system-focused agents learn cooperation with their teammates regardless of the behavior of other teams. Figure 5.3 shows that global out-team cooperation is able to be sustained even with the existence of one self-focused team; however, this trend disappears as cost increases. Observed in Figures 5.4 and 5.5, out-team cooperation is a function of the number of system-focused teams, meaning team-focused agents learn defection against non-teammates when the cost is higher (i.e., $c \geq 2$). When $c = 2$ (Figure 5.4), the existence of just one selfish-focused team prevents high out-team cooperation regardless of the value of ν. Despite this higher cost, out-team cooperation is supported when all teams are team-focused and $\nu \in \{0.2, 0.5\}$, but fails to materialize when teammates are rarely paired (i.e., $\nu = 0.06$).

In summary, out-team cooperation emerges when teammates are paired more often, there exist more non-teammates that are either team- or system-focused, and the cost of

cooperation is low. High levels of cooperation tend to be robust to a small number of self-focused agents in the population if the cost of cooperation is low; however, higher cost or more self-focused defecting agents tend to make agents learn to only cooperate with their teammates despite other potential gains to be had through mutual cooperation with agents in other teams.

Winning Credo

Figure 5.6 shows which full-focused credo the team that achieved the highest team-wide reward followed in each of the environments studied in Figures 5.3 to 5.5. While self-focused teams have the ability to gain more reward (a maximum of b in Table 2.1 compared to a maximum of $b - c$ in the common interest game in Table 3.2), our results show they often fail to achieve the highest reward. The orange hexagons indicate that team-focused teams overwhelmingly collect the highest team reward when playing the IPD with various combinations of other teams in the environment. Interestingly, we find that team-focused agents are able to maintain cooperative policies with their teammates in settings with large amounts of self-focused agents. When no team-focused teams are present, self-focused teams tend to dominate the system-focused teams; however, system-focused teams do better when $c = 1$ and $\nu = 0.5$ since cooperative system-focused teammates are paired more often.

One main result in Figure 5.6 is similar to findings in evolutionary game theory (EGT). At the single agent level, EGT has shown that a single defector will perform best in a population of cooperating agents [222]. Our work expands this finding to the team level and explores this concept with RL agents – a team of self-focused agents tends to perform best in a collection of system-focused teams (bottom of each environment). We find this is not the case only when self-focused agents are paired at least half of the time. We find that team-focused agents are able to maintain mutual cooperation with their teammates and other team-focused agents to achieve the highest rewards in the majority of settings. Contrary to EGT that states defection will spread over a population once introduced, the defective behavior of the self-focused team does not make team-focused agents' become non-cooperative with certain groups.

Losing Credo

Figure 5.7 shows the full-focused credo of the team that achieved the lowest mean team reward in each environment. Team-focused teams tend to only receive the lowest team reward in situations with low probability of being paired with a teammate ν and low cost of cooperation c (top left triangle plot). In these settings, team-focused agents get exploited by the self-focused teams they interact with more due to the low probability of being paired with a teammate ν, shown through high cooperation in the out-team setting

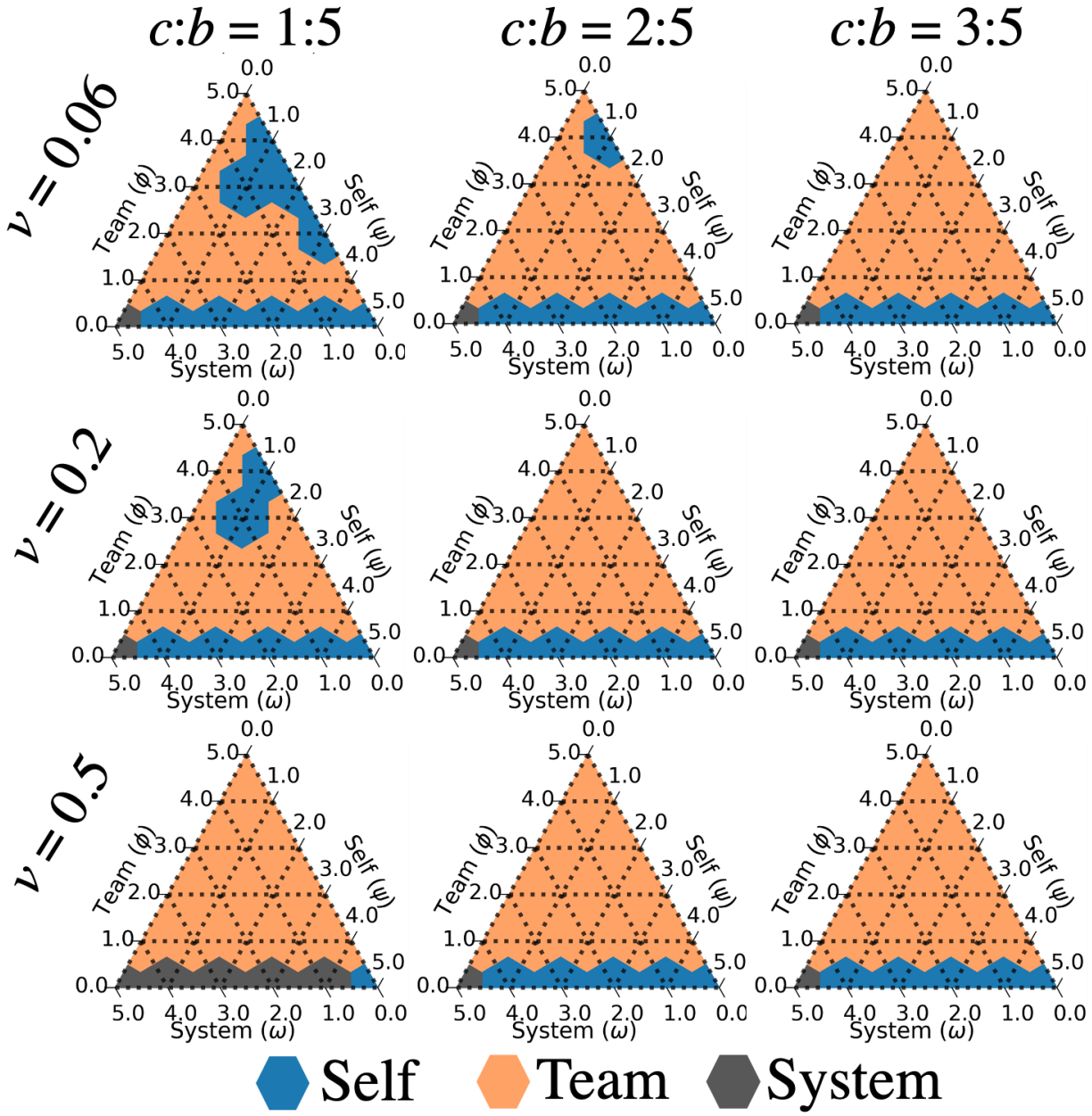

Figure 5.6: **IPD:** Full-focused credo that achieved the **highest** team-wide average reward in different environments. We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$, cost $\in \{1, 2, 3\}$, and the benefit is 5.

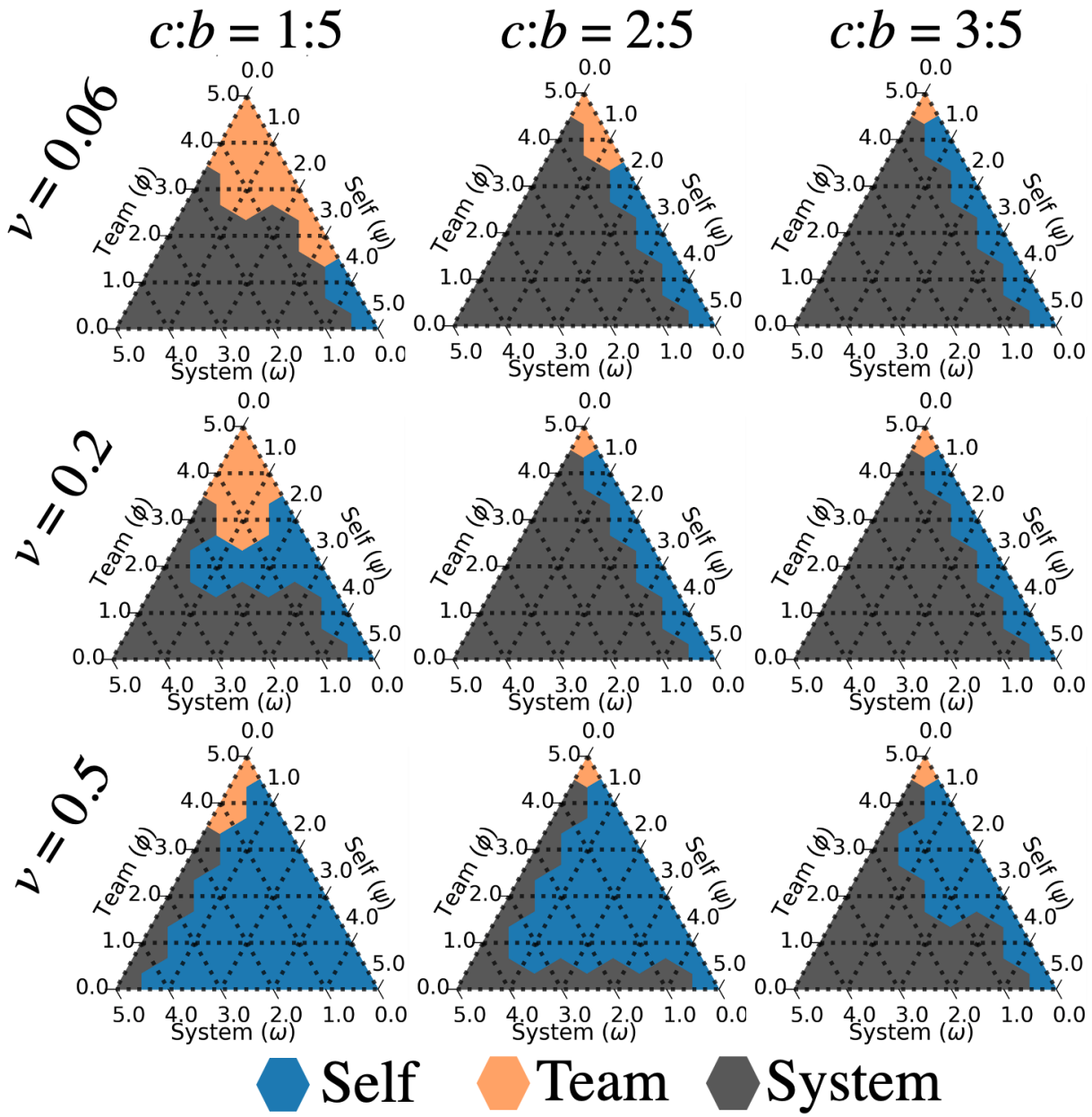

Figure 5.7: **IPD:** Full-focused credo that achieved the **lowest** team-wide average reward in different environments. We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$, cost $\in \{1, 2, 3\}$, and the benefit is 5.

in Figure 5.3. We also observe that system-focused teams tend to perform the worst when $c \geq 2$ and $\nu \leq 0.2$. This is due to self- and team-focused agents cooperating less with other teams as c increases; however, system-focused agents still attempt cooperation and get exploited. Contrarily, when ν is high and teammates are paired more often, self-focused teams perform the worst since selfish agents are paired together more often.

Similar to the results in Figure 5.6, our results in Figure 5.7 echo those of EGT. For example, our results show that system-focused teams tend to perform the worst in most environments when there is a self-focused team present (bottom of each environment). This expands the EGT result of cooperators doing worse in the presence of a few defectors [222]. We also find that system-focused teams tend to perform the worst in the majority of other environments when team-focused teams are present. This is due to less cooperation between different teams caused by the increased cost of cooperation c and the system-focus agents being exploited by the defecting self- and team-based agents.

Multi-Focus Credo

Next, we experiment with settings where agents can simultaneously partially optimize for their own, their team's, or the system's goals through various definitions of their credo parameters, denoted *multi-focus* credo. We use the same environmental settings as Figure 5.1, so that $\nu \in \{0.06, 0.2, 0.5\}$, the cost $c \in \{1, 2, 3\}$, and the benefit $b = 5$ to understand how credo and environmental parameters impact how agents learn. We evaluate the case where all agents have the same credo parameters, that is, $\mathbf{cr}_i = \mathbf{cr}_j \forall i, j \in N$.

Figures 5.8 to 5.10 show our results for various combinations of credo with 0.2 step credo increments, teammate pairing probability ν (rows), and the cost of cooperation c (columns). Each setting of different cost has nine different environments (three environments per-figure), each with 21 combinations of credo represented by the intersections of dotted lines from the three axes of each triangle in Figures 5.8 to 5.10. In the IPD, mutual cooperation yields the highest mean population reward. Similar to Figures 5.3 to 5.5, the hexagonal area around each intersection point is colored according to the global percent of actions which were to cooperate from blue (less cooperation) to red (more cooperation) over the last 25% of the timesteps. RL agents are able to condition their policy on the information of their counterpart's team, allowing us to observe how they learn behavior towards different groups. For each environment, we plot the total cooperation (left plots), cooperation with teammates (In-Team; middle plots), and cooperation with non-teammates (Out-Team; right plots).

Across all environments, agents achieve high cooperation when they have full system-

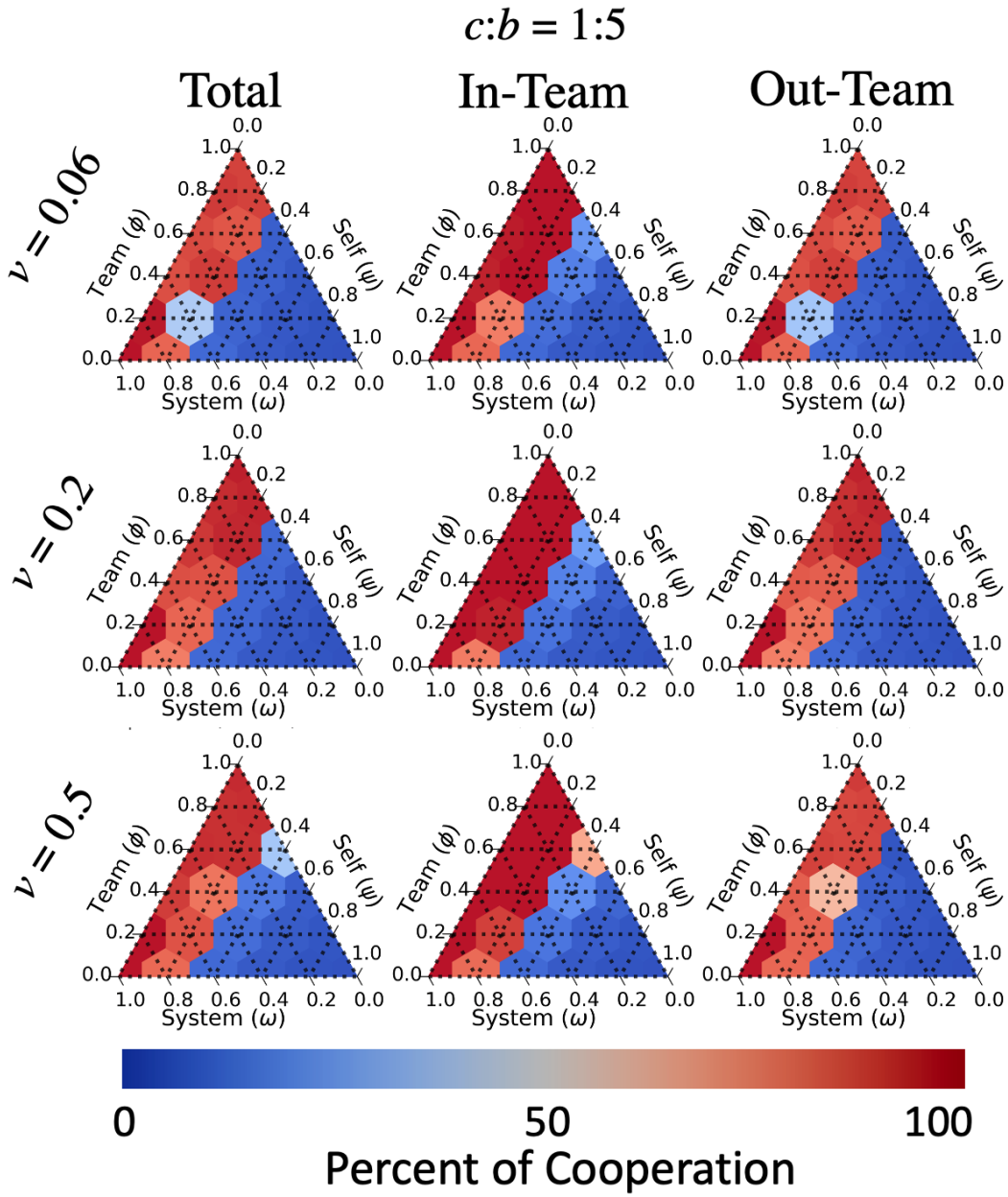

Figure 5.8: **IPD:** Cost is 1 and all agents follow the same credo. Percent of actions that are cooperation (Total), only with teammates (In-Team), and only with non-teammates (Out-Team). We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$ and the benefit is 5.

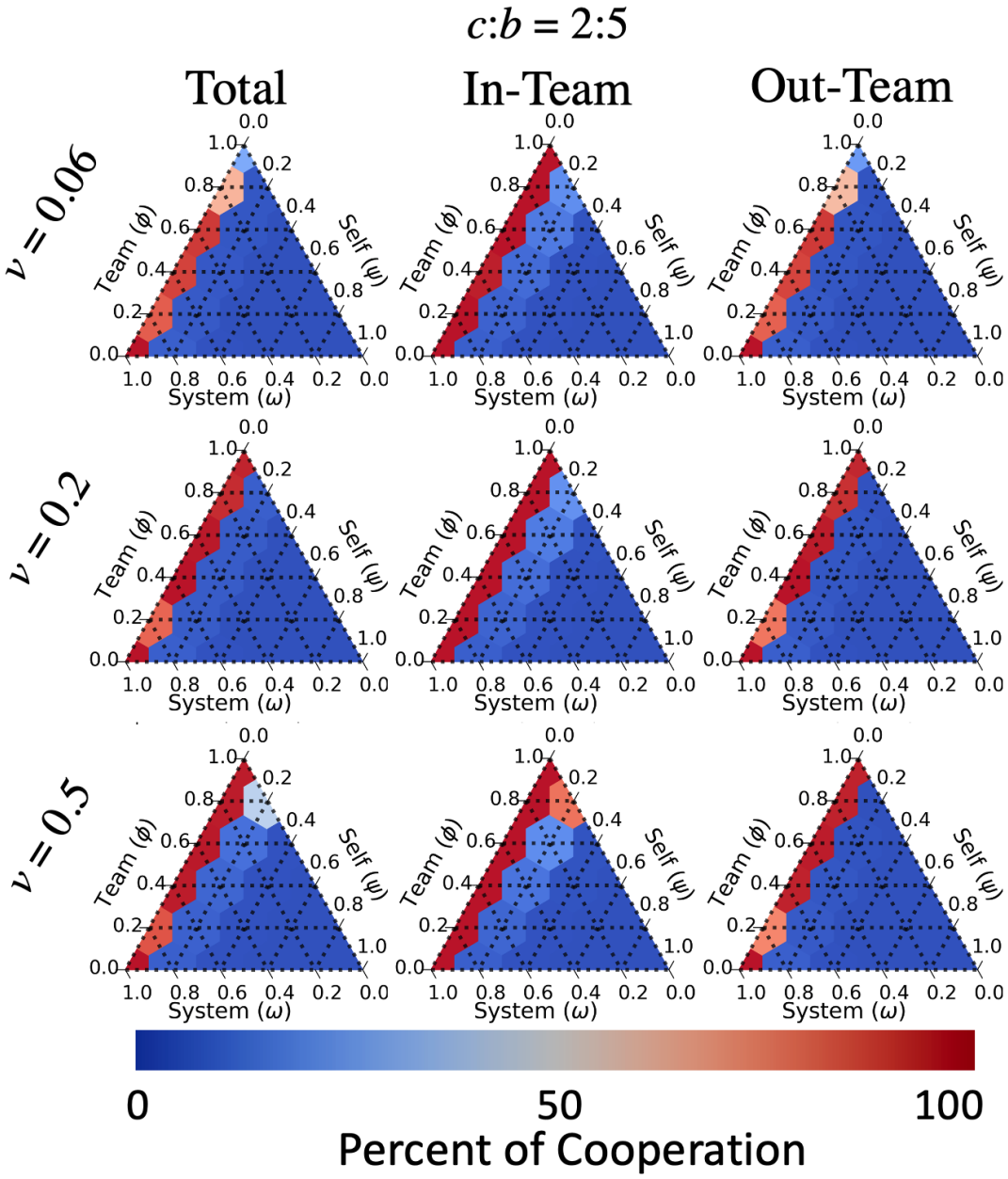

Figure 5.9: **IPD:** Cost is 2 and all agents follow the same credo. Percent of actions that are cooperation (Total), only with teammates (In-Team), and only with non-teammates (Out-Team). We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$ and the benefit is 5.

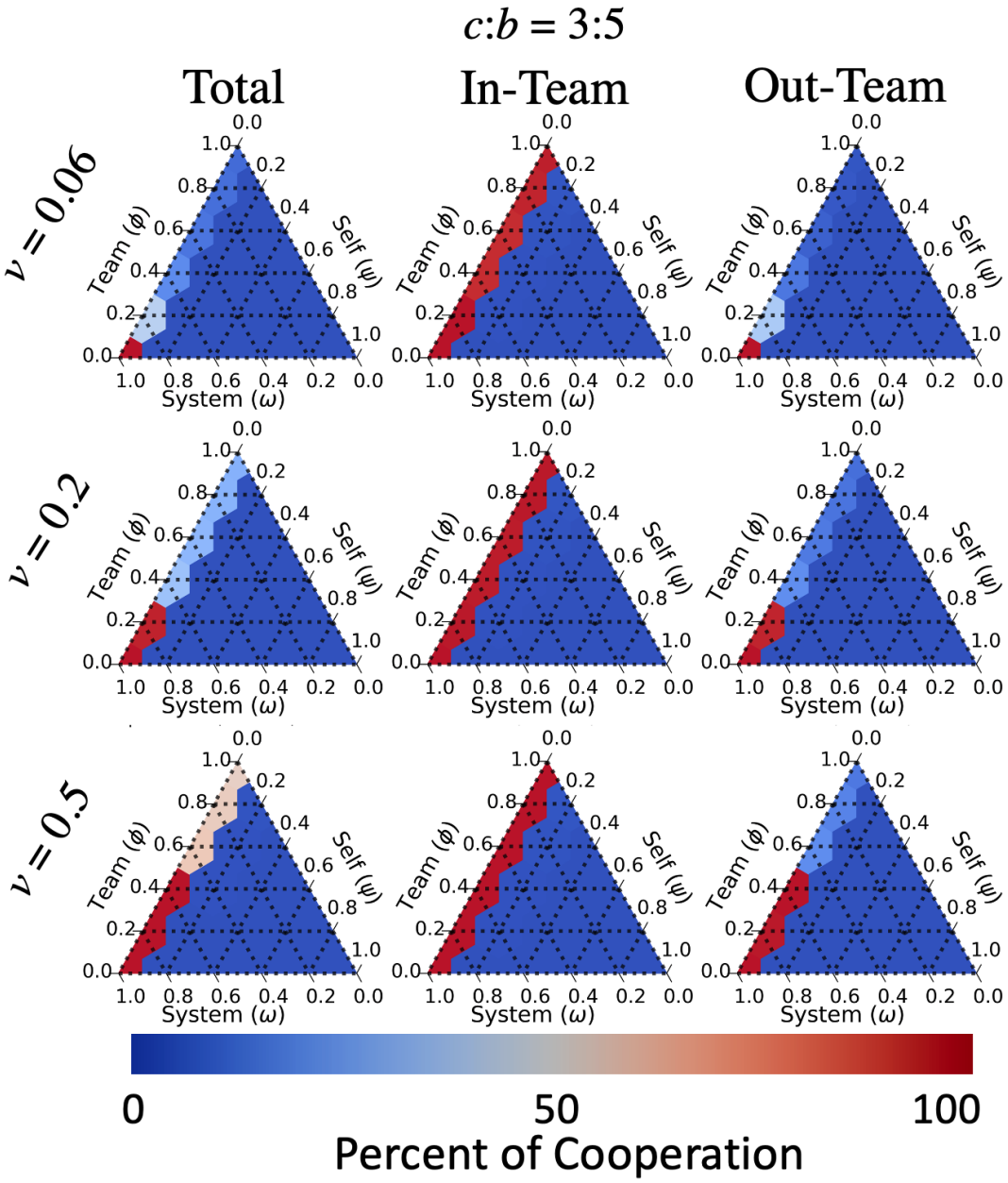

Figure 5.10: **IPD:** Cost is 3 and all agents follow the same credo. Percent of actions that are cooperation (Total), only with teammates (In-Team), and only with non-teammates (Out-Team). We experiment with different probabilities of being paired with a teammate $\nu \in \{0.06, 0.2, 0.5\}$ and the benefit is 5.

focus (left corners of each triangle) and learn defection when agents are fully self-focused (right corners of each triangle). Despite the incentive to defect in eight of nine environments, fully team-focused agents learn high cooperation in five environments (top corners of each triangle). The pattern of cooperation in each team-focused environment where agents learn cooperation is similar to the behavior in Figure 5.2, where agents adapt their cooperative behavior with teammates (In-Team) towards non-teammates (Out-Team). In environments with a lower cost of cooperation (i.e., $c = 1$ in Figure 5.8), cooperation is robust if full team-focus can not be achieved, such as when self-focus is $\psi_i = 0.2$. In these settings when agents are slightly self-focused, the rate of cooperation is higher when agents have high team-focus compared to high system-focus (i.e., darker red with high team-focus compared to system-focus when $\psi_i = 0.2$). However, this cooperation level depends on the probability of being paired with teammates ν and cost c. This is shown by both in-team and out-team cooperation decreasing as a function of ν observed in Figure 5.9 and no partially self-focused credo leading to cooperation when $c = 3$ in Figure 5.10. Unlike previous implementations of teams that assume agents have full common interest [99, 13, 36], we find teammates are not required to be fully aligned to achieve cooperative behavior in some settings.

These results show that teams of highly team-focused agents have the ability to support more cooperation when some self-focus exists than settings with high system-focus despite incentives to defect. Contrary to Gretzky's belief in the beginning of this chapter, our results indicate teams still achieve good results despite some self-focus among agents in teams. To understand this significance, consider situations where full common interest among teammates may not be guaranteed or all agents are unable to be controlled. Shown next, results in the Cleanup domain actually improve beyond full system-focus with certain credo parameters.

5.4.2 Cleanup Gridworld Game Results

Using the Cleanup environment, we experiment with $N = 6$ agents learning with Proximal Policy Optimization (PPO) [214] divided into three teams ($|\mathcal{T}| = 3$) of two agents each ($|T_i| = 2$) with the multi-focus credo setting (i.e., when agents can partially optimize their behavior for self, team, or system rewards simultaneously). Past work that has used the Cleanup domain typically uses five agents ($N = 5$) for a time of 1.6×10^8 environment steps (each episode is 1,000) [96, 254]. We increase the population to six agents ($N = 6$) to allow for three equal sized teams and calculate metrics over the last 25% of timesteps, similar to the IPD evaluation. Agent observability is limited to a 15 $\times$ 15 RGB window centered on the agent's current location. Teammates appear as the same color and optimize their own

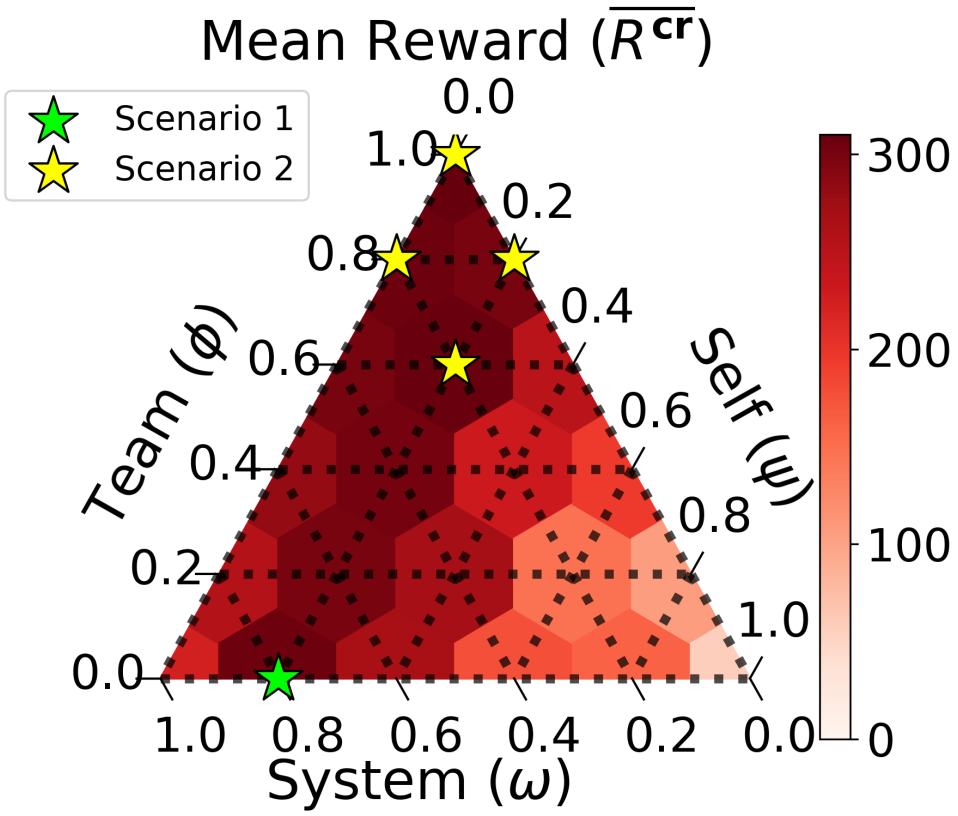

Figure 5.11: **Cleanup:** Mean population reward for every credo in our evaluation. These experiments have $|\mathcal{T}| = 3$ teams of two agents each. The scenarios with the highest reward often have agents with slight self-focus. We identify two types of credo scenarios that achieve the highest reward, when credo has slight self-focus paired with high system-focus (green star) and when team-focus is high (yellow stars).

$R_i^{\mathbf{cr}}$ after each timestep. Each experiment is completed eight times with different random seeds.

Mean Population Reward in Cleanup

While success in the IPD relies on agents choosing to cooperate in direct interactions, success in Cleanup requires agents to coordinate and form an effective joint policy to clean the river enough for apples to grow. Figure 5.11 shows the mean credo-based population reward per-episode $\overline{R^{\mathbf{cr}}}$ for all 21 credo configurations in Cleanup (intervals of 0.2), removing the subscript i when referencing all agents. The mean population reward gives insight into how well agents learn to solve the dilemma. Each hexagon corresponds with a combination of credo, centered at the intersections of three dotted lines from each axis (self, team, and system). Hexagons are colored according to the mean population reward from low (white) to high (red).

Fully self-focused agents fail to solve the dilemma, receiving the lowest mean population reward of any scenario. Previous work has found that the highest rewards in Cleanup are obtained when agents optimize for reward signals from all agents (i.e., system-focus) [254, 145, 71]. However, we find that some self- or high team-focus improves the mean reward significantly over the system-focused setting. We divide the five highest-reward environments into two scenarios shown in Figure 5.11. First, **Scenario 1**, when agents with high system-focus also have slight self-focus (Figure 5.11, green star), and second, **Scenario 2**, when agents have high team-focus relative to their other credo parameters (Figure 5.11, yellow stars). These scenarios achieve at least 30% higher mean population reward per-episode than the fully cooperative setting (system-focused; left corner).

Achieving High Reward in Scenario 1: The first scenario we examine is when highly system-focused agents have slight self-focus, $\mathbf{cr}_i = \langle 0.2, 0.0, 0.8 \rangle$ (green star). Agents with this credo achieve 33% higher reward per-episode compared to a population with full common interest. This result is comparable to a similar finding in past work [55], where a cooperative group performs best when agents have some selfish preferences of how to complete a task. This suggests that, despite using an entirely different domain, agents with high common interest but slight self-focus may consistently contribute to high group performance and is worthy of more exploration.

Achieving High Reward in Scenario 2: With the introduction of teams-focus, we more closely examine another credo scenario that contains four of the top five experiments with high mean population reward. The yellow stars in Figure 5.11 show experiments when agents have high team-focus relative to their other credo parameters, specifically

$\mathbf{cr}_i \in \{\langle 0.0, 1.0, 0.0\rangle, \langle 0.2, 0.8, 0.0\rangle, \langle 0.0, 0.8, 0.2\rangle, \langle 0.2, 0.6, 0.2\rangle\}$. As discussed in Scenario 1, agents with high system-focus experience a decrease in rewards when they are *too* system-focused. In Scenario 2, high team-focused agents achieve high rewards regardless if self-focus is zero or 0.2, echoing our result in the IPD that teammates are not required to have full common interest to achieve good results. These insights may be useful when attempting to influence credo in settings where agents are unable to guarantee their amount of self-focus or team commitment.

Division of Labor in Global Joint Policies

We find that agents in the highest-reward experiments (stars in Figure 5.11) often learn to divide labor and specialize to either clean the river or pick apples. This ability to coordinate with other teams and fill roles significantly impacts the global reward. This distribution of roles forms a global joint policy that we can analyze to determine how credo parameters impact the behaviors that agents learn. We observe the best division of labor strategy when two agents clean the river (i.e., cleaners) and four agents pick apples (i.e., pickers). In the following analysis, each line in Figures 5.12 and 5.13 represent the behavior of a single agent. For example, a line labeled "a-0/T_0" represents agent 0 belonging to team 0. Teammates appear as different shades of the same color (T_0 blue, T_1 red, and T_2 green). In adjacent trials, agents with the same label (i.e., a-0/T_0) may learn different behavior, making aggregating policies from all eight trials difficult. Therefore, we present figures from one trial representing the most commonly learned behavior in each setting.

Division of Labor in Scenario 1: We first analyze the division of labor in Scenario 1 (green star in Figure 5.11). Figure 5.12 shows the number of apples picked (top) and cleaning actions taken (bottom) by all six. The left plots show when agents are fully system-focused ($\mathbf{cr}_i = \langle 0.0, 0.0, 1.0\rangle$) and the right plots shows when agents are slightly self-focused ($\mathbf{cr}_i = \langle 0.2, 0.0, 0.8\rangle$). The full system-focused population develops into four cleaning agents and two apple pickers, with each cleaner receiving rewards from both pickers regardless of team membership (due to system-focus). This amount of reward suppresses any desire for cleaning agents to learn to pick apples, causing the population to reach a local minimum. The two apple pickers pick over 700 apples each resulting in a mean population reward of $R^{\mathbf{cr}} = 230.3$. However, increasing the self-focus to $\phi_i = 0.2$ (right plots) provides enough individual incentive to for four agents to learn to pick apples and collect 600 apples each for $R^{\mathbf{cr}} = 305.5$. Due to high system-focus, the two cleaning agents receive enough reward from all four pickers to incentivize them to continue cleaning, and the entire system achieves 33% higher reward by escaping the previous local minimum.

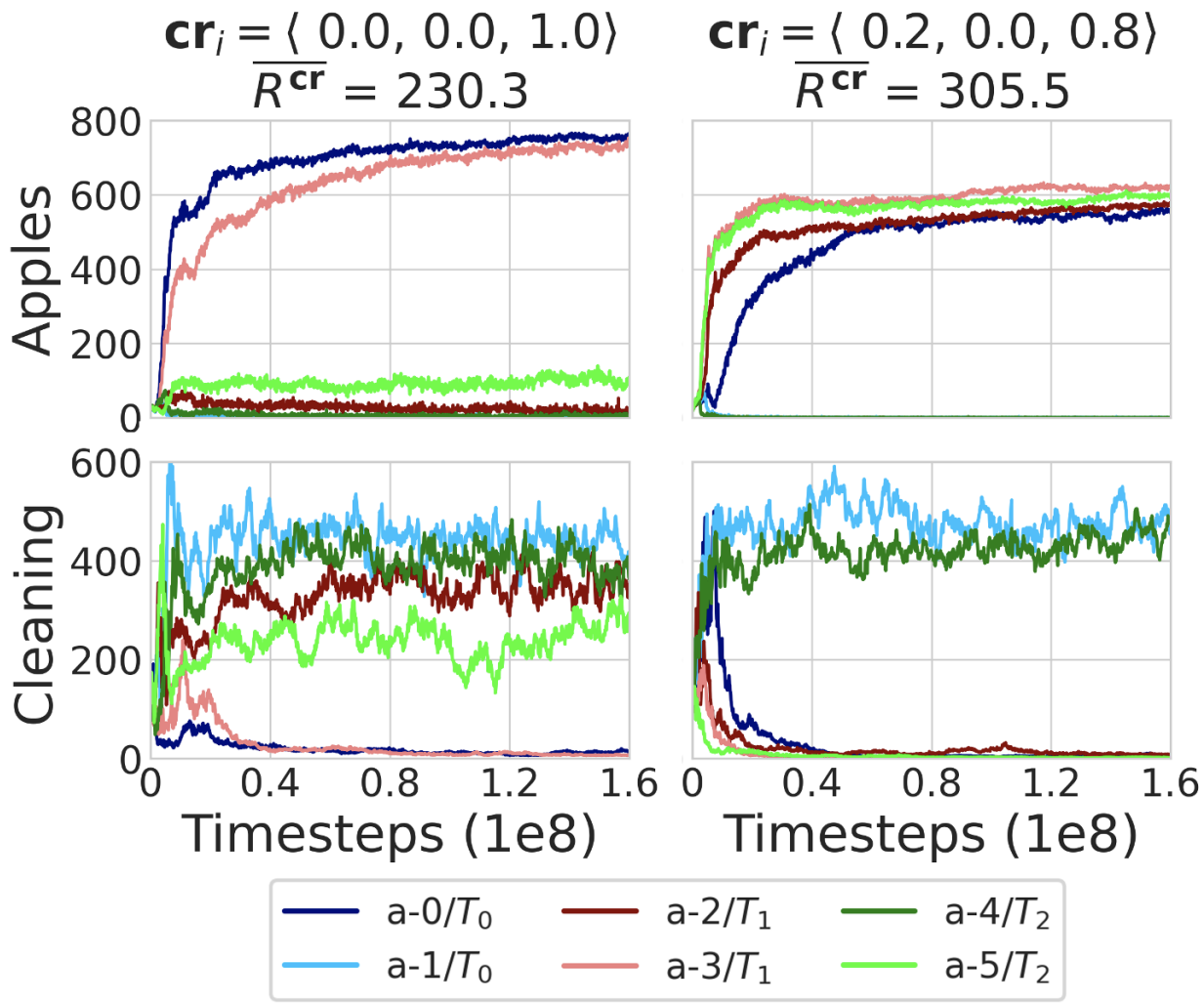

Figure 5.12: **Cleanup:** (Scenario 1) high system-focus with slight self-focus. Agents are labeled so that "a-0/T_0" represents agent #0 belonging to team #0. Each column of plots shows (left) fully system-focused agents and (right; green star in Figure 5.11) when agents become slightly self-focused. Better division of labor strategies are learned when self-focus increases from zero to 0.2 by enticing four agents to pick apples instead of just two, leading to 33% higher reward.

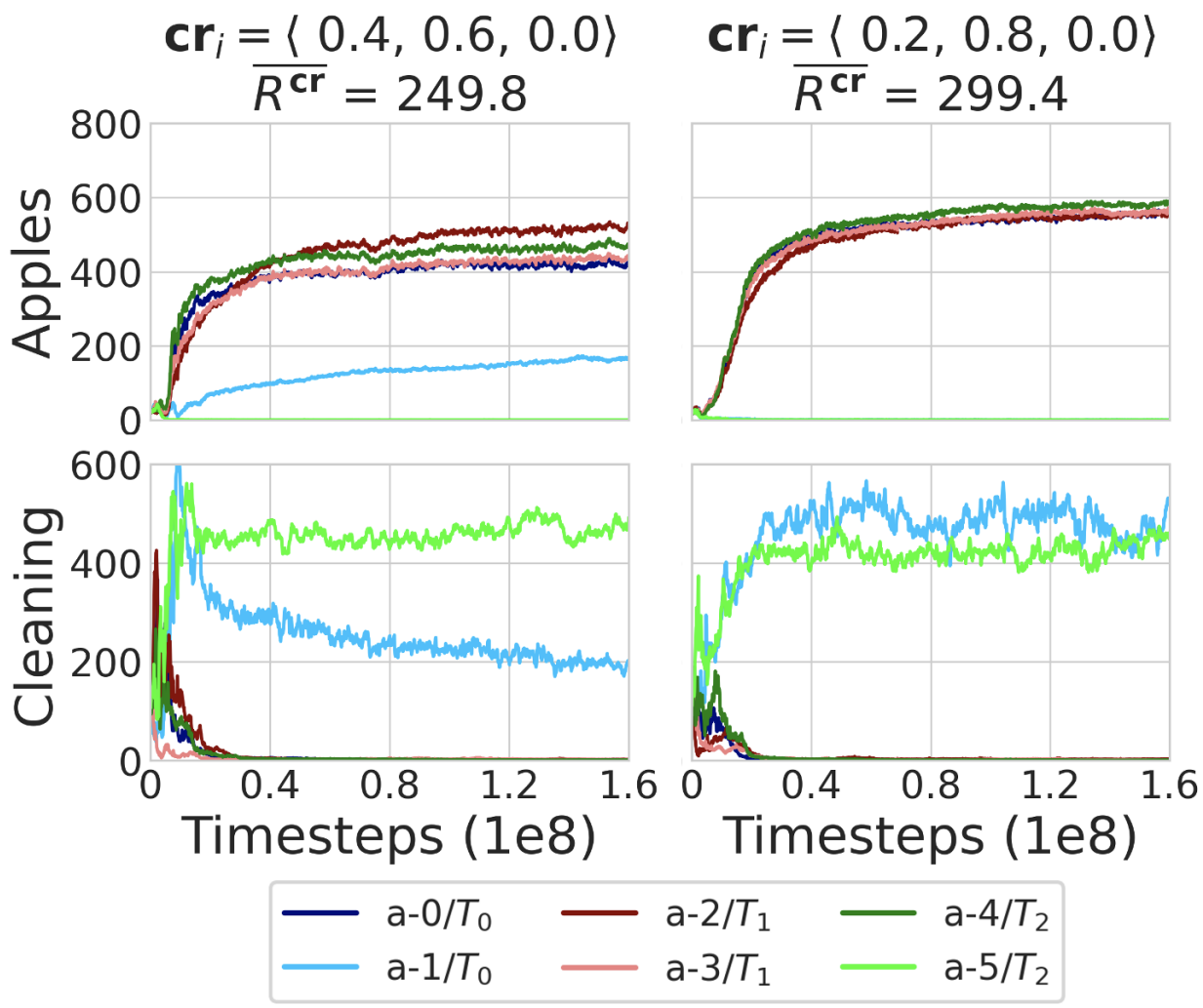

Figure 5.13: **Cleanup:** (Scenario 2) high team-focus achieves high rewards despite a small amount of self-focus. Each column of plots shows when team-focus is 0.6 (left) and 0.8 (right, a yellow star in Figure 5.11), offset with self-focus. As team-focus increases, two of the teams end up having one teammate cleans the river, leading to better global division of labor. This strategy is maintained when agents are fully team-focused as well.

Division of Labor in Scenario 2: We now analyze when agents have high team-focus compared to their other credo parameters (yellow stars in Figure 5.11). Of the four experiments in this scenario, we choose $\mathbf{cr}_i = \langle 0.2, 0.8, 0.0 \rangle$ and compare with $\mathbf{cr}_i = \langle 0.4, 0.6, 0.0 \rangle$.

The columns of Figure 5.13 represent when agents increase team-focus from 0.6 (left) to 0.8 (right), with the remaining credo being self-focus. When team-focus is 0.6 (left), only one team (T_2, green) learns to divide into the different roles of one river cleaner and one apple picker. While a-0 on T_0 (dark blue) fully learns to pick apples, their teammate (a-1) does not fully learn the role of river cleaner. This agent attempts to also pick apples and free ride on the cleaning of a-5. T_1 does not commit either agent to clean the river, resulting in fewer than two full river cleaners overall. This hinders population reward, since fewer than two total cleaners is unable to generate enough apples to support the remaining apple pickers. Thus, the four main apple pickers only collect an average of just over 400 apples each for a mean population reward of $R^{\mathbf{cr}} = 249.8$.

In the right column when agents have higher team-focus ($\mathbf{cr}_i = \langle 0.2, 0.8, 0.0 \rangle$, yellow star scenario), two teams learn to divide into one river cleaner and one apple picker, ensuring two agents are always cleaning. This produces enough apples for four pickers to collect about 600 each and both cleaners receive enough shared reward to overcome the incentive to free ride. As a result, the population earns $R^{\mathbf{cr}} = 299.4$, which is 20% more reward than when $\mathbf{cr}_i = \langle 0.4, 0.6, 0.0 \rangle$ (left) and 30% more reward than the full common interest setting $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$ (Figure 5.12 left). This division of labor is consistently learned when team-focus is high (yellow stars).

Overall, our results show specific combinations of credo support globally beneficial behavior among a population of teams. We expand a result from [55] to social dilemmas showing some selfishness improves group performance. Furthermore, we identify that agent specialization within their component teams results in high team-focus achieving more reward than fully system-focused credo.

Reward Equality Among the Population

Similar to Chapter 4, we now analyze reward equality among the population to understand if certain credo parameters create inequality. Since agents do not receive any exogenous reward for cleaning the river, it is important to consider the implications and potential side effects of credo and teams on population equality, or how evenly reward is distributed among a population of agents. We model population reward equality as the inverse Gini index, similar to past work [145]:

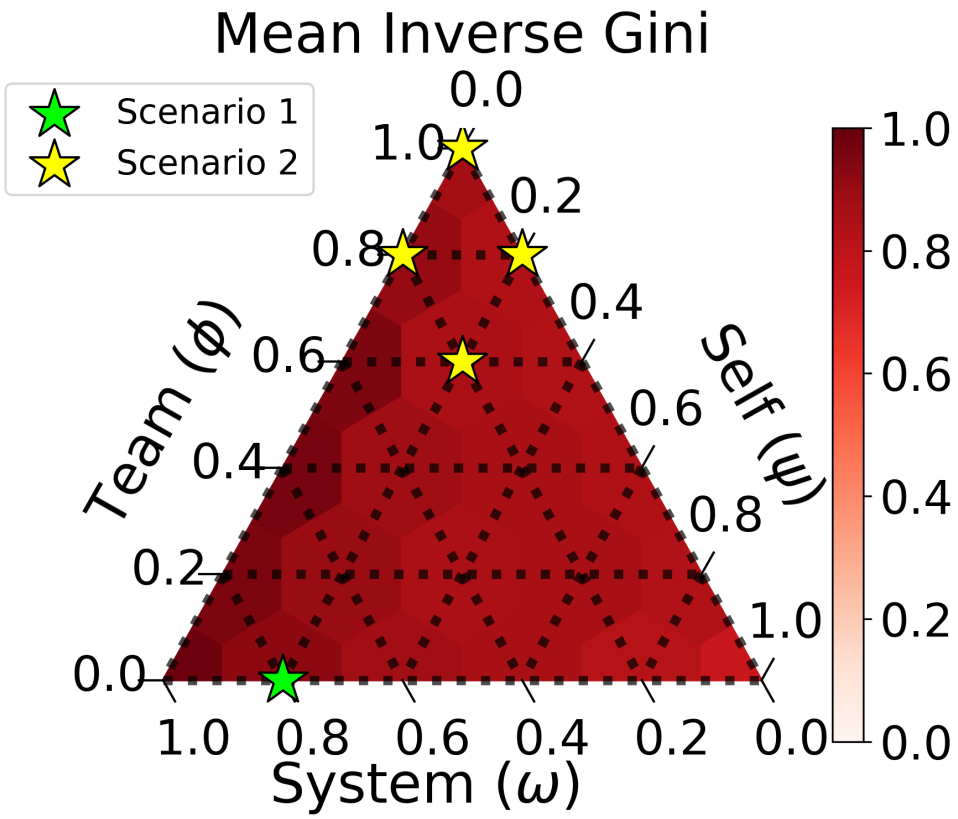

Figure 5.14: **Cleanup:** Inverse Gini index for every credo in our evaluation. These experiments have three teams ($|\mathcal{T}| = 3$) teams of two agents each ($|T_i| = 2$). Despite drastically different rewards, the credos that achieve high rewards also have high equality.

$$Equality = 1 - \frac{\sum_{i=0}^{N} \sum_{j=0}^{N} |R_i^{\mathbf{cr}} - R_j^{\mathbf{cr}}|}{2N^2 \overline{R^{\mathbf{cr}}}}, \tag{5.3}$$

where values closer to 1 represent more equality. Figure 5.14 shows our results for equality, where darker red corresponds with the reward being more equal across the population. The full system-focused case, by definition, has perfect equality since all agents share rewards equally. Scenario 1 still has high system-focus and also achieves high equality. In Scenario 2, we observe the two agents that learn to clean the river always emerge from two teams learning to divide their labor (i.e., one team does not only clean the river). Since each team has at least one apple picker agent, and agents have high team-focus to share rewards with cleaners, the population maintains high equality. Both scenarios achieve more equality than fully self-focused agents $\mathbf{cr}_i = \langle 1.0, 0.0, 0.0 \rangle$ while obtaining significantly higher reward.

5.5 Discussion

We proposed a model, credo, that regulates how agents optimize their behavior for different groups they belong to (i.e., self (a group of one), teams, or system). Our analysis serves as a proof of concept for exploring how agents simultaneously optimize for multiple objectives and learn to cooperate and coordinate. Our main contributions are two-fold.

- We show how agents form cooperative policies that are robust to some amount of self-focus. Furthermore, in a population where some agents may be fully self-focused, team-focused agents overwhelmingly achieve the highest reward.
- We find that agents achieve high population reward in Cleanup with high team-focus or slight self-focus (paired with high system-focus) compared to other combinations of credo parameters. This is achieved by agents learning more efficient global joint policies of division of labor under specific credo parameter combinations with teams.

Our results show how teams are not required to have full common interest to achieve high reward, unlike previous definitions of teams [99, 13, 36]. This has significant implications in settings where the amount of agents' self-focus may be unknown or full alignment of a group's interests may not be possible or desired.

A key takeaway of this work is that a fully cooperative system of individual learners may not achieve the highest reward, despite several recent studies using this scenario as

a basis for comparison [254, 71, 145]. This is consistent with our finding in Chapter 4 and is contrary to the spirit of Gretzky's observation from the introduction: a certain amount of personal striving can be beneficial for the overall system. We observed that fully cooperative populations did not take full advantage of the efficiencies of labor division and task specialization, which did arise when agents had some self- or team-focus. This tended to be particularly problematic as the number of agents sharing rewards increased, indicating a correlation between team structure, credo, and learning role specialization.

Similar to a recent single-agent finding [9], we hypothesize this is potentially due to the increased complexity in the credit assignment problem as the reward-sharing group increases in size. We explore this idea further in Chapter 6. The credo model offers a potential solution to mitigate this credit assignment problem in the context of teams. If agents' credo parameters are actively tuned to emphasize self- or team-focus, agents that are incurring credit assignment problems may gain a stronger feedback signal for their actions, guiding them towards better policies. On the other hand, agents initialized to be fully self-focused may converge to more cooperative policies if they tuned their credo to encompass teams or the system as a whole. In general, this may be viewed as a form of meta-learning where some credo-regulating policy learns to shape the environment for a lower-level behavioral policy. We envision a credo-tuning approach being implemented in two ways: centralized or decentralized. Using a centralized credo-tuner has the advantages of striving for global goals such as egalitarian or utilitarian ideas of equity, diversity, or productivity at a cost of potentially significant overhead. A fully decentralized credo-tuning model may make analyzing systems and the resulting equilibria even more challenging but carries the advantages of scaling linearly with the number of agents. We implement a preliminary decentralized model of self-tuning credo agents in Section 6.6 of Chapter 6 and achieve promising preliminary results.

We see more interesting directions for this work such as analyzing situations when teammates have different credo, how agents could influence the credo of teammates or other teams, or studying how results vary with different team structures. This includes understanding which processes achieve the highest behavioural influence in decentralized communities in the presence of team structures and group memberships. We hope that this work helps inspire future directions studying multiagent teams, multi-objective optimization, and the design and impacts of incentives to improve system performance.

5.6 Conclusions

This chapter introduced credo, a model that regulates how agents optimize for multiple objectives in the presence of teams. Agents' credo parameters determine how much of their reward function is influenced by various groups they belong to: themselves (a group of one), any teams they belong to, and the entire system. Credo relaxes the assumption that teammates fully share rewards. By implementing and studying credo, we have uncovered interesting theoretical and empirical results. Our results indicate that highly team-focused or slightly self-focused (with high system-focus) agents achieve the highest mean population reward in Cleanup, significantly higher than fully cooperative populations while maintaining high reward equality. These results are robust to some amount of self-focus among highly team-focused teammates and suggest some correlation between team structure, credo, and learning role specialization. While we have analyzed the influence of teams and credo on policy development in the previous two chapters, we have not provided a theoretical understanding as to *why* these groups perform better than a fully cooperative population. In the next chapter, we provide theoretical underpinnings to this result and expand on our hypothesis about the connection to the credit assignment problem.

Chapter 6

How Teams Impact Learning

The previous chapters explored how dividing a population of agents into teams and changing agents' credo parameters impacts the individual and joint policies that are developed. One main finding of the previous chapters is that team-focused agents, or system-focused agents with a small amount of self-focus, both achieve significantly higher mean population reward than a fully aligned cooperative population in some mixed-motive domains. This is significant considering that past research assumes a fully cooperative population achieves the highest reward in mixed-motive environments and uses this setting as a basis for comparing with their results [266, 71, 145]. Formally understanding why fully cooperative populations may be sub-optimal is important. While the two previous chapters use social dilemma domains, we now broaden the scope to understand general characteristics of environments that support this type of result. The goal of this chapter is to provide theoretical groundwork as to the conditions under which team structures properly support individual learning agents to develop better policies.

6.1 Introduction

To effectively work as a team, agents must learn coordination and cooperation with other agents in the environment. In settings with individual learning agents, teams are typically defined so that agents learn from their individual experiences, but share environmental rewards, creating a single team reward [144, 13]. In this chapter, we further investigate how teams and different team structures influence and guide the underlying learning process of individual agents.

Past researchers studying agent cooperation often compare their results with the fully cooperative population (i.e., a single team), assuming this population achieves the most reward in mixed-motive domains [266, 71, 145]. Furthermore, recent work [55] and our previous chapters have indicated that non-fully cooperative populations can actually learn more productive joint policies than a fully cooperative population. Even though a larger team has more agents at its disposal to perform tasks, smaller teams can achieve better global outcomes because agents learn more efficient joint policies. We have also shown that more effective joint policies are learned with some amount of self-focus credo. While these phenomena have been observed across multiple domains, the cause for better joint policies with mixed incentives is not fully understood.

This chapter provides theoretical groundwork as to *why*, and under *which conditions*, smaller teams outperform larger teams, as shown in Chapters 4 and 5. We focus on two areas of how teams impact learning:

1. How the introduction of teammates initially improves the ability for individual agents to learn about valuable areas of the state space (Section 6.3).
2. How the credit assignment problem (i.e. learning the value of taking a particular action) becomes more challenging as a team gets larger (Section 6.4).

These two axes of analysis characterize how teammates can be beneficial for learning to a point, since too many teammates can lead to sub-optimal results. In particular, we make the following contributions:

- In Section 6.3 we theoretically explore how teams can reduce the complexity of learning problems in certain environments.
- In Section 6.4 we show how sub-optimal team structures increase the difficulty for agents to identify valuable experiences, expanding previous work [9] to the multiagent team setting.
- In Section 6.5 we validate our theory empirically using widely used multiagent testbeds.

6.2 Background

We follow the same stochastic game definition and model of teams as the previous chapters. Much of our theory explores concepts within the internal dynamics of a single team even

though our theory depends on all N agents and scales to environments with multiple teams; thus, we use "team structure" to denote "team size" when considering a single team to remain consistent with previous chapters. As done previously, we refer to the set of all teams as $\mathcal{T}$, the set of teams agent i belongs to as $\mathcal{T}_i$, and a specific team that agent i belongs to as $T_i \in \mathcal{T}_i$. Agents on a team share rewards evenly by the definition in Equation 3.1. For readability, in this chapter we define the size of a team $|T_i| = n$ and modify the notation of the team reward function to be $TR_{i[n]}$ to emphasize the feature of team size.

Much of our theory relies on features of agent or team trajectories. We define $\tau_i = \{(s_i^1, a_i^1), (s_i^2, a_i^2), \cdots, (s_i^H, a_i^H)\}$ to be a trajectory of individual state-action pairs generated by agent i following π_i over H timesteps. A joint policy for T_i is the collection of individual behavior policies of all n agents in T_i, $\boldsymbol{\pi}_{T_i}$. A joint trajectory for team T_i, $\boldsymbol{\tau}_{T_i}$, is the collection of joint state-action pairs generated by agents in T_i. We are required to index trajectories in three ways:

1. $\boldsymbol{\tau}_{T_i}^t$ is the joint state and joint action at time t, $\boldsymbol{\tau}_{T_i}^t = (\mathbf{s}_{T_i}^t, \mathbf{a}_{T_i}^t)$.
2. $\boldsymbol{\tau}_{T_i}^{1:t-1}$ is the joint trajectory for team T_i up to time $t-1$, $\boldsymbol{\tau}_{T_i}^{1:t-1} = \{(\mathbf{s}_{T_i}^1, \mathbf{a}_{T_i}^1), \cdots, (\mathbf{s}_{T_i}^{t-1}, \mathbf{a}_{T_i}^{t-1})\}$.
3. $\boldsymbol{\tau}_{T_i}^{-t}$ is the H-timestep joint trajectory for team T_i without timestep t, $\boldsymbol{\tau}_{T_i}^{-t} = \{(\mathbf{s}_{T_i}^1, \mathbf{a}_{T_i}^1), \cdots, (\mathbf{s}_{T_i}^{t-1}, \mathbf{a}_{T_i}^{t-1}), (\mathbf{s}_{T_i}^{t+1}, \mathbf{a}_{T_i}^{t+1}), \cdots, (\mathbf{s}_{T_i}^H, \mathbf{a}_{T_i}^H)\}$.

Let $Z(\boldsymbol{\tau}_{T_i})$ be a random variable denoting the team random return obtained after team T_i completes the joint trajectory τ_{T_i} following their individual policies that compose $\boldsymbol{\pi}_{T_i}$. We define $Z_{T_i} \triangleq Z(\mathbf{s}_{T_i}, \mathbf{a}_{T_i})$ to be a random variable denoting the team reward observed at the joint state of all teammates $\mathbf{s}_{T_i}$ having taking joint action $\mathbf{a}_{T_i}$ and following their individual policies thereafter. Note that $\mathbf{s}_{T_i}$ is dependent on all N agents in the system by definition of stochastic games.

6.3 Identifying Valuable State-Action Pairs

We study the most restrictive case where teammates have no communication or coordination mechanisms and focus only on features of the team reward function. This isolates the impact that teammates have on the development of each others' policies to only their value functions. Therefore, we are able to analyze how teams distribute reward through

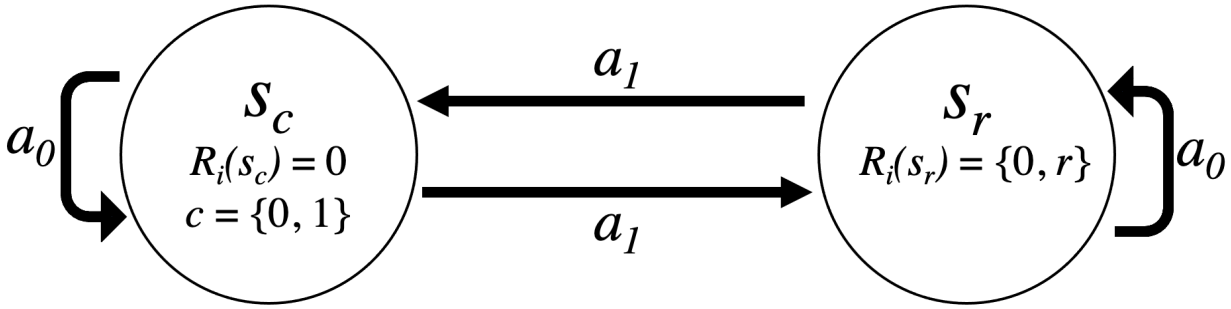

Figure 6.1: **2-States:** Diagram of our two state example environment. A stochastic game diagram with two agents is given in Figure 6.2.

the shared team-based reward function to understand how teams impact learning. By the definition of a stochastic game, rewards obtained from the environment depend on the joint states and actions of all agents. Thus, there can exist *reward-causing state-action pairs* – experiences that may not yield reward themselves, but allow reward to be obtained elsewhere in the environment [8]. Identifying these pairs can be challenging, since independently each state may provide little or no reward and agents need to learn about them indirectly.

We want to understand when teams of agents can leverage these reward-causing state-action pairs. We distinguish between the direct reward an agent receives from the environment when transitioning into their own observed state s_i^{t+1}, $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_i^{t+1})$, and the team reward, $TR_{i[n]}$. Note the condition on an agent's individually observed next state instead of the condition on the joint state. We identify an environmental property where the team reward signal is stronger than the individual reward signal (i.e., $\mathbb{E}\left[R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_i^{t+1})\right] < \mathbb{E}\left[TR_{i[n]}\right]$) because of the reward-causing state-action pair effect. This signal causes agents to become more attracted to, and thus learn to execute, reward-causing state-action pairs more often.

Consider the two state environment shown in Figure 6.1 – a modified version of the 4-States environment presented in Chapter 3 without the extra s_3 and s_4 states. This environment can support any number of agents ($N \geq 1$), and the state transitions and rewards depend on the joint action of all N agents. For simplicity, we assume the existence of only one team (i.e., $n = N$). Similar to the explanation of states in the 4-State environment in Chapter 3, we emphasize two types of states in the 2-States environment to assist our reward dynamics discussion:

- s_i^t is the physical state that agent i observes at time t.

- $\mathbf{s}^t$ is the joint state of all agents, or the state of the environment (i.e., the collection of s_i^t for all $i \in N$).

There exists two physical states that agents individually observe: s_c and s_r. Agents have two actions: stay at their current state (a_0) or move to the other state (a_1). The "c" in s_c corresponds with a binary signal (explained below) and the "r" in s_r refers to a reward state.

There is never a non-zero environmental reward given to agent i for transitioning to s_c, thus $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_c^{t+1}) = 0$. However, any agent (regardless of team affiliation) visiting s_c changes a binary signal c that allows reward to be collected at s_r. Thus, the possible rewards (dependent on c) given to any agent in s_r are $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_r^{t+1}) = \{0, r\}$, where $r > 0$. When agent i individually transitions to s_r, their reward (before sharing with their team) is $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_r^{t+1}) = 0$ if $c = 0$, and their reward is $R_i^t(\mathbf{s}^t, \mathbf{a}^t, s_r^{t+1}) = r$ if $c = 1$. Once reward is consumed at s_r, c has to be reset by visiting s_c again. Thus, the reward-causing state-action pair in this environment is to "visit s_c", causing a reward to be obtained when visiting s_r. With teams, the rewards given to individual agents for their actions are transformed into the team reward by Equation 6.1 for agents to learn from (same as Equation 3.1 with notation $TR_{i[n]}$, repeated for the reader).

$$TR_{i[n]} = \frac{\sum_{j \in T_i} R_j(S, A, S')}{|T_i|}, \tag{6.1}$$

Figure 6.2 shows a stochastic game diagram of this two state environment with two agents: i and j. The possible scenarios of the game are labeled so that $s_c(i, j)$ represents both agents being in physical state s_c. The reward represents the total reward yielded from the environment in that specific game state (i.e., reward $= 2r$ represents both i and j receiving r). For any agent to obtain the reward of r at s_r, some agent in the environment must visit s_c to change the binary signal to $c = 1$. With just two agents, there are multiple joint policies that yield optimal reward on which agents must learn to coordinate. Specifically, the two agents could 1) both move between s_c and s_r together, 2) transition from s_c to s_r (vice versa) with a_1 to never be in the same state, or 3) both agents use a_0 to always stays in s_c or s_r.

We can generalize features of this environment to support theory about how teams impact learning under certain conditions. In doing so, our theory is applicable to any multiagent environment where the following assumptions hold:

1. Agents' policies are initialized at random and fully explore the state space in the limit.

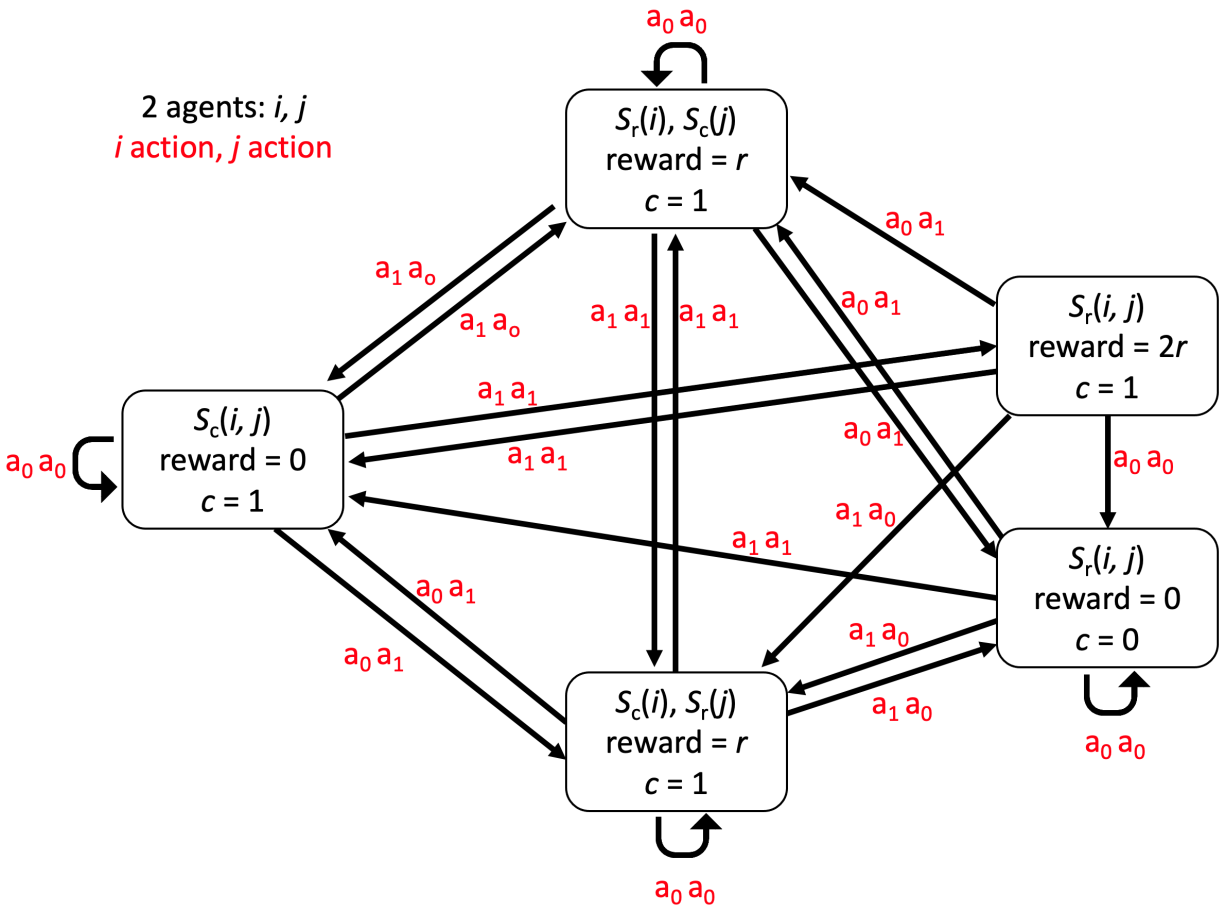

Figure 6.2: **2-States:** Stochastic game diagram induced from our two state environment in Figure 6.1 with two agents. Game states are labeled so that $s_c(i, j)$ represents both agents (i and j) being in physical state s_c.

2. The environment yields mid-episode rewards (not only at termination state) and any agent can collect a non-zero reward.

3. Executing a reward-causing state-action pair returns the minimum reward in the environment if the agent is not in a team (e.g., visiting s_c returns a reward of 0, the minimum reward of the environment in Figure 6.1).

Theorem 1. *There exists an environment where increasing the team size increases the probability of an agent receiving a reward for executing any reward-causing state-action pair that is greater than if they were not in a team.*

Proof. Due to agent's policies being initialized uniformly at random at the beginning of learning, we assume full coverage of the state space by all independent agents in the limit (Assumption 1). Subsequently, suppose agent i is executing a reward-causing state-action pair that yields the minimum reward in the environment (Assumption 3). Any teammate moving to a reward state increases the reward i receives for executing that reward-causing state-action pair through $TR_{i[n]}$ compared to when i acts individually. This is because i would receive the minimum reward in the environment for executing a reward-causing state-action pair if they did not have teammates; whereas, a teammate being in a reward state shifts some reward to the reward-causing state-action pair through $TR_{i[n]}$. Assuming independence among agents, the probability of any teammate j being in a reward state s_r is equal to the product of agents **not** being in s_r subtracted from 1. Let $0 < \zeta < 1$ be the probability that a teammate j is **not** located in a reward state, s_r, where $\zeta_j = \zeta_k$ for each $j, k \in T_i$ (i.e., ζ is assumed to be equal for all teammates). For a team of size n, the probability of any teammate being in a reward state at any timestep is $p(s_j = s_r) = 1 - \zeta^{(n-1)}$. Since $0 < \zeta < 1$, the second term $\zeta^{(n-1)} \to 0$ as $n \to \infty$. As a result, the overall probability of any teammate being in a reward state $p(s_j = s_r)$ converges to 1 as team size increases. □

This theorem has direct implications on the policy that i learns – more positive reward for executing a particular state-action pair will cause i to execute that pair more often. A larger team monotonically increases the **probability** that any teammate will be in a corresponding reward state and instantaneously share this reward with i through $TR_{i[n]}$. From the perspective of individual agents, this distributes the environment's reward function to other valuable areas of the state space.

Consider our two state environment in Figure 6.1. Agents individually receive a reward of 0 for visiting s_c but receive a reward of r for visiting s_r when $c = 1$. Without teammates ($n = 1$), agent i only receives the environmental reward when visiting s_c (i.e., reward of 0).

With teammates, i receives a reward (through $TR_{i[n]}$) of at least $\frac{r}{n} > 0$ when visiting s_c if at least one teammate is visiting s_r. The probability of receiving this reward (or greater) increases if there are more teammates that can be in reward states. The implications of this is that i will learn the benefit of executing it's part in a reward-causing state-action pair, leading it to execute this role more often.

6.4 Team Impacts on Credit Assignment

Whereas the previous section showed how introducing teammates increases the probability that agents receive a better reward for executing reward-causing state-action pairs (i.e., visiting s_c), this section analyzes the relationship between team structure and the distribution of rewards across all state-action pairs as a function of team size. We use information theory to explore how sub-optimal team structures impact the ability of agents to perform credit assignment despite a higher probability of receiving non-zero reward.

6.4.1 Information Sparsity in Single-Agent Settings

Credit assignment is concerned with identifying the value of past actions on the observed future outcomes and rewards. In single agent Markov Decision Processes (MDPs), information theory has been used to formalize conditions which make credit assignment infeasible, such as when the environment does not provide enough information (through reward) for an agent to learn an optimal policy [9]. We expand this concept to our setting. Let $s_i \in S_i$ and $a_i \in A_i$ represent any arbitrary state and action by an agent i within their individual state and action spaces. Following the single-agent case definitions in Arumugam et al., [9] (i.e., if $N = 1$), let Z_i be a random variable denoting the return for a single agent having taken action a_i in state s_i, and following π_i thereafter. The information gained by π_i is also a random variable, defined as:

$$\mathcal{I}^{\pi_i}_{s_i,a_i} = D_{KL}(p(Z_i|s_i,a_i)||p(Z_i|s_i)), \tag{6.2}$$

the Kullback-Leibler (KL) divergence between $p(Z_i|s_i,a_i)$, the distribution over returns for random state-action pairs conditioned on a particular state and action (i.e., the Q-value), and $p(Z_i|s_i) = \Sigma_{a_i \in A_i} \pi_i(a_i|s_i) p(Z_i|s_i,a_i)$, the distribution over random returns for the state-action pair conditioned on a particular state s_i (i.e., the value function). Equation 6.2 is the distributional analogue to the advantage function in reinforcement learning (RL),

$A^{\pi_i} = Q^{\pi_i}(s_i, a_i) - V^{\pi_i}(s_i)$, the difference between the value of taking action a_i at state s_i and the expected value of state s_i. Let d^{π_i} be the distribution of states visited and actions taken by i's policy. The expected amount of information carried by the actions of π_i about the return of those state-action pairs is defined as:

$$\mathcal{I}(A_i; Z_i|S_i) = \mathbb{E}_{(s_i,a_i)\sim d^{\pi_i}} \left[D_{KL}(p(Z_i|s_i, a_i)||p(Z_i|s_i)) \right]. \tag{6.3}$$

Difficulties with credit assignment emerge when $\mathcal{I}(A_i; Z_i|S_i)$ is small enough that the actions of a policy carry almost no correlation with the reward signal. Prior work defined an ϵ-information sparse MDP as when $\mathcal{I}^{\pi_i}(A_i; Z_i|S_i) \leq \epsilon$ for any initial policy at the beginning of training [9]. However, Equation 6.3 and ϵ-information sparsity do not fully translate to the multiagent team setting since they only consider the *expected* information. Teams modify the distribution, or variance, of information (Equation 6.2) across state-action pairs conditioned on the experienced values of the team reward random variable, Z_{T_i}.

For example, consider a **non**-ϵ-information sparse single-agent MDP environment where one state-action pair yields reward r and every other state-action pair gives a reward of zero. If r is divided evenly and distributed so that every state-action pair yields the same reward, $\mathcal{I}(A_i; Z_i|S_i)$ is unchanged (due to expectation) but the agent's policy carries no correlation with the reward signal. The agent would be unable to learn the same optimal policy as before (i.e., visiting the state which previously yielded r).

6.4.2 Information Sparsity with Teams

We enrich the definition of information sparsity in the context of stochastic games. This must consider two aspects of information. First, similar to before, the expected information gained by i's individual policy given their team reward function by substituting Z_i with Z_{T_i} in Equation 6.2, $\mathcal{I}^{\pi_i}(A_i; Z_{T_i}|S_i)$. Second, we must also consider the variance of information gained by i's policy over the distribution of their individual state-action pairs given their team reward function, $\text{var}\left[\mathcal{I}^{\pi_i}_{S_i,A_i}(Z_{T_i})\right]$ (see Appendix C.1 for the extended KL-Divergence derivation).

Definition 6. *Given a stochastic game with non-stationary policy class π_H, let $\boldsymbol{\pi}^0_N$ denote the set of initial policies for all N agents employed at the very beginning of learning. For small constants $\epsilon > 0$ and $\mu > 0$, a stochastic game is (ϵ, μ)-information sparse if:*

$$\sup_{\pi_i \in \boldsymbol{\pi}^0_N} \mathcal{I}^{\pi_i}(A_i; Z_{T_i}|S_i) \leq \epsilon,$$

or

$$\sup_{\pi_i \in \boldsymbol{\pi}_N^0} var\left[\mathcal{I}_{S_i,A_i}^{\pi_i}(Z_{T_i})\right] \leq \mu.$$

Definition 6 states that the actions of any agent i's policy (within their team's joint policy) given their shared team reward function must carry enough information with high enough variance for i to be able to learn. Otherwise, the stochastic game is considered (ϵ, μ)-information sparse. Low variance of information is detrimental to credit assignment since an agent would receive similar rewards regardless of their policy. By redistributing rewards, teams that fully share rewards can significantly modify var $\left[\mathcal{I}_{S_i,A_i}^{\pi_i}(Z_{T_i})\right]$ compared to settings without teams.

6.4.3 Risks of Sub-Optimal Team Structure

We now analyze convergence properties of the team reward as a function of team size, conditioned on the behavior of all N agents (i.e., global team structure). Since $TR_{i[n]}$ is determined by the experiences of all teammates, we focus on the joint policy of agents in T_i, $\boldsymbol{\pi}_{T_i}$, which determines the team return over a joint trajectory, $Z(\boldsymbol{\tau}_{T_i})$.

First, assume we have a stochastic game with N individual agents (no teams) that is **not** (ϵ, μ)-information sparse. By Definition 6, this environment has enough information with high enough variance for individual agents to be able to learn. Creating teams of agents in this game impacts the team structure and the reward signals agents learn from. In Section 6.3, we showed how increasing a team's size increases the probability of i receiving a better reward signal for executing a reward-causing state-action pair than without teams. However, this section shows how the ability to effectively identify these state-action pairs that cause reward depends on an appropriate team structure.

We now provide theoretical background to show how a sub-optimal team structure transforms this non-(ϵ, μ)-information sparse stochastic game into an (ϵ, μ)-information sparse stochastic game by decreasing the variance of information through $TR_{i[n]}$ below μ as team size increases. In practice, N (or the size of a team n) only needs to be sufficiently large to reduce the variance of information below μ as agents are grouped together in a team. This has implications on an agent's ability to perform credit assignment and learn an effective policy. To formalize this, we leverage a finding in Arumugam et al. [9] that we adapt to the multiagent team setting which equates information with reward entropy.

Proposition 1. *Let $\boldsymbol{\pi}_{T_i}$ be the joint fixed behavior policy of agents in T_i that generates a joint trajectory of experiences $\boldsymbol{\tau}_{T_i}$, where the randomness of state-action pairs in $\boldsymbol{\tau}_{T_i}$*

depends on all N agents (by the definition of a stochastic game). Let $TR^t_{i[n]}$ be a random variable denoting the team reward at any timestep t (where the randomness of the deterministic reward follows from the randomness of the joint state-action pairs of individual agents in T_i at time t, depending on all N agents, $\boldsymbol{\tau}^t_{T_i}$). It follows that:

$$\mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}^t_{T_i} | \boldsymbol{\tau}^{-t}_{T_i}) = \mathcal{H}(TR^t_{i[n]} | \boldsymbol{\tau}^{1:t-1}_{T_i}).$$

Proof. The chain rule of mutual information gives us:

$$\mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}^t_{T_i} | \boldsymbol{\tau}^{-t}_{T_i}) = \mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}^t_{T_i}, \boldsymbol{\tau}^{-t}_{T_i}) - \mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}^{-t}_{T_i}) \tag{6.4}$$

$$= \mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}_{T_i}) - \mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}^{-t}_{T_i}). \tag{6.5}$$

By the definition of mutual information, we can expand in terms of entropy:

$$= \mathcal{H}(Z(\boldsymbol{\tau}_{T_i})) - \mathcal{H}(Z(\boldsymbol{\tau}_{T_i}) | \boldsymbol{\tau}_{T_i}) - \mathcal{H}(Z(\boldsymbol{\tau}_{T_i})) + \mathcal{H}(Z(\boldsymbol{\tau}_{T_i}) | \boldsymbol{\tau}^{-t}_{T_i}) \tag{6.6}$$

$$= \mathcal{H}(Z(\boldsymbol{\tau}_{T_i}) | \boldsymbol{\tau}^{-t}_{T_i}) - \mathcal{H}(Z(\boldsymbol{\tau}_{T_i}) | \boldsymbol{\tau}_{T_i}). \tag{6.7}$$

We know $Z(\boldsymbol{\tau}_{T_i})$ is a deterministic function of $\boldsymbol{\tau}_{T_i}$ due to the deterministic aggregation (mean reward) of n deterministic reward functions of all teammates. The deterministic individual reward functions are already dependent on all N agents; thus, we can drop the second term and simplify to:

$$= \mathcal{H}(Z(\boldsymbol{\tau}_{T_i}) | \boldsymbol{\tau}^{-t}_{T_i}). \tag{6.8}$$

Since we know each agent in T_i is optimizing their discounted sum of future team rewards, we know $Z(\boldsymbol{\tau}_{T_i}) = \Sigma_{t=1}^{H} \gamma^{t-1} TR^t_{i[n]}$, and can substitute for $Z(\boldsymbol{\tau}_{T_i})$:

$$= \mathcal{H}(TR^t_{i[n]} | \boldsymbol{\tau}^{-t}_{T_i}), \tag{6.9}$$

$$= \mathcal{H}(TR^t_{i[n]} | \boldsymbol{\tau}^{1:t-1}_{T_i}, \boldsymbol{\tau}^{t+1:H}). \tag{6.10}$$

Finally, since $TR^t_{i[n]}$ is unable to be impacted by the future (i.e., anything greater than t), we can remove the correlation with $\boldsymbol{\tau}^{t+1:H}$:

$$= \mathcal{H}(TR^t_{i[n]} | \boldsymbol{\tau}^{1:t-1}_{T_i}). \tag{6.11}$$

□

The equality in Proposition 1 states that the information of the joint policy for team T_i at time t is equal to the entropy, a measure of missing information or uncertainty [219], of the team reward at timestep t, $TR^t_{i[n]}$, given the team-wide joint trajectory up to time t. For example, if $TR^t_{i[n]}$ returns the same value at each timestep regardless of the joint policy, the entropy of this reward function is zero and the information gained by the team's joint policy, and each agent's individual policy within this joint policy, is zero.

Our next step is to show how the variance of $TR_{i[n]}$ converges to zero as a function of increasing team size. The variance describes the distribution of potential team rewards given the randomness of state-action pairs experienced by agents in T_i.

Lemma 1. *The team reward random variable $TR_{i[n]}$ for any state-action pair converges to the mean environmental reward (mean of any agent's individual reward function) as team size increases in the limit (i.e., $TR_{i[n]}(\mathbf{s}^t, \mathbf{a}^t, \mathbf{s}^{t+1}) \to \overline{R_i}$ as $n \to \infty$).*

Proof. Since the team reward is an aggregation of n individual and uniformly random rewards samples from identical reward functions, $TR_{i[n]} \approx \mathcal{N}\left(\overline{R_i}, \frac{\sigma^2_{R_i}}{\sqrt{n}}\right)$ by the Central Limit Theorem, where $\text{var}[R_i] = \sigma^2_{R_i}$. The variance $\text{var}\left[TR_{i[n]}\right] = \frac{\sigma^2_{R_i}}{\sqrt{n}}$, with a derivative of $\text{var}\left[TR_{i[n]}\right]' = -\frac{\sigma_{R_i}}{\sqrt{n^3}}$. Since $\sigma_{R_i} = \sqrt{\sigma^2_{R_i}}$ is the standard deviation of R_i (i.e., distance from $\overline{R_i}$), we know $\sigma_{R_i} > 0$. Furthermore, σ_{R_i} is a constant and $n \geq 1$; thus, $\text{var}\left[TR_{i[n]}\right]'$ is negative and converges to zero as n increases in the denominator. □

Using Proposition 1 and Lemma 1, we conclude that the information carried by the joint policy of teammates over the joint trajectory $\boldsymbol{\tau}_{T_i}$ converges to zero as team size increases.

Theorem 2. *The information in a stochastic game at time t, $\mathcal{I}(Z(\boldsymbol{\tau}_{T_i}); \boldsymbol{\tau}^t_{T_i} | \boldsymbol{\tau}^{-t}_{T_i})$, converges to 0 as the size of a team, n, increases in the limit.*

Proof. By Proposition 1, we can use the entropy of $TR^t_{i[n]}$ to determine the information of $Z(\boldsymbol{\tau}_{T_i})$ at time t of a trajectory. By the Central Limit Theorem and Lemma 1, let $TR^t_{i[n]}$ be a Gaussian distributed random variable so that $TR^t_{i[n]} \approx \mathcal{N}\left(\overline{R_i}, \frac{\sigma^2_{R_i}}{\sqrt{n}}\right)$. For readability, let

the variance $\sigma^2 = \frac{\sigma^2_{R_i}}{\sqrt{n}}$. We rewrite the entropy of $TR_{i[n]}$ at time t given the joint trajectory up to t, $\mathcal{H}(TR^t_{i[n]}|\boldsymbol{\tau}^{1:t-1}_{T_i})$, in terms of the function's variance:

$$\begin{aligned}
\mathcal{H}(TR^t_{i[n]}|\boldsymbol{\tau}^{1:t-1}_{T_i}) &= -\int_{TR_{i[n]}} p(TR_{i[n]}) \log p(TR_{i[n]}) \\
&= -\mathbb{E}\left[\log \mathcal{N}(\overline{R_i}, \sigma^2)\right] \\
&= -\mathbb{E}\left[\log\left[\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}(\frac{R_i - \overline{R_i}}{\sigma^2})^2}\right]\right] \\
&= \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2\sigma^2}\mathbb{E}\left[(R_i - \overline{R_i})^2\right] \\
&= \frac{1}{2}\log\left(2\pi\sigma^2\right) + \frac{1}{2}
\end{aligned} \tag{6.12}$$

Since π is a constant, the variance $\sigma^2 = \frac{\sigma^2_{R_i}}{\sqrt{n}}$ regulates the entropy of $TR^t_{i[n]}$. By Lemma 1, we know $\lim_{n\to\infty} \frac{\sigma^2_{R_i}}{\sqrt{n}} \to 0$. Thus, the entropy and information carried by the actions of a policy in a stochastic game at time t converges to zero as their team size increases.

□

Since $\mu > 0$, defining larger teams **will** turn a non-(ϵ, μ)-information sparse stochastic game into an (ϵ, μ)-information-sparse stochastic game if the team is too large. In this setting, $TR_{i[n]}$ would not provide enough information about agents' individual policies and has implications on credit assignment, leaving agents unable to learn. Note that our theory uses equal reward sharing for $TR_{i[n]}$; however, the results of Theorems 1 and 2 are robust to any deterministic reward sharing function among teammates so long as teammates get some share of the team's reward. For Theorem 1, any deterministic reward sharing function will increase the probability of getting a higher reward for reward-causing state-action pairs as a function of the reward sharing group size. For Theorem 2, the only impact of an alternative sharing mechanism is on the convergence value of Lemma 1 (i.e., Lemma 1 uses a fixed point of $\overline{R_i}$). However, the entropy of the reward signal would still converge to zero regardless of the reward convergence fixed point (i.e., a modified Lemma 1).

Theorems 1 and 2 imply the existence of an *optimal* team structure. Increasing the size of teams can help agents identify reward-causing state-action pairs (Theorem 1); however, sub-optimal team structures carry the risk of infeasible credit assignment (Theorem 2). Since ϵ and μ are domain dependent, discovering the best team structure to help agents

learn remains subject to many domain specific variables. We can theoretically define a general rule that this team structure follows:

$$\begin{aligned} \max \quad & n \\ \text{s.t.} \quad & \sup_{\pi_i \in \pi_N^0} \mathcal{I}^{\pi_i}(A_i; Z_{T_i} | S_i) > \epsilon \\ & \sup_{\pi_i \in \pi_N^0} \text{var}\left[\mathcal{I}^{\pi_i}_{S_i, A_i}(Z_{T_i})\right] > \mu. \end{aligned}$$

To investigate features of this optimal structure in practice, we next empirically evaluate teams across multiple multiagent domains that support increasingly large populations of agents.

6.5 Empirical Results

In this section, we evaluate how the size of teams affect team performance and the policies agents learn. The learning algorithms used in this evaluation are Q-learning, Deep Q-Networks (DQN), and Proximal Policy Optimization (PPO) and our environments include 4-States, the Iterated Prisoner's Dilemma (IPD), the Cleanup Gridworld Game (Cleanup), and Neural MMO (NMMO). We consistently observe a similar trend across all domains: performance initially increases with more teammates, but decreases once teams are initialized to be too large. Thus, our results highlight a "sweet spot" team structure that helps guide agents towards learning good policies in different environments. We highlight features of agents' policies in each environment that provide further insight into their learning processes.

6.5.1 4-States Environment Results

An action transitions agents to their intended next state with 90% probability and to another random state with 10% probability. We fix the number of teams to be one ($|\mathcal{T}| = 1$) and increase n by a factor of 2 to remove the impact of other teams on the binary signal. Agents use Q-Learning with $\gamma = 0.9$ and ϵ-exploration ($\epsilon = 0.3$) for 50 trials of 1,000 episodes (100 steps each).

Due to the small number of states, larger teams in 4-States can generate more reward, even if agents act randomly (more agents can collect a reward of 1 each in s_r). Thus,

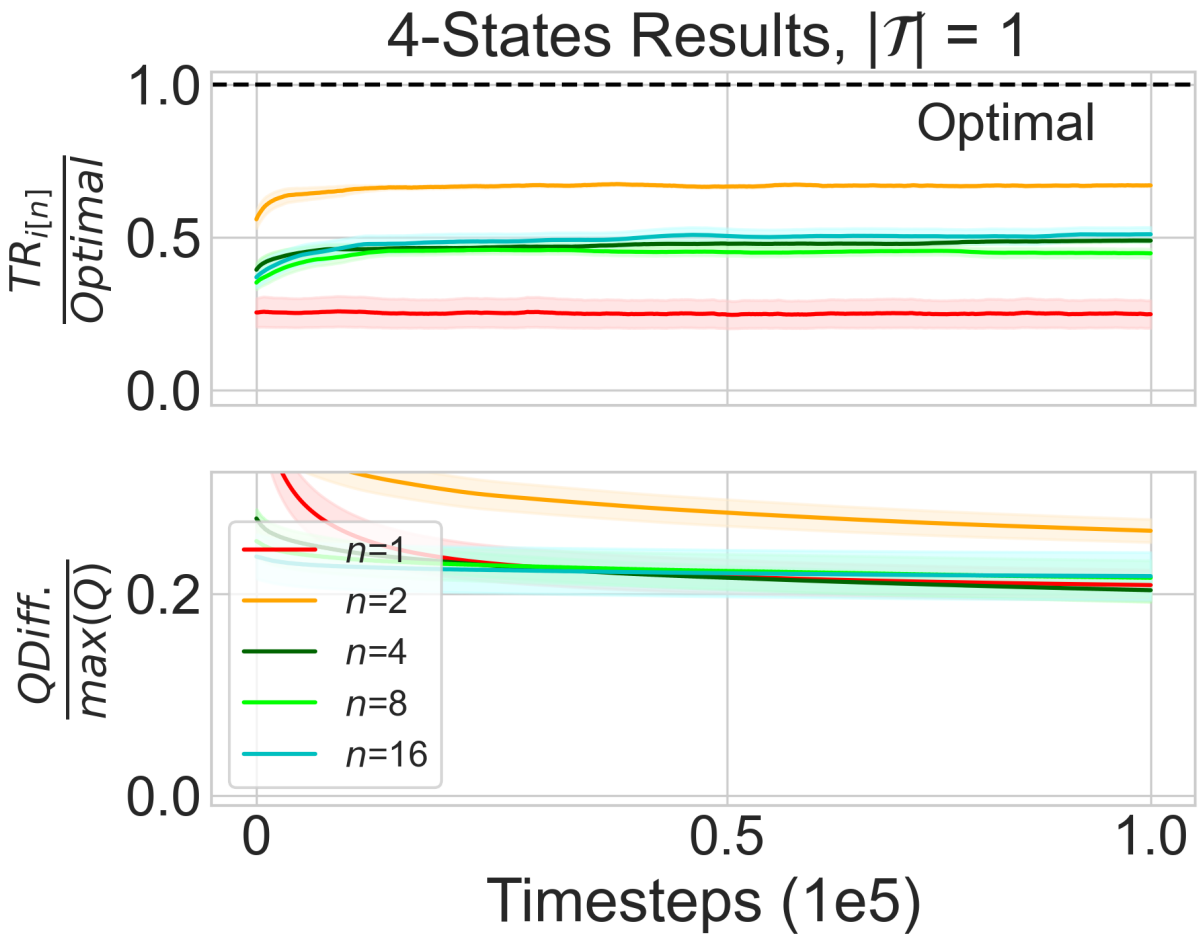

Figure 6.3: **4-States:** Team reward (top) and mean difference in Q-values normalized by maximum Q-value (bottom). We find that teammates are able to coordinate and achieve high team rewards and understand the value of actions when $n = 2$; however, large teams cause agents to struggle with coordination and agents have smaller differences between the expected value of their actions. This indicates that agents have not learned the value of particular actions as well in larger teams.

we measure team reward as a fraction of each team structure's theoretical optimal reward assuming no randomness (mean episode reward of $\frac{1}{2}$ for $n = 1$, $\frac{n-1}{n}$ for $n > 1$). Figure 6.3 (top) shows the team reward compared to optimal (y-axis) over timesteps of our experiments (x-axis). Each line represents a different team size with 95% confidence intervals. When $n = 1$, only 25.2% of the optimal reward is achieved. Increasing to $n = 2$ dramatically increases the reward to 66.5% of the optimal solution, and larger teams result in diminishing returns. Considering ϵ-exploration and stochastic transitions impose about 33% unintended actions and transitions, $n = 2$ performs well.

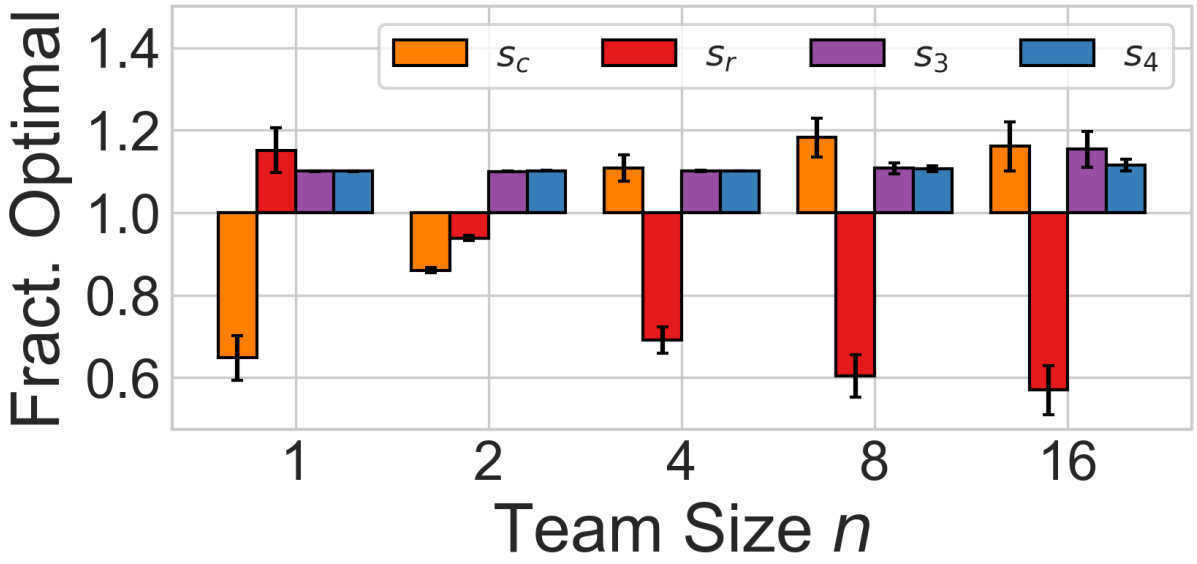

Figure 6.4: **4-States:** Mean state visitation fraction of optimal joint policy for different team sizes (95% confidence intervals). Positive bars indicate more visits to that state than the optimal strategy and negative bars indicate fewer. Teams when $n = 2$ perform closest to the optimal joint policy.

The y-axis of Figure 6.3 (bottom) shows the mean difference in Q-values between actions, scaled by the maximum Q-value in the table at each timestep. Lower values indicate agents expect similar values for any action and have not learned the reward dynamics of the environment. This plot follows the same trend as the reward: agent learn more disparate Q-values when $n = 2$, but larger teams cause agents to learn similar values for all actions. This indicates a decrease of environmental information as n grows.

Figure 6.4 shows the team state visitation frequencies as a fraction of the optimal policy with 95% confidence intervals (i.e., transitioning between s_c to s_r when $n = 1$, and one

agent in s_c while $n-1$ agents in s_r when $n > 1$). When $n = 1$, the agent fails to learn the value of transitioning to s_c. Agents perform closest to optimal when $n = 2$, suggesting they learn the value of visiting both s_c and s_r while avoiding s_3 and s_4. The agents are unable to fully converge to the optimal policy due to the stochastic transition function and ϵ-greedy action selection. With larger n, agents tend to visit s_c more often than optimal and s_r with less frequency, suggesting they fail to learn the reward-causing dynamics of the environment with larger groups, supporting our theory.

6.5.2 Iterated Prisoner's Dilemma (IPD) Results

We fix the cost $c = 1$, benefit $b = 5$, and define two teams ($|\mathcal{T}| = 2$) with increasing sizes of each team where $n = 1$ (no teams), $n = 2$ (one teammate), and then multiples of 5 to study general trends with larger teams. We fix $\nu = 97\%$ (non-teammates are 16 times more likely than teammates) and 100% when $n = 1$ (agents do not play themselves). Each experiment lasts 1.0×10^6 episodes where $N = 30$ agents learn using DQN [153], repeated for 20 trials each.

Figure 6.5 shows our results in the IPD environment for the mean population reward (top) and the difference in Q-values for C and D when paired with non-teammates (bottom). Both graphs share the same x-axis, representing the timesteps of our experiments.

Since mutual cooperation is the result with the highest mean population reward, we use reward as a proxy for learned cooperation (higher is better). When $n = 1$, agents converge to the Nash Equilibrium of mutual defection and obtain the lowest mean population reward. Consistent with Chapter 4, our results show how having even one teammate allows agents learn cooperation and achieve high mean population reward despite only being paired with this teammate 3% of the time. However, team growth has diminishing returns. When $n = 30$, the mean population reward approaches the mean reward of the environment, suggesting agents behave randomly (i.e., $\overline{R_i} = 2$ when cost is 1, benefit is 5). This is a direct example of Lemma 1.

The bottom graph shows how initially providing agents with teammates ($n = 2$) increases the difference in Q-values significantly since agents learn the benefit of mutual cooperation. Agents adapt this behavior towards other teams and the population experiences high cooperation and high reward. Further increasing team size tends to reduce the difference in Q-values until agents have little Q-value difference when $n = 30$, at which point agents behave essentially randomly.

As a further analysis into how teams impact learning, Figure 6.6 shows the mean maximum eigenvalue (λ_{max}) of agents' policy network Hessian matrices as they learn ($\log_{10}$

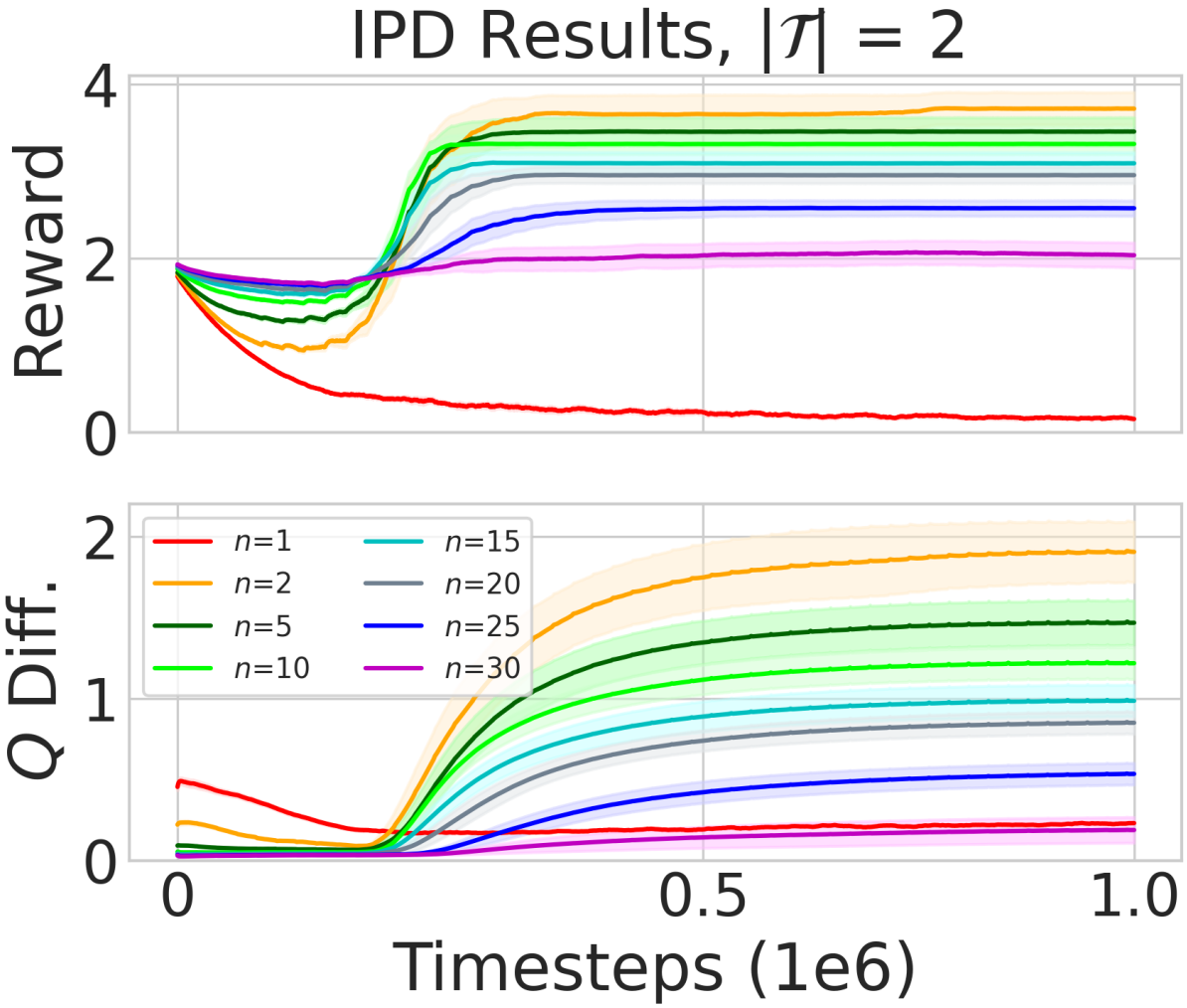

Figure 6.5: **IPD:** Mean population reward (top) and mean difference in agents' Q-values (bottom). We observe smaller differences between Q-values for cooperation and defection as agents are on larger teams, indicating agents have less preference for either action and behave randomly when $n = 30$.

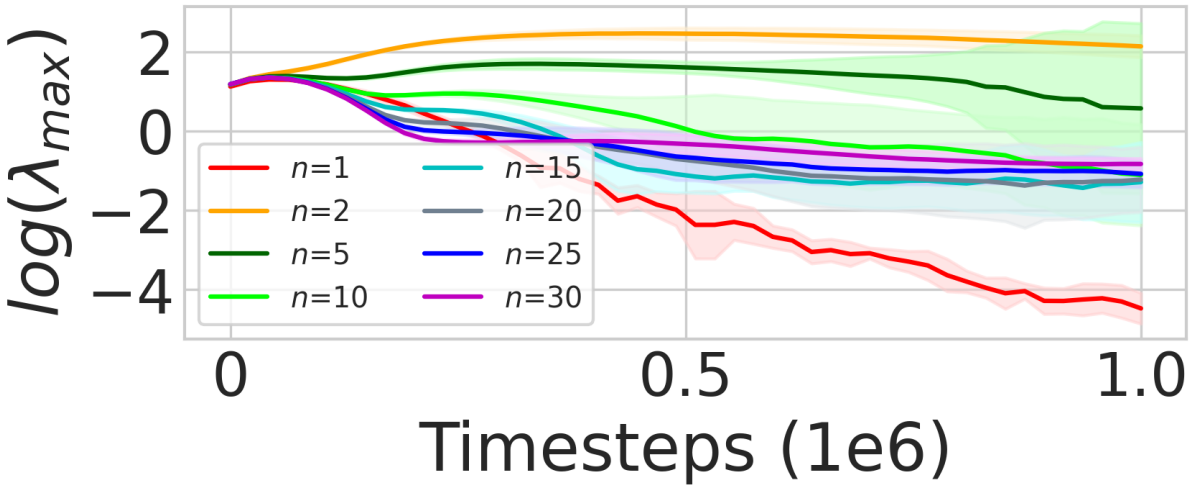

Figure 6.6: **IPD:** Mean maximum eigenvalue (λ_{max}) of agents' Hessian matrices. This represents the flatness of the loss landscape. We find that λ_{max} initially increases with teammates; however, large teams leads to a flattening of the loss landscape and agents learn random behavior when $n = 30$.

scale). Lower values of λ_{max} represent a flatter optimization surface [107] that makes convergence through stochastic gradient descent more difficult. When $n = 1$, the high rate of 0 reward leads to a flat optimization landscape, but when $n = 2$ or 5, λ_{max} is the highest among all team structures we study. As teams grow larger, the loss landscape flattens and convergence to a minima becomes more difficult. This highlights that teams shape the loss landscape to assist convergence to a cooperative minima [184], but large team structures flatten the landscape and reduce convergence abilities.

6.5.3 Cleanup Gridworld Game Results

We configure Cleanup with one team ($|\mathcal{T}| = 1$) and increase team size to remove impacts of other teams on the conditional reward structure. We implement PPO agents for 10 trials of 1.6×10^8 episodes (1,000 timesteps each) using the Rllib RL library.[1]

Figure 6.7 shows the team reward (top) and mean policy entropy (bottom) along the y-axes with 95% confidence intervals, and timesteps along the x-axis. We use **policy** entropy (π_i entropy) to better understand role specialization on teams, where lower π_i

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

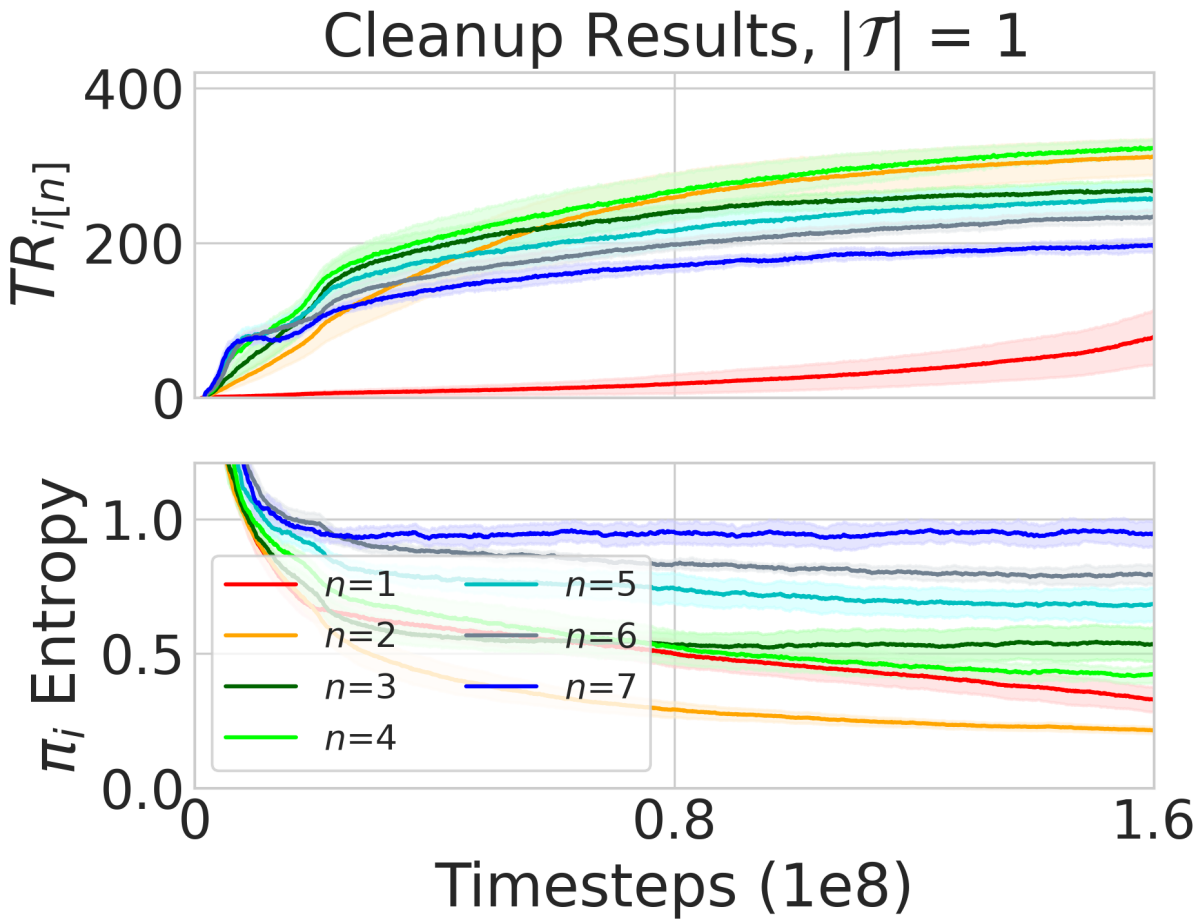

Figure 6.7: **Cleanup:** Team reward (top) and mean policy entropy (bottom) with 95% confidence intervals. We find that $n = 2$ and $n = 4$ achieve the highest team reward in Cleanup and $n = 2$ achieves the lowest π_i entropy. Larger teams lead to lower team reward and higher π_i entropy which indicates more random policies.

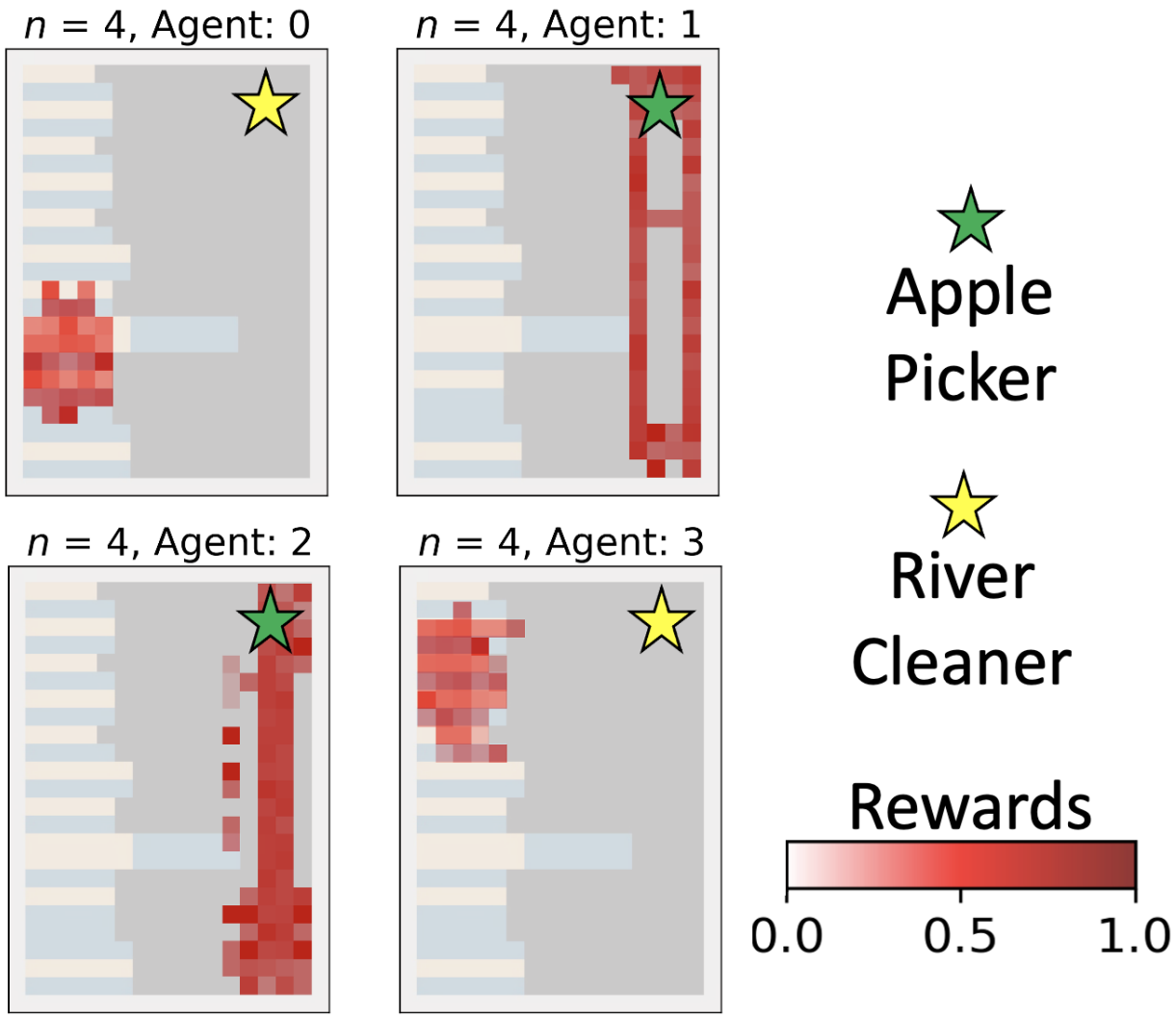

Figure 6.8: **Cleanup:** Team reward obtained at each location for different agents when $n = 4$. Green stars indicate agents that learn to pick apples and yellow stars indicate agents that learn to clean the river. We compare with Figure 6.9 when $n = 6$ to show that agents converge to specialized cleaning roles when $n = 4$.

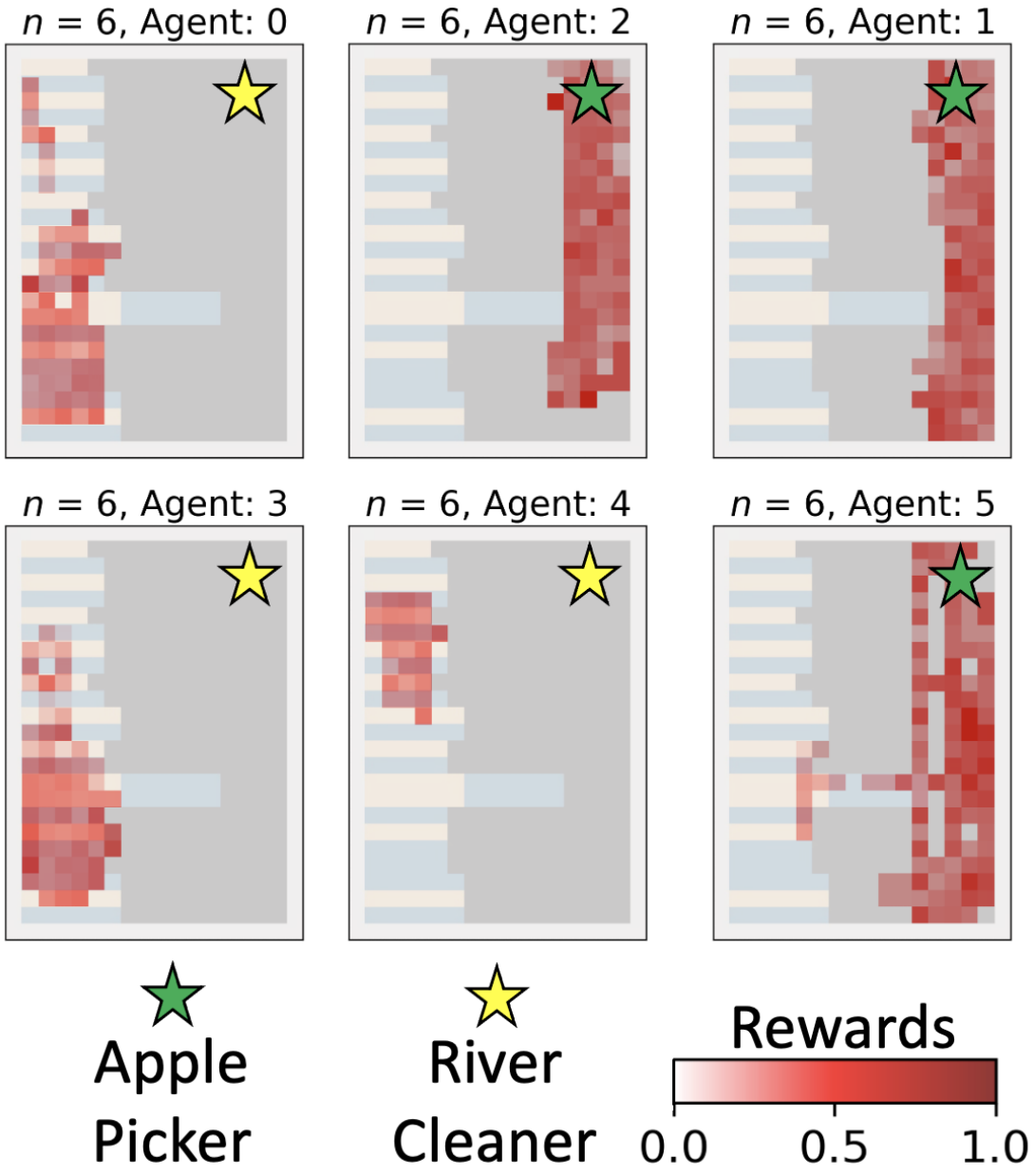

Figure 6.9: **Cleanup:** Team reward obtained at each location for different agents when $n = 6$. Green stars indicate agents that learn to pick apples whereas yellow stars indicate agents that learn to clean the river. Agents converge to overlapping redundant roles when $n = 6$ compared to specialized cleaning roles in Figure 6.8.

entropy implies higher role specialization and less random actions. Our results follow a similar trend as seen in 4-States and the IPD. More reward is initially obtained by adding teammates and is highest when $n = 2$ or $n = 4$, due to a division of labor: half of the agents specialize in each role of cleaning the river or picking apples. When $n = 3$, two agents specialize in river-cleaning roles while only one collects apples, causing slightly less team reward due to more sharing than when $n = 2$, but collecting fewer apples than when $n = 4$. Team structures with $n > 4$ tend to have decreasing team reward, following our theoretical findings in Section 6.4.3. We observe that when $n = 2$, mean π_i entropy is lowest and as n increases, agents policies tend to become more random. Our results indicate a correlation between team reward and agents' convergence to specialized roles, measured by lower π_i entropy, and find the lowest mean π_i entropy when $n = 2$.

Figures 6.8 and 6.9 show the mean team reward agents receive at different map locations when they are in a team of $n = 4$ (Figure 6.8) and $n = 6$ (Figures 6.9), where darker red indicates more reward. We indicate roles that agents converge to with colored stars in the top right corner of each plot: a yellow star for river cleaning agents and a green star for apple picking agents. When $n = 4$, we find that the two agents that specialize in river-cleaning roles (agent 0 and 3) also spatially divide the labor into different parts of the river, one in the top half and one in the bottom half. This allows their two other teammates (agents 1 and 2) to collect apples and reward for the team. However, when $n = 6$ we observe that three agents specialize in river-cleaning roles (agents 0, 3, and 4), but are less specialized in their cleaning locations. Agents 0 and 3 tend to clean the same segment of the river, converging to redundant policies that do not generate significantly more apples for their apple-picking teammates to collect.

6.5.4 Neural MMO Results

We implement PPO agents for six trials of 1.5×10^7 environmental timesteps (episodes are 1,000 timesteps each) using Rllib, similar to in Cleanup. Figure 6.10 shows the NMMO results. When $n = 1$, the agent fails to learn the value of collecting both food and water which results in no reward. As teammates are introduced, the agents learn complimentary harvesting roles and gain the highest team reward when $n = 2, 3, 4$. However, we observe diminishing returns with larger teams (when $n > 4$). We hypothesize that these values are highly correlated with the number of inventory item types and harvesting tasks. Similar to Cleanup, agents have less π_i entropy in settings where they achieve higher team reward, suggesting that agents in these teams have converged to specific roles and act less randomly than when they have zero or many teammates. This result supports our theory and is

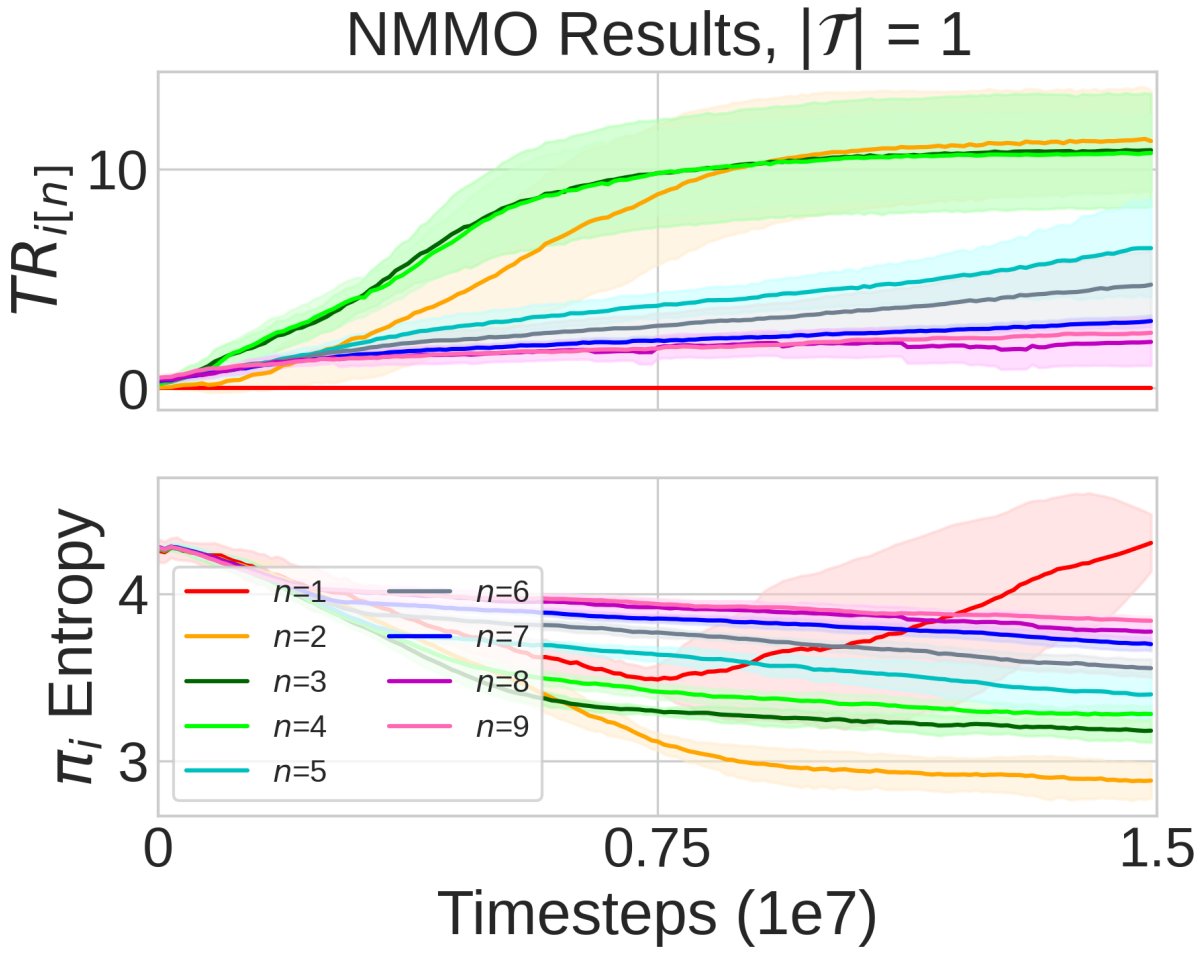

Figure 6.10: **NMMO:** Team reward (top) and mean policy entropy (bottom) with 95% confidence intervals.

consistent with our other experiments: a sufficient number of teammates results in more favorable policies, but too many teammates leads to diminishing returns.

Figure 6.11 shows the movement of agents when $n \in \{1, 2, 4, 5, 8, 9\}$. When $n = 1$ (Figure 6.11 top left), the agent has difficulty learning about the value of both food and water, resulting in the agent staying in the center region of the map where there is only grass and stone (Figure 3.3). When the agent is given a teammate ($n = 2$; Figure 6.11 top right), agents reliably converge to complimentary roles and explore different regions of the environment, collecting either food or water and sharing their resources. This behavior is also observed when $n = 4$ with two agents collecting food or water each (Figure 6.11 middle left). This joint policy generates one of the best team reward results in our evaluation, showing the benefits of initially adding teammates to discover reward-causing state-action pairs. When $n \in \{5, 8, 9\}$ in Figure 6.11, agents still learn complimentary roles; however, they tend to interfere with each other and cover similar areas of the environment instead of spatially dividing the gridworld to be more efficient. This result is consistent with our spatial results in Cleanup shown in Figure 6.9 where the cleaning agents clean the same area of the river when their team is larger. The NMMO environment is significantly large so that this duplication is avoidable. Despite this, agents have difficulty learning how to be spatially diverse and maximize the effectiveness of their joint policy. Furthermore, when $n = 8$ and $n = 9$, we observe some agents returning to the center grass/stone area later in a trajectory which contributes no positive reward for their team.

6.6 Self-Tuning Credo

Theorem 1 shows how there exist environments where teammates promote exploration and increase reward for executing reward-causing state-action pairs. However, team structures that maximize reward for reward-causing state-action pairs may lead to an (ϵ, μ)-information sparse stochastic game by Theorem 2. Designing favorable team structures for learning may be a difficult problem for researchers or practitioners because of domain variables. This may require trial and error and static configurations could be brittle to changes to the underlying environment dynamics. The credo model presents a unique opportunity where agents could recover stronger information signals themselves by modifying their credo parameters for various groups.

While we explored favorable team structures and credo parameters throughout Chapters 4 and 5, not all environments can be thoroughly evaluated in practice. The motivation for self-tuning credo is not to determine favorable team structures with fully team-focused agents as in Chapter 4. Instead, the goal is to allow agents to discover favorable learning

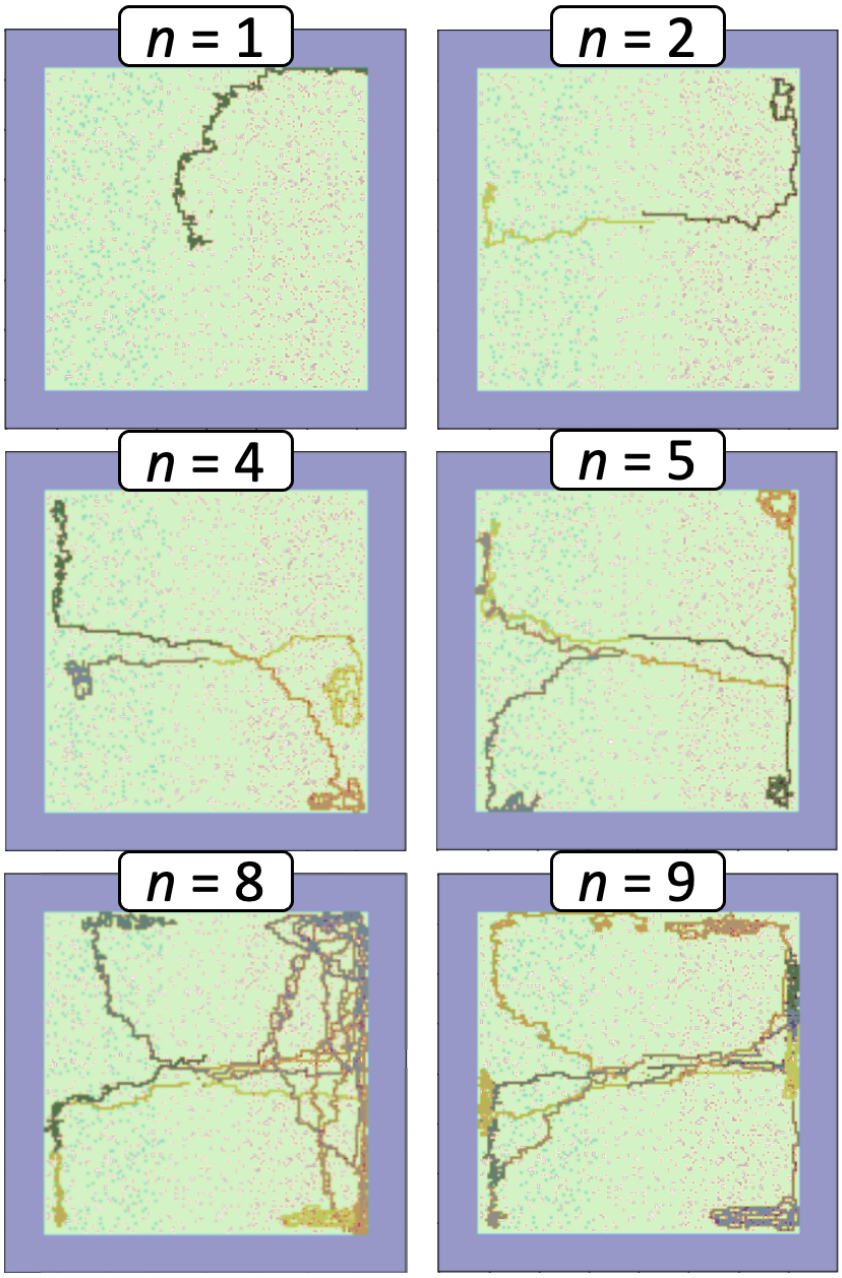

Figure 6.11: **NMMO:** Agent trajectories for different executions when $n \in \{1, 2, 4, 5, 8, 9\}$. Each different color represents the path of a different agent in the system. All agents are on the same team and fully share rewards.

conditions in *any* environment given a pre-defined team structure. Our hope is that this may make the design of team structures less critical, so that researchers or practitioners can define a team structure (that does not change) and agents will adapt their credo parameters to learn favorable policies in that setting. Agents' credo parameters for the given team structure may be sub-optimal when initialized; however, agents could discover favorable learning settings by modifying their credo parameters. One example is Figure 4.3 of Chapter 5, where a population of fully aligned system-focused agents achieve significantly less reward in Cleanup than if they were slightly self-focused. Agents initialized to be system-focused, but capable of modifying their credo parameters, might discover the benefits of being slightly self-focused themselves and discover a better joint policy. This would eliminate the need for researchers and practitioners to engineer specific credo parameters for different team structures in specific environments.

We propose and implement self-tuning credo agents to overcome sub-optimal learning settings like the fully aligned system-focused agents in Figure 4.3 of Chapter 5. We perform an initial evaluation using simple self-tuning credo agents to show how these agents can discover favorable joint policies despite sub-optimal initializations of team structures and credo parameter combinations.

6.6.1 Self-Tuning Credo Framework

We consider the same stochastic game base environment presented in Chapter 3 and used in previous chapters. Recall from Chapter 5 that agent i's credo is defined as a vector of parameters that sum to 1, represented $\mathbf{cr}_i = \langle \psi_i, \phi_i^{T_1}, \ldots, \phi_i^{T_{|\mathcal{T}_i|}}, \omega_i \rangle$, where:

- ψ is the credo parameter for i's individual reward IR_i,
- $\phi_i^{T_i}$ is the credo parameter for the reward $TR_i^{T_i}, \forall T_i \in \mathcal{T}_i$, and
- ω_i is the credo parameter for the reward i receives from the system SR_i.

Since self-tuning credo agents can have different credo parameters for the same group, we modify agent i's credo-based reward function $R_i^{\mathbf{cr}}$ so that all rewards collected by a team or the system are allocated in proportion to the credo values for those groups. Specifically, $R_i^{\mathbf{cr}}$ is calculated as:

$$R_i^{\mathbf{cr}} = \psi_i IR_i + \sum_{T_i \in \mathcal{T}_i} \frac{\phi_i^{T_i}}{\sum_{j \in T_i} \phi_j^{T_i}} TR_i^{T_i} + \frac{\omega_i}{\sum_{j \in N} \omega_j} SR_i. \tag{6.13}$$

Algorithm 1 Self-tuning credo algorithm

Require: $N \geq 1$, $E \geq 1$, $|\mathcal{T}| \geq 1$

$t \leftarrow 0$ ▷ Initialize time to 0.
Initialize $\mathbf{cr}_i \leftarrow \langle \psi_i, \phi_i, \omega_i \rangle, \forall i \in N$ ▷ Initialize credo parameters for each agent.
Initialize $\pi_i^{\mathbf{cr}}, \forall i \in N$ ▷ Initialize credo policy for each agent.
Initialize $\pi_i, \forall i \in N$ ▷ Initialize behavioral policy for each agent.
while $t < \infty$ **do**
    $\overline{\mathbf{R^E}} \leftarrow behavioral\_episodes(\boldsymbol{\pi}, \mathbf{cr}, E)$ ▷ Execute low-level policies for E episodes.
    for $i \in N$ **do**
        $V_i^{\mathbf{cr}}(\mathbf{cr}_i) \leftarrow V_i^{\mathbf{cr}}(\mathbf{cr}_i) + \gamma \overline{R_i^E}$ ▷ Update value estimate of $\mathbf{cr}_i$, where $\overline{R_i^E} \in \overline{\mathbf{R^E}}$.
        $\mathbf{cr}_i^{'} \leftarrow \pi_i^{\mathbf{cr}}(\mathbf{cr}_i)$ ▷ Define new credo for agent i, high-level action.
        $\mathbf{cr}_i \leftarrow \mathbf{cr}_i^{'}$ ▷ Update credo parameters.
    end for
end while

This is a necessary modification for the scenario where agents may have different credo parameters for the same group to ensure all rewards are re-allocated to agents in the population. To maintain consistency with Chapter 5, 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:

$$TR_i^{T_i} = \sum_{j \in T_i} \phi_j^{T_i} R_j(S, A_j, S), \tag{6.14}$$

$$SR_i = \sum_{j \in N} \omega_j R_j(S, A_j, S). \tag{6.15}$$

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 the situation where teammates may not have the same credo within a team.

Agent Architecture

The agent design is inspired by hierarchical reinforcement learning (HRL) and meta-learning. We draw specifically from *feudal* HRL where a single RL *manager* policy has a

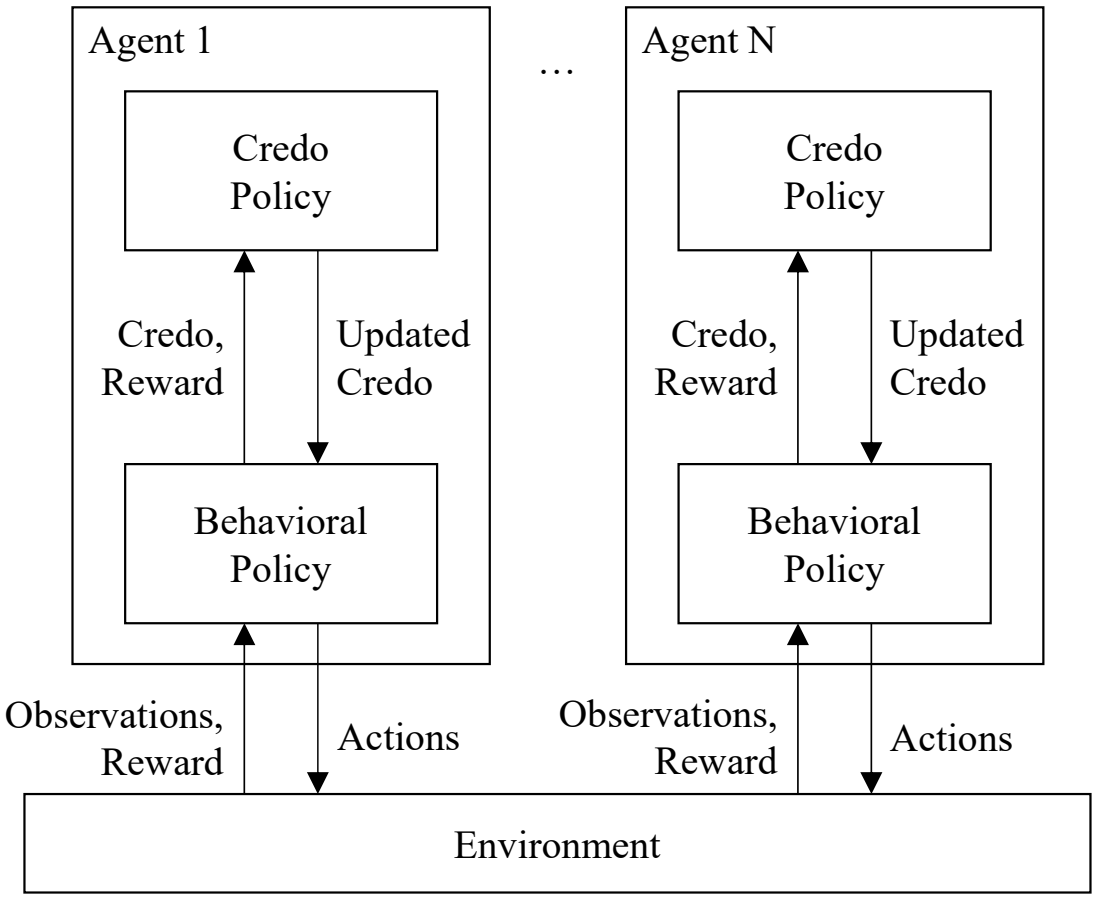

Figure 6.12: Overview of the proposed self-tuning credo agent framework. Each agent has two policies that operate at different time scales: a low-level behavioral policy that acts within an environment and a high-level credo-tuning policy that operates every $E \geq 1$ episodes. The credo-tuning policy shapes the optimization landscape for the behavioral policy while the learned behavior impacts the reward function for the credo-tuning policy.

single RL *sub-manager* policy that acts in the environment [43]. The manager sets a task (the credo policy sets credo parameters for the agent) that influence how the sub-manager (the behavioral policy in our case) learns in the environment [78]. Since credo parameters are connected to an agent's reward function, our framework can also be viewed as being similar to meta-learning [88]. Meta-learning algorithms aim to improve a learning algorithm by iteratively learning hyperparameters that result in improved learning conditions (i.e., "learning to learn") [238, 92].

An overview of our proposed agent framework and learning process is given in Figure 6.12 with N agents interacting with the same environment. Pseudocode is given in Algorithm 1, where $\boldsymbol{\pi}$ is the joint behavioral policy of all agents, $\mathbf{cr}$ represents the collection of credo parameters for all agents, $V_i^{\mathbf{cr}}$ is agent i's credo-facing value function, and $\overline{\mathbf{R}^{\mathbf{E}}}$ represents a list of all agents' mean rewards over E behavioral episodes (i.e., $\overline{R_i^E}$ is agent i's mean credo-based rewards over E behavioral episodes).

Throughout Chapters 4, 5, and 6, agents have only executed behavioral policies that learn in teams with static credo parameter settings (note that agents in Chapter 4 follow the full team-focus definition). In this setting, we define agents to have two internal policies that operate at different time scales: a "low-level" behavioral policy and a "high-level" credo policy. 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 (in line 6 of Algorithm 1). At each timestep of an episode, rewards are shared with other agents according to the agent's credo parameters $\mathbf{cr}_i$, similar to Chapter 5. The high-level policy, $\pi_i^{\mathbf{cr}}$, modifies the agent's credo parameters at a slower time scale. For value-based versions of $\pi_i^{\mathbf{cr}}$, as used in our implementation, the high-level policy maintains a value function for different combinations of credo parameters using the low-level policy's mean reward over E behavioral episodes, $\overline{\mathbf{R}^{\mathbf{E}}}$ (line 8 in Algorithm 1). Conditioned on the previous credo parameters, $\mathbf{cr}_i$, the high-level agent produces updated credo parameters $\mathbf{cr}_i'$ to shape i's reward function in the following E behavioral episodes (lines 9 and 10 in Algorithm 1). The high-level credo policy operates at a larger time scale than the low-level behavioral policy to allow the low-level policy to gain experience with a particular credo setting 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 or models the other (i.e., both are individual learning policies). Instead, each policy influences the optimization landscape and learning problem of the other without their actions being directly observed in their state spaces. For example,

1. **Low-Level Influence on High-Level:** the low-level policy produces rewards for

the high-level policy; if the low-level policy fails to obtain high reward, the high-level credo-tuning policy fails to get positive feedback.

2. **High-Level Influence on Low-Level:** the credo parameter changes made by the high-level policy shapes the reward function of the low-level policy for the next set of E episodes which impacts the behaviors that the low-level policy learns. The low-level policy does not observe the change in credo parameters, or what those credo parameters are; however, their received reward is shaped by the parameters defined by the high-level policy.

Tuning the amount of shared reward within groups regulates the amount of reward agents receive for executing reward-causing state-action pairs (Theorem 1) and information in their reward signals (Theorem 2). 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 (themselves, any teams they may belong to, and the system).

Implementation

Low-level Behavioral Policy: The Proximal Policy Optimization (PPO) [214] implementation in Chapter 5 used an older version of the RLlib[2] library (version 0.8.5) that made interrupting agents' behavioral training loops every E episodes for the high-level credo-tuning policy infeasible. Thus, we adopt the same architecture as the agents in Chapter 5 to an updated version of RLlib (version 2.1.0) to incorporate the self-tuning credo agent architecture shown in Figure 6.12.

High-level Credo Policy: Any type of RL algorithm can be used for the high-level credo policy. To reduce sample complexity, we implement the high-level credo policy as a Q-Learning agent with ϵ-greedy exploration ($\epsilon = 20\%$) [257]. Consistent with Chapter 5, agents belong to only one team, resulting in credo vectors with three parameters (i.e., $\mathbf{cr}_i = \langle \psi_i, \phi_i, \omega_i \rangle$). We limit possible agent credo values to intervals of 0.2, creating 21 possible credo configurations for the high-level policy to observe (shown in Figure 5.11 of Chapter 5). With three credo parameters, the agent can take actions to increase any credo parameter by 0.2, decrease any other credo parameter by 0.2, or not change the credo parameters. This results in seven discrete actions. 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

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

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 (such as choosing to further increase system-focus), 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. The value of E is defined by the number of concurrent behavioral training environments discussed in our evaluation.

6.6.2 Empirical Evaluation

We perform a preliminary evaluation with a population of self-tuning credo agents in the Cleanup environment. We choose the Cleanup environment since we have extensively explored the results of various joint policies throughout the previous chapters of this dissertation. To be consistent with Chapter 5, we instantiate six agents and construct three teams of two agents each (i.e., $|\mathcal{T}| = 3$, $|T_i| = 2$).

Experiment Methodology

We design an experiment to evaluate if self-tuning credo agents can overcome a sub-optimal initialization of fully system-focused credo parameters and discover a joint policy that achieves higher mean population rewards (Scenario 1 or 2 in Figure 5.11 in Chapter 5). We initialize agents to be fully system-focused (i.e., $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$). Note that this configuration represents a fully cooperative population; however, agents are able to modify their credo parameters towards a smaller team (through the team-focused credo parameter). The low-level behavioral policy trains on data from every 96 episodes since our Rllib implementation has six parallel workers with 16 environments each. The number of workers and their environments are learning hyperparameters that are consistent with Chapters 4, 5, and 6. Thus, the high-level credo policy takes an action to update the agents' credo parameters every $E = 96$ episodes. The behavioral policy never observes the credo parameters but instead experiences changes to their rewards.

This experiment is equivalent to initializing agents with credo parameters in the bottom left corner of Figure 5.11; however, agents' credo policies are now able to adjust the agent's credo parameters. The design of this experiment is used to determine if self-tuning credo agents can discover stronger reward signals and converge to a better joint policy than the fully system-focused population.

We compare the population of credo tuning agents to two populations where credo parameters remain static throughout entire experiments. In the static team-focus experiment, agents maintain $\mathbf{cr}_i = \langle 0.0, 1.0, 0.0 \rangle$ for the entire experiment. In the static system-focus

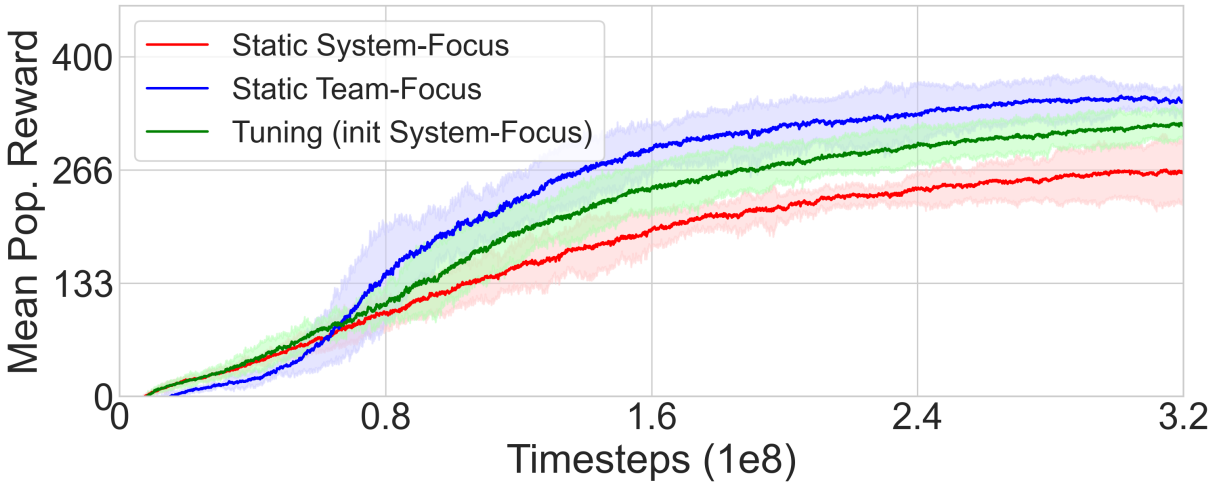

Figure 6.13: **Cleanup:** Mean population reward over time for each experiment in our evaluation. Results are the mean across 4 trials for each experiment with 95% confidence intervals. The static team-focused agents have been observed to achieve the highest mean population reward in Cleanup among different credos (Figure 5.11, Scenario 2). This shows that self-tuning credo agents that are initialized with system-focused credo can increase their mean population reward above the initial fully system-focused settings.

experiment, agents maintain $\mathbf{cr}_i = \langle 0.0, 0.0, 1.0 \rangle$ for the entire experiment. We execute four trials of each experiment configuration to measure variability.

6.6.3 Preliminary Results

This section presents preliminary results from the experiments described in Section 6.6.2. We observe the same patterns with the static experiments as in Chapters 4 and 5: teams that are fully team-focused perform significantly better than when agents are fully system-focused. Consistent with previous results, fully team-focused populations converge to the best observed global joint policy of two river cleaning agents and four apple picking agents. The fully system-focused population converged to sub-optimal joint policies of either three river cleaning agents and three apple picking agents or four river cleaning agents and two apple picking agents (depending on the random seed).

We found that updating the PPO agents from RLlib 0.8.5 to RLlib 2.1.0 modified their learning processes so that agents learn more gradually (despite no changes to the algorithm configurations). Thus, while our direct learning curves are not comparable to

the Cleanup results in Chapters 4, 5, and 6, teams achieving significantly more reward than the fully cooperative system is consistent and we extend the duration of the experiments from 1.6×10^8 to 3.2×10^8 environment steps.

Mean Population Reward

Figure 6.13 shows the mean population reward with 95% confidence intervals over the four trials of each experiment: static system-focus, static team-focus, and self-tuning credo 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 Chapters 4 and 5, we find that static agents that are fully team-focused (blue) perform significantly better than static system-focused agents (red). The static system-focused agents are equivalent to a fully cooperative system since all agents in the system fully share their rewards. Higher reward is achieved by 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 5.11 that agents with full team-focused credo is one setting that achieves the highest observed reward in this configuration. The expectation of these initial experiments with self-tuning credo agents is not to outperform the fully team-focused credo since those agents achieve the highest observed mean population reward in our Chapter 5 evaluations. Instead, the hope is that self-tuning credo agents learn and perform better than their initialized settings (i.e., fully system-focused credo; the red line) towards the level of the static team-focused credo setting. The green line in Figure 6.13 shows the mean population reward for the self-tuning credo agents initialized with fully system-focused credos with 95% confidence intervals. Through the first 800,000 timesteps of the experiment, the credo-tuning agents (green) learn along approximately 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 fully system-focused credo population by the end of the experiment (320 reward for credo-tuning agents compared to 264 reward for static system-focused agents). This shows that self-tuning credo agents are able to achieve more mean population reward than the fully system-focused setting despite their sub-optimal initialization.

Division of Labor in Global Joint Policy

We now examine the credo-tuning experiment results in more detail. Figure 6.14 shows the amount of apples consumed (top) and cleaning beam actions (bottom) by each credo-

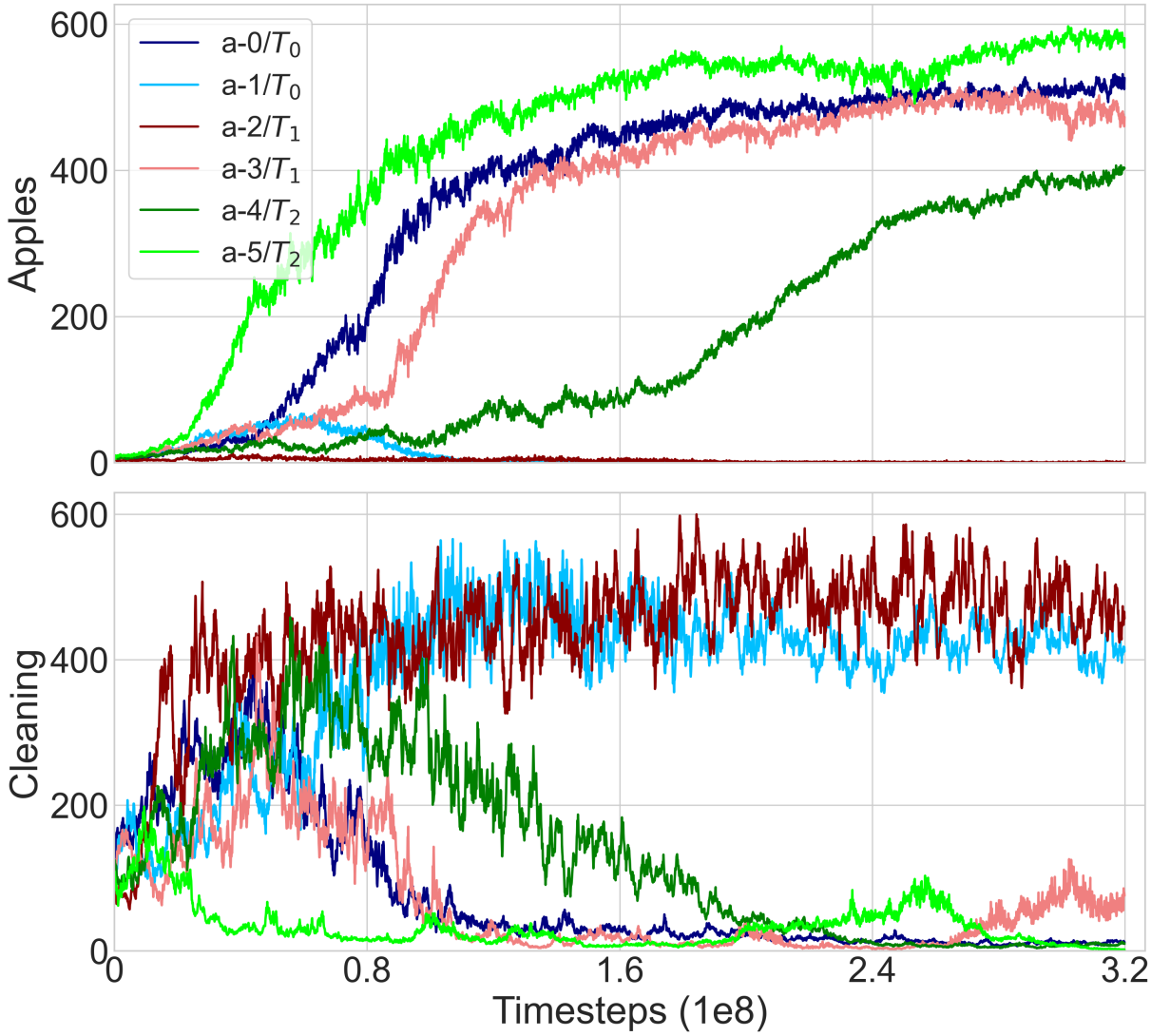

Figure 6.14: **Cleanup:** Amount of apples consumed (top) and cleaning beam actions (bottom) by each agent for one trial of the credo-tuning experiments with agents initialized with system-focused credo (green line in Figure 6.13). Agents are labeled so that "a-0/T_0" is agent #0 on team #0. Teammates 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 autonomously discover the better joint policy of four apple pickers and two cleaning agents autonomously (which is the same as fully team-focused agents) and generate more reward (Figure 6.13).

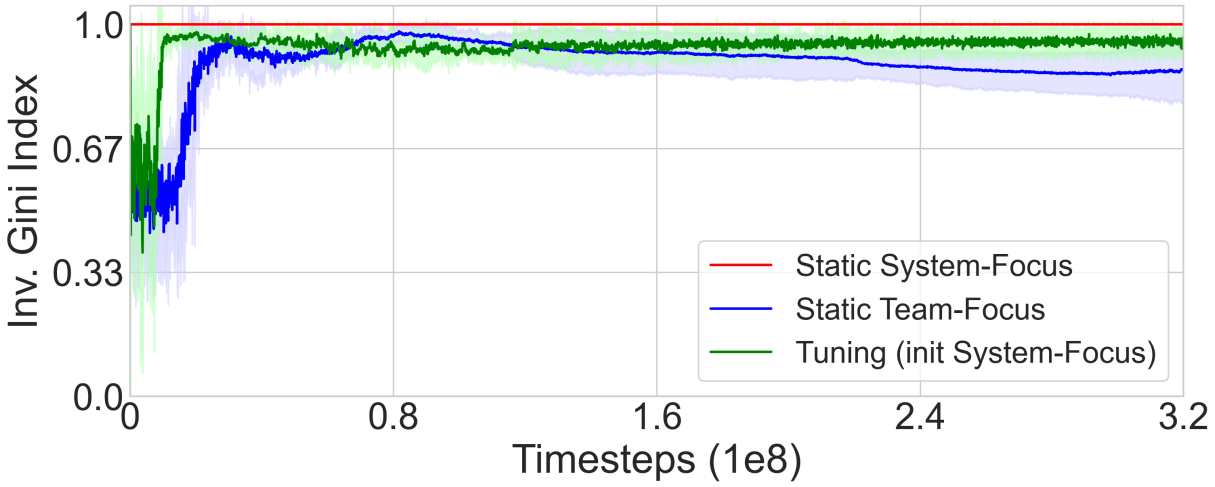

Figure 6.15: **Cleanup:** Inverse Gini index curve for each experiment in our evaluation. Results are the mean across 4 trials for each experiment reported with 95% confidence intervals. Static system-focused credo is defined to have full equality and is always 1. This shows that credo-tuning agents achieve slightly higher equality than the static team-focused agents.

tuning agent in one trial where the agents are initialized to be fully system-focused (the green line in Figure 6.13). Despite being initialized as fully system-focused, these agents are members of one of three teams (T_0, T_1, or T_2) that are one of their modifiable credo parameters. 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 some fully system-focused trials in Chapter 5, the agents in the credo-tuning experiment initially specialize into roles of three apple pickers 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. We analyze the credo parameters of these agents in a later subsection. This discovers the global joint policy of four apple picker agents and two river cleaning agents (joint policy of the static, fully team-focused agents) despite agents being initialized with fully system-focused credo. This results in an increase in mean population reward when compared with the static fully system-focused scenario. While the mean population reward level of fully team-focused agents is not quite reached, these agents appear to mostly discover the same global joint policy as the fully team-focused agents of two river cleaning agents and four apple picking agents by the end of the experiment. Perhaps longer training would see convergence to

the reward level of the fully team-focused population (blue in Figure 6.13) given this joint policy.

Population 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 reward equality to examine if tuning credo leads to significant inequality among the population. We model population reward equality as the inverse Gini index, consistent with past work [145] and the previous chapters:

$$Equality = 1 - \frac{\sum_{i=0}^{N} \sum_{j=0}^{N} |R_i^{\mathbf{cr}} - R_j^{\mathbf{cr}}|}{2N^2 \overline{R^{\mathbf{cr}}}}, \tag{6.16}$$

where values closer to 1 represent more equality. Figure 6.15 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-focused 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 initialization and is worthy of further exploration, we find that credo-tuning agents discover a setting that achieves high reward while maintaining high reward equality across the population.

Dynamic Credo Parameters of Each Agent

Figure 6.16 shows how the credo parameters change over time for each agent in the trial shown in Figure 6.14. Each plot is titled and colored according to the agent's label and color in Figure 6.14. The y-axis of each plot shows the credo parameter values and the x-axis of each plot shows timesteps of the experiment. The values shown for each credo parameter are the mean sliding window of 10 samples (increments of one); thus, some results appear between two discrete credo steps (such as 0.1 being between 0.0 and 0.2).

Figure 6.16 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

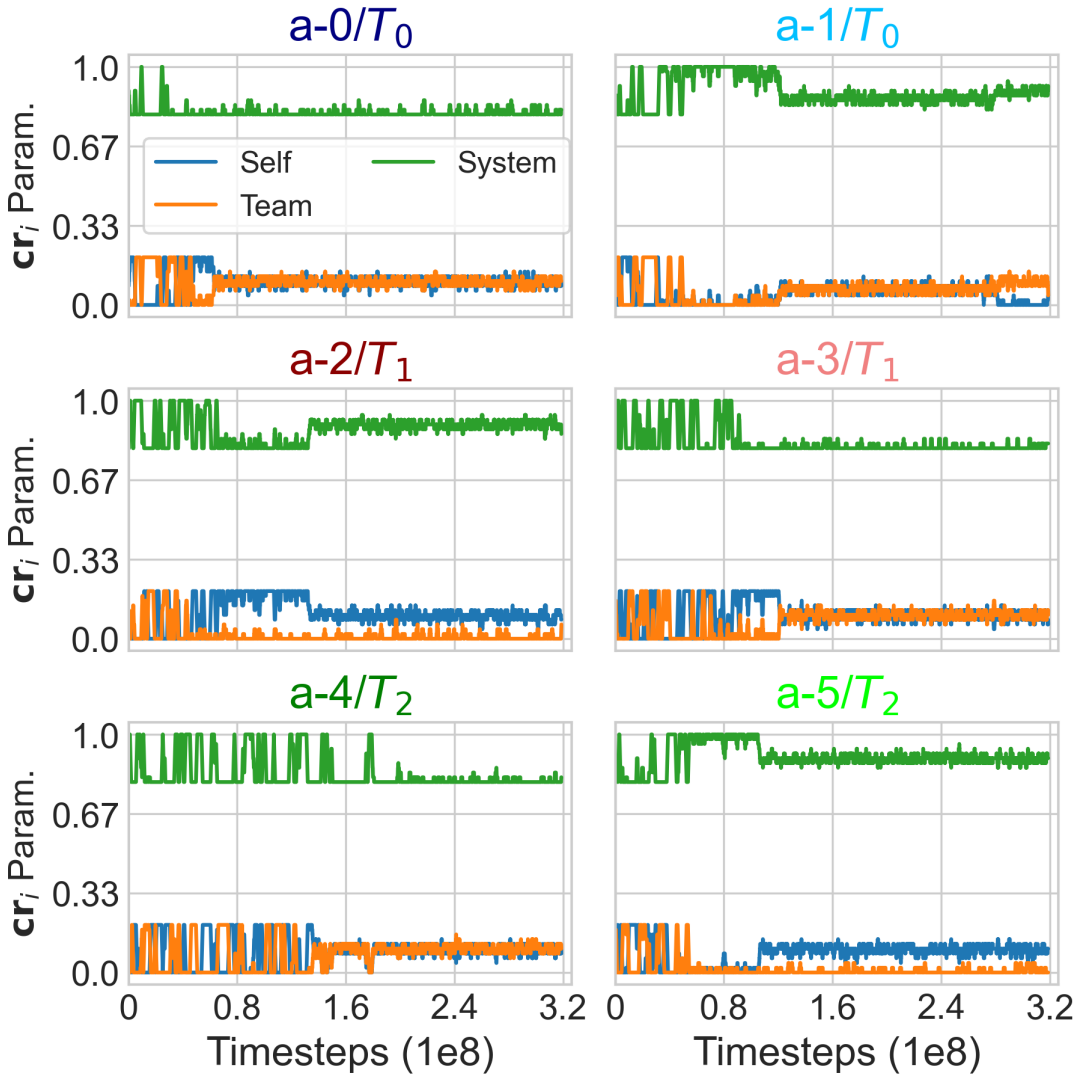

Figure 6.16: **Cleanup:** Credos of all six agents over time in the same credo-tuning trial as Figure 6.14. Each plot shows the credo parameters for a different agent shown in Figure 6.14. Each y-axis represents credo parameter space and each x-axis represents timesteps. We observe that heterogeneous 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.

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 that 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 discovering a globally better joint policy while maintaining high equality.

6.7 Discussion

This chapter provides an understanding as to why, and under which conditions, smaller teams can outperform larger teams in certain environments. Introducing teammates can help agents identify reward-causing state-action pairs (Section 6.3), but too many teammates can make credit assignment more difficult which hinders learning (Section 6.4). This provides theoretical explanations behind the empirical results of some other recent research [55, 184].

A common perception about RL theory is that convergence to the optimal policy is guaranteed given infinite computation. While this finding is true for single-agent RL [230], convergence guarantees are known to not hold in many multiagent settings [37]. This chapter's context of multiagent teams, even in a scenario with one fully team-focused team (i.e., a cooperative population), is a setting where convergence to an optimal joint policy **is not** guaranteed, even with infinite computation. We show this through Theorem 2, since information converges to zero as a function of team size. However, information does not need to be zero for RL to fail (it can fail when information is sufficiently small); thus, in practice, infinitely large team size is not required for RL agents to fail to learn. We are unable to guarantee non-convergence since random policy updates could *theoretically* result in the optimal joint policy; however convergence to this policy is not guaranteed.

To recover stronger learning signals in scenarios with sub-optimal team structures and credo parameters, in this chapter we propose, design, and implement a self-tuning credo

agent capable of dynamically changing its internal credo parameters. Our preliminary results show how individual self-tuning credo agents can recover the best observed joint policy in Cleanup of four apple picking agents and two river cleaning agents despite being initialized as fully system-focused. The motivation for self-tuning credo is to generalize to environments where the best joint policy or team structure is unknown and experimentation may be costly. In those settings, the goal of self-tuning credo agents is that they could autonomously discover favorable credo parameter distributions that discovers good joint policies for a given team structure and environment.

While we provide insights into the importance of teams and team structure to shape learning problems and reward functions for individual learning agents, there are several opportunities for future work specifically related to the work in this chapter. These include precisely measuring ϵ and μ from domain variables or experimenting with the impact of alternate definitions for teams or reward functions. Other direct future work includes analyzing self-tuning credo agents in settings with other initial credo parameter settings or instances where agents are members of multiple teams.

6.8 Conclusions

This chapter contributes theoretical underpinnings toward understanding how teams shape the learning environments for individual agents to achieve the results we observe in Chapters 4 and 5. The development of this work is influenced by the fields of organizational psychology (OP) and anthropology. Our result in Theorem 1 echoes findings of division of labor and complimentary traits found in studies of collective intelligence and the OP subfield of multiteam systems [143], whereas Theorem 2 parallels the concepts behind the maximum number of social connections that can be maintained [54]. This chapter formalizes these concepts in terms of information theory and defines a multiagent setting where learning is not possible, an (ϵ, μ)-information sparse stochastic game. The culmination of the previous three chapters motivates the idea of self-tuning credo – a system where individual agents modify their group alignment to receive benefits of reward sharing and recovering stronger feedback signals when necessary (i.e., when in sub-optimal groups such as a fully cooperative system in Cleanup). Since optimal team structures are highly domain dependent and may be unknown in various settings, self-tuning credo agents may have the potential to learn the credo parameters that achieve high rewards in any setting. This mitigates the burden of researchers or practitioners having to determine team structures of fully team-focused agents or specific credo parameters in settings where favorable configurations may be unknown.

Chapter 7

Conclusions and Future Work

This chapter presents an overview of our contributions in this dissertation. We expand on various discussion points and provide avenues for future work.

7.1 Summary of Contributions

The thesis statement of this dissertation is that teams can have significant advantages in guiding the development of policies for individual agents that learn from experience.

In the previous chapters, we found that teams have a significant impact on the behavior that AI agents learn and identified settings where teams are specifically beneficial for policy development in challenging domains. First, we explored the impact of team structures on how agents learn. We developed a model of multiagent teams and explored how team structures impact game-theoretic incentives of interaction. With learning agents, we found that agents in teams developed pro-social policies with agents in separate teams under certain conditions, leading to high global rewards despite game-theoretic incentives that encouraged defection. In a gridworld game, we found that team structures of multiple teams promote role specialization and the development of a more efficient global joint policy that achieved the highest observed global rewards. Agents learn role specialization from only their team-based shared reward signal and multiple teams coordinate to converge to efficient global joint policies (i.e., global distributions of roles). Populations with multiple teams achieve this high reward while maintaining high global equality. These results provide an initial insight as to how teams can shape the policies that agents develop under various assumptions of group alignment and team structure.

Second, we relax the assumption that teammates are fully aligned with the goals of their team and explore the impact of teams with various degrees of mixed incentives. We present credo, a model that defines how agents optimize their behavior for various groups they belong to: themselves, any teams they may belong to, and the entire system. We find that teams of learning agents that are fully team-focused autonomously learn to cooperate with other team-focused agents (on other teams) and learn to not be exploited by a small amount of self-focused agents in certain conditions. When all agents in the population have the same credo, we find that agents with high team-focus are more robust to some amount of self-focus than when agents are more system-focused. In the gridworld environment, agents learn efficient role specialization in multiple scenarios, not just when they are fully team-focused. First, highly system-focused agents generate significantly more reward if they are also slightly self-focused compared to if they are fully system-focused. Second, highly team-focused agents generate high reward regardless of whether they are slightly self-focused or not. These scenarios achieve significantly more global reward than the setting previously assumed to achieve the highest reward in mixed-motive environments (i.e., a fully cooperative population). These results motivate our next contribution.

Our last area of contribution is centered around understanding how teams and certain credo parameters lead to some populations that discover efficient joint policies, while others reliably hinder the emergence of these policies. We provide theoretical underpinnings that draw connections between team structure, reward signals, and the policies that agents develop in specific environments. We expand a single-agent concept of information sparsity to the multiagent setting. Our theory shows how sub-optimal team structures can turn a stochastic game into a setting where agents are not able to reliably learn high performing policies (although these could theoretically still be discovered through random updates). Our empirical results align with our theory across four different environments with various learning algorithms, including RL and deep RL that are either on-policy or off-policy and value-based or policy gradient.

7.1.1 Tuning Credo to Improve Learning

Our results motivate a deeper question about the impact that team structure can have on multiagent learning settings, and the role credo can play to mitigate potential risks. The environments we explored in this dissertation are testbeds that represent the underlying dynamics of broader classes of problems. While we can observe agents' joint policies and determine which joint policy performs best, there likely exist environments where the best joint policy of a population is unknown or may change over time. The idea that team structures can help agents develop effective joint policies, but can also limit

their development if teams are sub-optimally defined, is a powerful concept that can be utilized for multiagent learning. In settings where the best team structure for learning is unknown, credo may be a way to enable agents to autonomously discover favorable joint policies by including different types of reward signals. Allowing agents the ability to self-regulate their internal credo parameters influences the amount of information they have to learn from; thus, tuning credo shows promise in giving learning agents the ability to overcome sub-optimal team structures. For example, agents in a fully cooperative system that is challenging to learn in could become slightly self- or team-focused to gain more information from their actions.

We briefly explore this direction of research and design an agent inspired by feudal hierarchical RL [43, 78] and meta-learning [238, 88, 92]. In our setting, agents have one credo-tuning policy (i.e., manager) that modifies the agent's credo parameters. The credo parameters shape the reward received by the agent's behavioral policy (i.e., sub-manager) in the environment. This one-to-one configuration is inspired by feudal HRL [43] and the high-level learning credo-tuning policy shaping the reward function for the low-level behavioral policy resembles a meta-learning problem [88]. The overall goal of this research is to adapt the benefits of team structures to any environment and eliminate the need for researchers and practitioners to engineer specific team structures and credo parameters in new or dynamic environments. Self-tuning credo agents have the potential to learn the credo parameters in any team structure and environment that benefit their learning processes and achieve high rewards.

Our work contains contributions in Game Theory, MAS, and MARL; however, we believe the largest contribution of this dissertation is highlighting how population structure has an impact on the policies that individual learning agents develop. In future environments with individual learning agents, we hope our work encourages researchers and engineers to consider population structure when designing and implementing systems with multiple individual learning agents.

7.2 Revisiting Motivating Examples

This thesis can be illustrated by two real-world examples: wildland fire fuel mitigation and invasion game team sports analytics.

First, studies have shown that wildland fire fuel mitigation has been insufficient to prevent major wildfire disasters despite 98% of properties adhering to California's defensible space code [164]. The defensible space code focuses on characteristics of a single property

instead of a broader, community-wide level. Furthermore, monitoring and policing fuel mitigation is a costly undertaking, despite recent efforts to make this process easier [190]. Mitigating wildfire fuel, despite the low-risk of being caught for non-compliance or for the good of one's neighbors, creates a social dilemma among neighboring properties. To solve the dilemma, homeowners would need to pay the cost of removing more fuel than legally required. In this first motivating example, we argue that a community-based (i.e., team-based) solution to fuel mitigation would modify the incentives and goals of a community to properly adhere to a community level defensible space code. Stricter legal requirements are not possible since fuel mitigation extends beyond property lines and property owners are not able to modify other people's property, even to improve their own safety. Property owners that pay the cost of removing more fuel on their own individual property to comply with their community approach (i.e., more fuel than would be required if they only considered their own property and own house) could receive the benefits of their neighbors doing the same on their individual properties. Thus, the costs and benefits are not purely monetary, but ingrained in the overall reduced risk of wildfire. Reinforcement learning (RL) could be used to identify effective team structures among land owners across various geographical distributions that result in the best mitigation practices. This would result in significantly better fuel mitigation and overcome the current social dilemmas humans fail to solve.

Second, while sports analytics has revolutionized "striking games" such as baseball, they have lagged behind in sports classified as "invasion games" (football/soccer, ice hockey, and basketball). The reason for this is that invasion games have more opportunities of teamwork, where multiagent interaction must be modeled to properly capture the dynamics of the sport. Furthermore, invasion game sports teams are often composed of disjoint groupings of player positions (e.g., forwards, defense, and goalies in ice hockey), each with potentially different playing styles or goals. For team sports analytics to fully expand to invasion games, models must identify and understand various role specializations among a team similar to how we have identified role specialization with learning agents across various domains. In this dissertation, we have shown the emergence of role specialization and successful teams using a variety of team structures and combinations of credo parameters. There are several areas where the work in this dissertation can make a direct impact. Modelling the distribution of roles and team structure within opponent teams can help construct or learn a best response strategy that could provide significant value for coaching. Identifying role distributions could be done using behavior cloning (supervised learning) or inverse reinforcement learning to learn players' or groups' value functions. Similar techniques could be used to analyze a team's current roster and identify sub-optimal distributions of roles and areas in the roster for improvement to benefit team management.

This may inform the use of monetary incentives at the individual or group level to modify credo towards settings more favorable for the team.

7.3 Broader Implications and Ethical Considerations

As with any technology, it is important to understand the risks and broader implications. The work in this dissertation is no exception.

The first example is that team structures can have negative implications if they are poorly defined. For instance, the team structure will promote inequality in settings where the most efficient joint policy relies on one team learning to free ride or benefit from the reward of other teams without contributing itself.

The second example is that defining explicit team structures among a population can exclude some agents from specific teams. Defining teams as a system designer may ignore potential the desires of agents for specific groups. While both of these examples could be mitigated with credo or self-tuning credo agents under certain conditions, reducing group alignment to a series of parameters can promote potentially unethical scenarios. In a real world scenario, if an institution developed the ability to predict or infer the credo parameters of agents or people, they could penalize or act unethically to reduce free will among the individuals. This is similar to existing concerns about the potential misuse of other AI technologies such as facial recognition [101].

Lastly, our work assumes the *ability* for system designers to impose or change team structures in environments. This ignores alternative underlying incentives for various team structures. We may fail to capture additional characteristics of team structures by modifying the structure of teams or incentivizing changes to agents' credo parameters based on productivity, equality, or any other measure of performance. In practice, all features of an environment and the long term impact of team structures must be considered. All of these points are interesting areas for important further research throughout the development of teams, credo, and Cooperative AI.

7.4 Similarities in the Natural World

In this section, we broadly connect the work in this dissertation to observations in the natural world. We identify connections from the level of human relationships to intercellular biological connections.

7.4.1 Human Level

Teams and working together are the underlying concepts behind many of humanity's greatest accomplishments. As noted in Chapter 2, organizational psychology and multiteam systems have explored features of successful teams and groups in the workplace, sports, and the military for several decades. However, we also highlight comparisons of our work with features behind Dunbar's number [54]. The premise behind Dunbar's number is that mean group size and the strength of connections within a group are highly correlated with that species' cognitive limits. In the context of primates, this is defined as the maximum number of individuals with whom an one can maintain social relationships through personal contact. Dunbar analyzes various hunter-gatherer and Western working societies for impact of group size, in much the same way our research analyzes team structure and size with RL agents. Dunbar draws correlations between group structures and the development of language in humans. The study also finds a marked negative affect of group size on group cohesion and job satisfaction when groups were too large or poorly defined. Interestingly, a study of Twitter data found that the size of user communities follow Dunbar's number hypothesis despite the platform enabling communication without physical restrictions on social interactions [72].

The idea that mean group size impacts social and cognitive development of species closely resembles the findings in this dissertation with AI agents. Chapter 4 shows how smaller teams within a population can allow agents to autonomously discover efficient divisions of labor and joint policies across multiple teams. Chapter 5 shows how these results can generalize to situations where teammates may not have common interests or even not optimize for the same goals (effectively removing that social connection). Chapter 6 provides theoretical groundwork behind learning environments and team structure. Connecting our research with the study of human teams and acknowledging the impact that social structures have had in human development strengthens the premise of our thesis: team structures can benefit the policies that agents develop and must not be overlooked in the development of AI.

7.4.2 Cellular Level

We identify similarities between our work and biological observations at the cellular level. *Gap junctions* are intercellular connections between neighboring cells that directly connect the cytoplasm of cells (through multiple connection channels), allowing for the diffusion of ions, second messengers, and small molecules [118, 160]. The goals, stresses, and rewards experienced by any connected cells are instantaneously experienced by neighboring cells

with gap junctions. Gap junctions are instrumental in multiple physiological phenomena, including embryonic development and are widely found in all tissues [177]; however, these connections and messages are not always fully shared between cells. Channels connecting cells can be regulated by post translational modifications (i.e., production of new proteins) that affect the channel open probabilities, gating, conductance, or selectivity [155, 10]. Gap junction channels have been found to contain multiple gating mechanisms that regulate the sharing of signals between cells [27] and aggregate to form assembled clusters of cells that are sometimes dynamically adjusted through shared channels [216].

We do not claim that our work contributes to the gap junction literature, but perhaps provides insights into and has interesting parallels to the work studying gap junctions. Cells that form gap junctions share signals and chemicals between each other, forming aggregated structures and plaques of signal-sharing cells. These plaque structures indicate that cells do not form gap junctions and share signals with all other cells, similar to how teammates only share rewards in Chapter 4 instead of sharing with the entire population. *Regulated* gap junctions are those where signals are not entirely shared, similar to how our model of credo regulates the amount of reward that is shared between agents in Chapter 5. Interestingly, the loss of gap junctional inter-cellular communication has been linked to carcinogenesis – the development of cancer cells [126]. Similarly, we find that decreases in agent reward sharing leads to poor global results (i.e., agents with high self-focus), whereas agents that share their rewards with their teammates achieve high rewards overall.

Similar to the self-tuning credo agents presented in Section 6.6 of Chapter 6, the regulation of inter-cell signals through gap junctions is observed to be dynamic through gating, cell aggregation, and channel removal [216]. However, an area of gap junction research that is not well understood is why continuous channel synthesis and removal is evolutionary and physiologically desirable [216]. We explore a similar problem in our context of multiagent teams of AI agents in Chapter 6 to understand why self-focus and mixed incentives may be beneficial for learning. Given some of the similarities between the behavior of gap junctions and of multiagent systems with teams explored in this dissertation, perhaps our findings may be of interest to the gap junction community. Furthermore, gap junction research could help inspire future work on multiagent systems with teams.

7.5 Future Work

There are several areas for future research directions inspired by the work in this dissertation. Some directions are direct short-term extensions of the work presented in the previous chapters, while others are broader long-term expansions.

7.5.1 Direct Short-Term Expansions

Initial directions for future research include exploring teams of unequal size, team structures where agents belong to multiple teams, and conditions under which low-level cooperation (i.e., nepotism or bribery) undermines global progress. Our model of teams allows the implementation of additional infrastructure among the agents beyond only their reward function; thus, longer term questions include analyzing how features such as communication, negotiation, trust, and sanctions impact our model and introduce new challenges. Furthermore, the study of teams in conjunction with emergent social norms, reputation mechanisms, and levels of cooperation are all open directions of future research.

Our work in Chapter 6 emphasizes how the best team structure is highly domain specific and depends on the distribution of reward-causing state-action pairs, reward states, and underlying reward structure of the environment. This raises the question of whether or not beneficial team structures could be learned from domain variables. Exploring this direction of research from the perspective of a social planner is an interesting area of future research.

In Chapter 5 we examined credo with a team structure of three teams with two agents each in Cleanup; however, Chapter 6 analyzes how team size has a significant impact learning. Therefore, an interesting extension of our work would be to examine how the impact of credo changes depending on team structure across different environments. Perhaps the best credo scenarios change regarding the underlying team structure, including the size of teams, number of teams, or if agents have the ability to belong to multiple teams. This also includes analyzing teams with vertical within-team structures such as levels of seniority or leadership. These directions will undoubtedly lead to deeper and more interesting questions behind the impact that team structures have on multiagent learning.

The further development of self-tuning credo agents is another interesting area of future research. We have presented an initial prototype for a decentralized system where agents can self-tune their own credo parameters inspired by hierarchical learning. Our preliminary evaluation shows promising results; agents are able to modify their credo parameters and discover the best observed joint policy of four apple picking and two river cleaning agents after being defined in a sub-optimal team structure/credo setting. However, our prototype remains simplistic and is unable to scale to continuous credo configurations since Q-learning (the credo-tuning policy) requires discrete state and action spaces. Further developing self-tuning credo agents is an exciting area of future work that fully encompasses the work done in this dissertation. These agents would have the ability to self-organize and discover efficient team structures, credo parameters, and potentially evolve to dynamic environments. This will require addressing broader problems of sample complexity, generalizability, and

further developing and understanding the connections with meta-learning.

7.5.2 Broader Long-Term Expansions

Recall from Chapter 2 that the single-agent TRPO algorithm guarantees policy improvements given a penalty β on the KL divergence; however, the authors suggest that choosing a single value of this penalty is challenging. The parameter β essentially restricts the *trusted region* (in parameter space) for policy updates. This space is also influenced by the agent's advantage function, the difference between the expected value of a state and the expected value for taking a specific action in that state. Throughout this dissertation we have shown multiple examples of how team structures or credo parameters regulate an agent's reward signal; thus, modifying the space of policy updates for a single agent depending on the information in this signal. For example, by Theorem 2 in Chapter 6, an infinitely large team results in a reward signal with an amount of information that converges to zero; thus, the agent's advantage function would also converge to zero. One can imagine that this would result in an infinitely small policy search space for parameter updates (i.e., the trusted region of updated in TRPO shrinks). Agents' credo parameters also influence the information in reward signals; thus, they also modify the size of this possible search space through the mixing of reward signals. Similar to the claims about β in the TRPO publication, Chapter 6 shows how constructing a favorable team structure from no prior domain knowledge is a challenging problem with a potentially sub-optimal impact on learning. Endowing agents with the ability to self-modify their credo parameters and team alignment may be thought of as a way to *dynamically* modify the policy search space (i.e., learning to dynamically modify β depending on context). Fully exploring the extent to which these problems are related is an interesting area for future work.

Finally, we revisit our motivating examples and suggest avenues for further development in real-world contexts. Future emphasis on team structures and simulating credo parameters could help inform or study how social dilemmas are solved, or not solved, among human agents. This includes developing incentives or team strategies to help solve the wildfire fuel mitigation dilemma that is currently unsolved. Many interesting problems exist in team sports analytics where multiagent systems, with a focus on teams and credo specifically, are promising avenues for exploration. Sports encompasses multiple levels of complexity: short-term strategizing (coaching) and long-term planning (management and development). Professional sports operate as a system of coordinating sub-groups of agents motivated by financial incentives. Some specific areas of future work include predicting a players' credo based on their actions to inform team construction (i.e., who plays with who), evaluating or predicting group development and agency from initial configurations,

learning the best team structures from a social planner perspective, or developing financial incentive strategies to influence a player's credo parameters. We further detail the relationship between multiagent systems and team sports analytics in recently published work [187]. These directions of future work will require immense development of simulations and push our research into real-world domains.

References

[1] Akshat Agarwal, Sumit Kumar, and Katia Sycara. Learning transferable cooperative behavior in multi-agent teams. *arXiv preprint arXiv:1906.01202*, 2019.

[2] Stefano V. Albrecht and Peter Stone. Reasoning about hypothetical agent behaviours and their parameters. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, 2017.

[3] Nicolas Anastassacos, Julian García, Stephen Hailes, and Mirco Musolesi. Cooperation and reputation dynamics with reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[4] Nicolas Anastassacos, Stephen Hailes, and Mirco Musolesi. Partner selection for the emergence of cooperation in multi-agent systems using reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7047–7054, 2020.

[5] Carl Anderson and Nigel R Franks. Teams in animal societies. *Behavioral Ecology*, 12(5):534–540, 2001.

[6] Carl Anderson and Nigel R Franks. Teamwork in animals, robots, and humans. *Advances in The Study of Behavior*, 33:1–48, 2003.

[7] Ewa Andrejczuk, Juan A Rodriguez-Aguilar, and Carles Sierra. A concise review on multiagent teams: contributions and research opportunities. In *European Conference on Multi-Agent Systems*, pages 31–39. Springer, 2016.

[8] Jose A Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. Rudder: Return decomposition for delayed rewards. *Proceedings of the 33rd Conference on Neural Information Processing Systems*, 32, 2019.

[9] Dilip Arumugam, Peter Henderson, and Pierre-Luc Bacon. An information-theoretic perspective on credit assignment in reinforcement learning. *Workshop on Biological and Artificial Reinforcement Learning at the 34th Conference on Neural Information Processing Systems*, 2020.

[10] Lene N Axelsen, Kirstine Calloe, Niels-Henrik Holstein-Rathlou, and Morten S Nielsen. Managing the complexity of communication: Regulation of gap junctions by post-translational modification. *Frontiers in Pharmacology*, 4:130, 2013.

[11] Yoram Bachrach, Richard Everett, Edward Hughes, A. Lazaridou, Joel Z. Leibo, Marc Lanctot, Mike Johanson, Wojciech Czarnecki, and T. Graepel. Negotiating team formation using deep reinforcement learning. *Artificial Intelligence*, 288:103356, 2020.

[12] Bowen Baker. Emergent reciprocity and team formation from randomized uncertain social preferences. *Proceedings of the 34th Conference on Neural Information Processing Systems*, 2020.

[13] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. Emergent tool use from multi-agent autocurricula. In *Proceedings of the 7th International Conference on Learning and Representations*, 2019.

[14] Cecile T Balkanski. *Modelling act-type relations in collaborative activity*. Harvard University, Center for Research in Computing Technology, 1990.

[15] Honglin Bao, Qiqige Wuyun, and Wolfgang Banzhaf. Evolution of cooperation through genetic collective learning and imitation in multiagent societies. In *Artificial Life Conference Proceedings*, pages 436–443. MIT Press, 2018.

[16] Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, Iain Dunning, Shibl Mourad, H. Larochelle, Marc G. Bellemare, and Michael H. Bowling. The Hanabi challenge: A new frontier for AI research. *Artificial Intelligence*, 280:103216, 2020.

[17] Andrew G Barto. Learning by statistical cooperation of self-interested neuron-like computing elements. *Human Neurobiology*, 4(4):229–256, 1985.

[18] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemyslaw Debiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Christopher

Hesse, R. Józefowicz, Scott Gray, Catherine Olsson, Jakub W. Pachocki, Michael Petrov, Henrique Pondé de Oliveira Pinto, Jonathan Raiman, Tim Salimans, Jeremy Schlatter, J. Schneider, S. Sidor, Ilya Sutskever, Jie Tang, F. Wolski, and Susan Zhang. Dota 2 with large scale deep reinforcement learning. *ArXiv*, abs/1912.06680, 2019.

[19] Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4):819–840, 2002.

[20] Cristina Bicchieri. *The grammar of society: The nature and dynamics of social norms*. Cambridge University Press, 2005.

[21] Saskia Bick, Kai Spohrer, Rashina Hoda, Alexander Scheerer, and Armin Heinzl. Coordination challenges in large-scale software development: A case study of planning misalignment in hybrid settings. *IEEE Transactions on Software Engineering*, 44:932–950, 2018.

[22] Torsten Biemann and Eric Kearney. Size does matter: How varying group sizes in a sample affect the most common measures of group diversity. *Organizational Research Methods*, 13(3):582–599, 2010.

[23] Robert Boyd, Herbert Gintis, and Samuel Bowles. Coordinated punishment of defectors sustains cooperation and can proliferate when rare. *Science*, 328(5978):617–620, 2010.

[24] Robert Boyd and Peter J Richerson. Punishment allows the evolution of cooperation (or anything else) in sizable groups. *Ethology and sociobiology*, 13(3):171–195, 1992.

[25] Gordon L Brady. Governing the commons: The evolution of institutions for collective action. *Southern Economic Journal*, 60(1):249–251, 1993.

[26] Garrett W Brown, Iain McLean, and Alistair McMillan. *The concise Oxford dictionary of politics and international relations*. Oxford University Press, 2018.

[27] Feliksas F Bukauskas and Vytas K Verselis. Gap junction channel gating. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1662(1-2):42–60, 2004.

[28] BusinessWire. Global $4.5 billion sports analytics market forecasts up to 2024. https://www.businesswire.com/news/home/20181205005823/en/Global-4.5-Billion-Sports-Analytics-Market-Forecasts, 2018. Accessed: 2021-04-08.

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

[30] Lucian Buşoniu, Robert Babuška, and Bart De Schutter. Multi-agent reinforcement learning: An overview. *Innovations in Multi-agent Systems and Applications-1*, pages 183–221, 2010.

[31] Dorothy R. Carter, Raquel Asencio, Hayley M. Trainer, Leslie A. DeChurch, Ruth Kanfer, and Stephen J. Zaccaro. Best practices for researchers working in multiteam systems. *Strategies for Team Science Success: Handbook of Evidence-Based Principles for Cross-Disciplinary Science and Practical Lessons Learned from Health Researchers*, pages 391–400, 2019.

[32] Lidia Ceriani and Paolo Verme. The origins of the Gini index: extracts from variabilità e mutabilità (1912) by Corrado Gini. *The Journal of Economic Inequality*, 10:421–443, 2012.

[33] Damien Challet and Y-C Zhang. Emergence of cooperation and organization in an evolutionary game. *Physica A: Statistical Mechanics and its Applications*, 246(3-4):407–418, 1997.

[34] Huo-Tsan Chang, Cheng-Chen Lin, C. Chen, and Yeong Ho Ho. Explicit and implicit team coordination: Development of a multidimensional scale. *Social Behavior and Personality*, 45:915–929, 2017.

[35] Paul Chelarescu. Deception in social learning: A multi-agent reinforcement learning perspective. *arXiv preprint arXiv:2106.05402*, 2021.

[36] Stephen Chung. Map propagation algorithm: Faster learning with a team of reinforcement learning agents. *Proceedings of the 35th Conference on Information Processing Systems*, 34, 2021.

[37] Caroline Claus and Craig Boutilier. The dynamics of reinforcement learning in cooperative multiagent systems. *Innovative Applications of Artificial Intelligence*, 1998(746-752):2, 1998.

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

[39] Allan Dafoe, Edward Hughes, Yoram Bachrach, Tantum Collins, Kevin R. McKee, Joel Z. Leibo, Kate Larson, and Thore Graepel. Open problems in cooperative AI. *ArXiv*, abs/2012.08630, 2020.

[40] Panayiotis Danassis, Zeki Doruk Erden, and Boi Faltings. Improved cooperation by exploiting a common signal. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[41] Robert B Davison, John R Hollenbeck, Christopher M Barnes, Dustin J Sleesman, and Daniel R Ilgen. Coordinated action in multiteam systems. *Journal of Applied Psychology*, 97(4):808, 2012.

[42] Robyn M. Dawes and David M. Messick. Social dilemmas. *International Journal of Psychology*, 35(2):111–116, 2000.

[43] Peter Dayan and Geoffrey E. Hinton. Feudal reinforcement learning. *Proceedings of the 6th Conference on Neural Information Processing Systems*, 5, 1992.

[44] Leslie A. DeChurch and Jessica R. Mesmer-Magnus. The cognitive underpinnings of effective teamwork: A meta-analysis. *The Journal of Applied Psychology*, 95 1:32–53, 2010.

[45] Leslie A. DeChurch and Stephen J. Zaccaro. Perspectives: Teams won't solve this problem. *Human Factors: The Journal of Human Factors and Ergonomics Society*, 52:329 – 334, 2010.

[46] Ankur Deka and Katia Sycara. Natural emergence of heterogeneous strategies in artificially intelligent competitive teams. In *International Conference on Swarm Intelligence*, pages 13–25. Springer, 2021.

[47] Jordi Delgado. Emergence of social conventions in complex networks. *Artificial intelligence*, 141(1-2):171–185, 2002.

[48] 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. *Proceedings of the 34rd Conference on Neural Information Processing Systems*, 33:13049–13061, 2020.

[49] Dennis J Devine. A review and integration of classification systems relevant to teams in organizations. *Group Dynamics: Theory, Research, and Practice*, 6(4):291, 2002.

[50] Frank Dignum, Virginia Dignum, Rui Prada, and Catholijn M Jonker. A conceptual architecture for social deliberation in multi-agent organizations. *Multiagent and Grid Systems*, 11(3):147–166, 2015.

[51] Dongsheng Ding, Chen-Yu Wei, Kaiqing Zhang, and Mihailo Jovanovic. Independent policy gradient for large-scale markov potential games: Sharper rates, function approximation, and game-agnostic convergence. In *Proceedings of the 39th International Conference on Machine Learning*, pages 5166–5220. PMLR, 2022.

[52] Heng Dong, Tonghan Wang, Jiayuan Liu, Chi Han, and Chongjie Zhang. Birds of a feather flock together: A close look at cooperation emergence via multi-agent rl. *arXiv preprint arXiv:2104.11455*, 2021.

[53] Yunlong Dong, Shengjun Zhang, Xing Liu, Yu Zhang, and Tan Shen. Variance aware reward smoothing for deep reinforcement learning. *Neurocomputing*, 458:327–335, 2021.

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

[55] Ishan Durugkar, Elad Liebman, and Peter Stone. Balancing individual preferences and shared objectives in multiagent reinforcement learning. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020.

[56] Ramy Elitzur. Data analytics effects in major league baseball. *Omega*, 90:102001, 2020.

[57] M Ellis. Similarities and differences in games: A system for classification. In *International Association for Physical Education in Higher Education Conference*, 1983.

[58] Elliot E Entin and Daniel Serfaty. Adaptive team coordination. *The Journal of Human Factors and Ergonomics Society*, 41:312 – 325, 1999.

[59] Ernst Fehr and Urs Fischbacher. Social norms and human cooperation. *Trends in cognitive sciences*, 8(4):185–190, 2004.

[60] Ernst Fehr and Urs Fischbacher. Third-party punishment and social norms. *Evolution and Human Behavior*, 25(2):63–87, 2004.

[61] Ernst Fehr and Ivo Schurtenberger. Normative foundations of human cooperation. *Nature human behaviour*, 2(7):458–468, 2018.

[62] FindLaw. California code, public resources code - prc section 4291. https://codes.findlaw.com/ca/public-resources-code/prc-sect-4291.html, 2019. Accessed: 2023-05-14.

[63] Stephen M Fiore, Eduardo Salas, and Janis A Cannon-Bowers. Group dynamics and shared mental model development. In *How People Evaluate Others in Organizations*, pages 335–362. Psychology Press, 2013.

[64] David Fitoussi and Moshe Tennenholtz. Minimal social laws. In *Innovative Applications of Artificial Intelligence*, 1998.

[65] Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Proceedings of the 30th Conference on Neural Information Processing Systems*, 29, 2016.

[66] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[67] Babak Fotouhi, Naghmeh Momeni, Benjamin Allen, and Martin A Nowak. Conjoining uncooperative societies facilitates evolution of cooperation. *Nature Human Behaviour*, 2:492–499, 2018.

[68] James H Fowler and Nicholas A Christakis. Cooperative behavior cascades in human social networks. *Proceedings of the National Academy of Sciences*, 107(12):5334–5338, 2010.

[69] Haotian Fu, Hongyao Tang, Jianye Hao, Zihan Lei, Yingfeng Chen, and Changjie Fan. Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

[70] Matthew E Gaston and Marie DesJardins. Agent-organized networks for dynamic team formation. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 230–237, 2005.

[71] Ian Gemp, Kevin R McKee, Richard Everett, Edgar A Duéñez-Guzmán, Yoram Bachrach, David Balduzzi, and Andrea Tacchetti. D3C: Reducing the price of anarchy in multi-agent learning. *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, 2022.

[72] Bruno Gonçalves, Nicola Perra, and Alessandro Vespignani. Modeling users' activity on twitter networks: Validation of dunbar's number. *PloS ONE*, 6(8):e22656, 2011.

[73] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT press, 2016.

[74] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Proceedings of the 28th Conference on Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[75] Barbara J Grosz and Sarit Kraus. Collaborative plans for complex group action. *Artificial Intelligence*, 86(2):269–357, 1996.

[76] Barbara J. Grosz and Candance L. Sidner. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 417–444. MIT Press, Cambridge, MA, 1990.

[77] Jayesh K Gupta, Maxim Egorov, and Mykel Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 66–83. Springer, 2017.

[78] Nico Gürtler, Dieter Büchler, and Georg Martius. Hierarchical reinforcement learning with timed subgoals. *Proceedings of the 35th Conference on Neural Information Processing Systems*, 34:21732–21743, 2021.

[79] David Ha and Yujin Tang. Collective intelligence for deep learning: A survey of recent developments. *Collective Intelligence*, 1(1):26339137221114874, 2022.

[80] J Richard Hackman. From causes to conditions in group research. *Journal of Organizational Behavior*, 33(3):428–444, 2012.

[81] Dylan Hadfield-Menell, McKane Andrus, and Gillian K. Hadfield. Legible normativity for ai alignment: The value of silly rules. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 2019.

[82] Dylan Hadfield-Menell, Stuart J. Russell, P. Abbeel, and A. Dragan. Cooperative inverse reinforcement learning. In *Proceedings of the 30th Conference on Neural Information Processing Systems*, 2016.

[83] Jianye Hao, Dongping Huang, Yi Cai, and Ho-fung Leung. The dynamics of reinforcement social learning in networked cooperative multiagent systems. *Engineering Applications Artificial Intelligence*, 58:111–122, 2017.

[84] Ferry Hendrikx, Kris Bubendorfer, and Ryan Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.

[85] Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz De Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*, 2017.

[86] Esther Herrmann, Josep Call, María Victoria Hernández-Lloreda, Brian Hare, and Michael Tomasello. Humans have evolved specialized skills of social cognition: The cultural intelligence hypothesis. *science*, 317(5843):1360–1366, 2007.

[87] Verlin B Hinsz and Kevin R Betts. Conflict in multiteam situations. *Multiteam Systems: An Organization Form for Synamic and Complex Environments*, 2012.

[88] Sepp Hochreiter, Steven A. Younger, and Peter R. Conwell. Learning to learn using gradient descent. In *Artificial Neural Networks—ICANN*, pages 87–94. Springer, 2001.

[89] Jesse Hoey. Equality and freedom as uncertainty in groups. *Entropy*, 23(11):1384, 2021.

[90] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)*, 42(1):1–31, 2009.

[91] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

[92] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(9):5149–5169, 2021.

[93] David Earl Hostallero, Daewoo Kim, Sangwoo Moon, Kyunghwan Son, Wan Ju Kang, and Yung Yi. Inducing cooperation through reward reshaping based on peer evaluations in deep multi-agent reinforcement learning. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 520–528, 2020.

[94] Junling Hu and Michael P Wellman. Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4(Nov):1039–1069, 2003.

[95] Shuyue Hu and Ho-fung Leung. Achieving coordination in multi-agent systems by stable local conventions under community networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017.

[96] Edward Hughes, Joel Z Leibo, Matthew Phillips, Karl Tuyls, Edgar Dueñez-Guzman, Antonio García Castañeda, Iain Dunning, Tina Zhu, Kevin McKee, Raphael Koster, et al. Inequity aversion improves cooperation in intertemporal social dilemmas. *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 31, 2018.

[97] Daniel R Ilgen, John R Hollenbeck, Michael Johnson, and Dustin Jundt. Teams in organizations: From input-process-output models to IMOI models. *Annual Review of Psychology*, 56:517–543, 2005.

[98] Tommi Jaakkola, Michael Jordan, and Satinder Singh. Convergence of stochastic iterative dynamic programming algorithms. *Proceedings of the 7th Conference on Neural Information Processing Systems*, 6, 1993.

[99] Max Jaderberg, Wojciech Czarnecki, Iain Dunning, Luke Marris, Guy Lever, A. Castañeda, Charlie Beattie, Neil C. Rabinowitz, Ari S. Morcos, Avraham Ruderman, Nicolas Sonnerat, Tim Green, Louise Deason, Joel Z. Leibo, David Silver, Demis Hassabis, Koray Kavukcuoglu, and Thore Graepel. Human-level performance in 3d multiplayer games with population-based reinforcement learning. *Science*, 364:859 – 865, 2019.

[100] Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Çaglar Gülçehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, 2019.

[101] Seyyed Ahmad Javadi, Richard Cloete, Jennifer Cobbe, Michelle Seng Ah Lee, and Jatinder Singh. Monitoring misuse for accountable'artificial intelligence as a service'. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 300–306, 2020.

[102] Nicholas Jennings. On being responsible. *3rd European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, 1992.

[103] Nicholas Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, 75:195–240, 1995.

[104] Idar A. Johannessen and Jan R. Jonassen. A subsea operation in action: A brief overview of how imr subsea operations are organized and executed. *Technical Report, Western Norway University*, 2018.

[105] Julio Juárez, Cipriano Santos, and Carlos Brizuela. A comprehensive review and a taxonomy proposal of team formation problems. *ACM Computing Surveys*, 54:1 – 33, 2021.

[106] Radu Jurca and Boi Faltings. An incentive compatible reputation mechanism. *IEEE International Conference on E-Commerce*, pages 285–292, 2003.

[107] Simran Kaur, Jeremy Cohen, and Zachary C Lipton. On the maximum Hessian eigenvalue and generalization. *arXiv preprint arXiv:2206.10654*, 2022.

[108] Harold Kelley and John Thibaut. Interpersonal relations: A theory of interdependence. *Journal Marriage and Families*, 1978.

[109] Ravshanbek Khodzhimatov, Stephan Leitner, and Friederike Wall. Interactions between social norms and incentive mechanisms in organizations. In *Coordination, Organizations, Institutions, Norms, and Ethics for Governance of Multi-Agent Systems XIV: International Workshop, COINE 2021, London, UK, May 3, 2021, Revised Selected Papers*, pages 111–126. Springer, 2022.

[110] David Kinny, Elizabeth Sonenberg, Magnus Ljungberg, Gil Tidhar, Anand Rao, and Eric Werner. Planned team activity. In *Artificial Social Systems: 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pages 227–256. Springer, 1994.

[111] Hiroaki Kitano, Minoru Asada, Yasuo Kuniyoshi, Itsuki Noda, and Eiichi Osawa. Robocup: The robot world cup initiative. In *Proceedings of the first international conference on Autonomous agents*, pages 340–347, 1997.

[112] George Dimitri Konidaris and Andrew G Barto. Building portable options: Skill transfer in reinforcement learning. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 895–900, 2007.

[113] Raphael Köster, Dylan Hadfield-Menell, Gillian K. Hadfield, and Joel Z. Leibo. Silly rules improve the capacity of agents to learn stable enforcement and compliance behaviors. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2020.

[114] Landon Kraemer and Bikramjit Banerjee. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, 190:82–94, 2016.

[115] Sarit Kraus. Negotiation and cooperation in multi-agent environments. *Artificial Intelligence*, 94:79–97, 1997.

[116] Tabajara Krausburg, Jürgen Dix, and Rafael Heitor Bordini. Feasible coalition sequences. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[117] Marcus Krellner and The Anh Han. Putting oneself in everybody's shoes-pleasing enables indirect reciprocity under private assessments. In *The 2020 Conference on Artificial Life*, pages 402–410. MIT Press, 2020.

[118] Nalin M Kumar and Norton B Gilula. The gap junction communication channel. *Cell*, 84(3):381–388, 1996.

[119] Matthew Lai. Giraffe: Using deep reinforcement learning to play chess. *ArXiv*, abs/1509.01549, 2015.

[120] Klodiana Lanaj, John R. Hollenbeck, Daniel R. Ilgen, Christopher M. Barnes, and Stephen J. Harmon. The double-edged sword of decentralized planning in multiteam systems. *Academy of Management Journal*, 56:735–757, 2013.

[121] Mathieu Lavallée and Pierre N Robillard. Are we working well with others? how the multi team systems impact software quality. *e-Informatica Software Engineering Journal*, 12(1):117–131, 2018.

[122] Dohoon Lee and Sara McLanahan. Family structure transitions and child development: Instability, selection, and population heterogeneity. *American Sociological Review*, 80(4):738–763, 2015.

[123] Joel Z. Leibo, Edward Hughes, Marc Lanctot, and Thore Graepel. Autocurricula and the emergence of innovation from social interaction: A manifesto for multi-agent intelligence research. *ArXiv*, abs/1903.00742, 2019.

[124] Joel Z. Leibo, Julien Pérolat, Edward Hughes, S. Wheelwright, Adam H. Marblestone, Edgar A. Duéñez-Guzmán, Peter Sunehag, Iain Dunning, and Thore Graepel. Malthusian reinforcement learning. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019.

[125] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel. Multi-agent reinforcement learning in sequential social dilemmas. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, 2017.

[126] Edward Leithe, Solveig Sirnes, Yasufumi Omori, and Edgar Rivedal. Downregulation of gap junctions in cancer cells. *Critical Reviews™ in Oncogenesis*, 12(3-4), 2006.

[127] Jeffery A LePine, Ronald F Piccolo, Christine L Jackson, John E Mathieu, and Jessica R Saul. A meta-analysis of teamwork processes: Tests of a multidimensional model and relationships with team effectiveness criteria. *Personnel Psychology*, 61(2):273–307, 2008.

[128] Annick Lesne. Shannon entropy: A rigorous notion at the crossroads between probability, information theory, dynamical systems and statistical physics. *Mathematical Structures in Computer Science*, 24(3), 2014.

[129] Hector Levesque, Philip R. Cohen, and José H. T. Nunes. On acting together. In *The AAAI Conference on Artificial Intelligence*, 1990.

[130] Michael Lewis. *Moneyball: The art of winning an unfair game.* WW Norton & Company, 2004.

[131] Toru Lin, Jacob Huh, Christopher Stauffer, Ser Nam Lim, and Phillip Isola. Learning to ground multi-agent communication with autoencoders. *Proceedings of the 35th Conference on Neural Information Processing Systems*, 34:15230–15242, 2021.

[132] Per Lindström, Ludwig Jacobsson, Niklas Carlsson, and Patrick Lambrix. Predicting player trajectories in shot situations in soccer. In Ulf Brefeld, Jesse Davis, Jan Van Haaren, and Albrecht Zimmermann, editors, *Machine Learning and Data Mining for Sports Analytics*, pages 62–75, Cham, 2020. Springer International Publishing.

[133] Jinfei Liu. Absolute Shapley value. *ArXiv*, abs/2003.10076, 2020.

[134] Dennis Ljung, Niklas Carlsson, and Patrick Lambrix. Player pairs valuation in ice hockey. In *Machine Learning and Data Mining for Sports Analytics: 5th International Workshop, MLSA 2018*, pages 82–92. Springer, 2019.

[135] Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, OpenAI, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Proceedings of the 31st Conference on Neural Information Processing Systems*, 30, 2017.

[136] Margaret M Luciano, Leslie A. DeChurch, and John Mathieu. Multiteam systems: A structural framework and meso-theory of system functioning. *Journal of Management*, 44:1065 – 1096, 2018.

[137] W. Macke, R. Mirsky, and P. Stone. Expected value of communication for planning in ad hoc teamwork. In *The AAAI Conference on Artificial Intelligence*, 2021.

[138] Michelle A. Marks, Leslie A. DeChurch, John Mathieu, Frederick J. Panzer, and Alexander Alonso. Teamwork in multiteam systems. *The Journal of Applied Psychology*, 90 5:964–71, 2005.

[139] Michelle A. Marks, John E. Mathieu, and Stephen J. Zaccaro. A temporally based framework and taxonomy of team processes. *Academy of Management Review*, 26:356–376, 2001.

[140] Mehdi Mashayekhi, Nirav Ajmeri, George F List, and Munindar P Singh. Prosocial norm emergence in multi-agent systems. *ACM Transactions on Autonomous and Adaptive Systems (TAAS)*, 17(1-2):1–24, 2022.

[141] John E Mathieu, John R Hollenbeck, Daan van Knippenberg, and Daniel R Ilgen. A century of work teams in the Journal of Applied Psychology. *Journal of Applied Psychology*, 102(3):452, 2017.

[142] John E Mathieu, Margaret M Luciano, and Leslie A DeChurch. Multiteam systems: The next chapter. *The SAGE Handbook of Industrial, Work & Organizational Psychology*, 2:333–353, 2018.

[143] John E. Mathieu, Michelle A. Marks, and S. J. Zaccaro. Multiteam systems. *Handbook of Industrial, Work, and Ordanizational Psychology*, 2, 2001.

[144] Laëtitia Matignon, Guillaume J Laurent, and Nadine Le Fort-Piat. Hysteretic Q-learning: an algorithm for decentralized reinforcement learning in cooperative multi-agent teams. In *International Conference on Intelligent Robots and Systems*, pages 64–69. IEEE, 2007.

[145] Kevin R. McKee, Ian Gemp, Brian McWilliams, Edgar A. Duéñez-Guzmán, Edward Hughes, and Joel Z. Leibo. Social diversity and social preferences in mixed-motive reinforcement learning. *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2020.

[146] Michael J Meaney. Epigenetics and the biological definition of gene× environment interactions. *Child Development*, 81(1):41–79, 2010.

[147] Ramona Merhej, Fernando P Santos, Francisco S Melo, Mohamed Chetouani, and Francisco C Santos. Cooperation and learning dynamics under risk diversity and financial incentives. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, pages 908–916, 2022.

[148] Ramona Merhej, Fernando P Santos, Francisco S Melo, and Francisco C Santos. Cooperation between independent reinforcement learners under wealth inequality and collective risks. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[149] Ramona Merhej, Fernando P Santos, Francisco S Melo, and Francisco C Santos. Cooperation and learning dynamics under wealth inequality and diversity in individual risk. *Journal of Artificial Intelligence Research*, 74:733–764, 2022.

[150] Manfred Milinski, Ralf D Sommerfeld, Hans-Jürgen Krambeck, Floyd A Reed, and Jochem Marotzke. The collective-risk social dilemma and the prevention of simulated dangerous climate change. *Proceedings of the National Academy of Sciences*, 105(7):2291–2294, 2008.

[151] Marvin Minsky. Steps toward artificial intelligence. *Proceedings of the IRE*, 49(1):8–30, 1961.

[152] Reuth Mirsky, William Macke, Andy Wang, Harel Yedidsion, and Peter Stone. A penny for your thoughts: The value of communication in ad hoc teamwork. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, 2020.

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

[154] Javier Morales, Maite Lopez-Sanchez, Juan A Rodriguez-Aguilar, Michael J Wooldridge, and Wamberto Weber Vasconcelos. Automated synthesis of normative systems. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, 2013.

[155] Alonso P Moreno and Alan F Lau. Gap junction channel gating modulated through protein phosphorylation. *Progress in Biophysics and Molecular Biology*, 94(1-2):107–119, 2007.

[156] I. Moust. Fighting fire with fire: Team learning in multi-team systems. Master's thesis, Maastricht University, 2011.

[157] Michael Muthukrishna. Corruption, cooperation, and the evolution of prosocial institutions. *Evonomics*, 2017.

[158] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Proceedings of the 32nd Conference on Neural Information Processing Systems*, 31, 2018.

[159] John Nash. Equilibrium points in N-person games. *Proceedings of the National Academy of Sciences of the United States of America*, 36 1:48–9, 1950.

[160] Joost Neijssen, Baoxu Pang, and Jacques Neefjes. Gap junction-mediated intercellular communication in the immune system. *Progress in Biophysics and Molecular Biology*, 94(1-2):207–218, 2007.

[161] Johnathan K. Nelson, Stephen Zaccaro, and Jeffrey L. Herman. Strategic information provision and experiential variety as tools for developing adaptive leadership skills. *Consulting Psychology Journal: Practice and Research*, 62:131–142, 2010.

[162] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *Proceedings of the 16th International Conference on Machine Learning*, volume 99, pages 278–287, 1999.

[163] Eleni Nisioti and Clément Moulin-Frier. Grounding artificial intelligence in the origins of human behavior. *ArXiv*, abs/2012.08564, 2020.

[164] Peter Norton. Computing defensibility. Master's thesis, UC Berkeley, 2020.

[165] Michael Noukhovitch, Travis Lacroix, Angeliki Lazaridou, and Aaron C. Courville. Emergent communication under competition. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[166] Frans A Oliehoek, Matthijs TJ Spaan, and Nikos Vlassis. Optimal and approximate Q-value functions for decentralized POMDPs. *Journal of Artificial Intelligence Research*, 32:289–353, 2008.

[167] Alexi Orchard and David Radke. An analysis of engineering students' responses to an AI ethics scenario. *The 13th Symposium on Educational Advances in Artificial Intelligence*, 2023.

[168] Afshin Oroojlooy and Davood Hajinezhad. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence*, pages 1–46, 2022.

[169] Elinor Ostrom. Governing the commons: The evolution of institutions for collective action. *Natural Resources Journal*, 32:415, 1990.

[170] Elinor Ostrom. Tragedy of the commons. *The New Palgrave Dictionary of Economics*, 2:1–4, 2008.

[171] Alan O'Sullivan. Dispersed collaboration in a multi-firm, multi-team product-development project. *Journal of Engineering and Technology Management*, 20:93–116, 2003.

[172] Gregory Palmer, Rahul Savani, and Karl Tuyls. Negative update intervals in deep multi-agent reinforcement learning. *arXiv preprint arXiv:1809.05096*, 2018.

[173] Alexander Peysakhovich and Adam Lerer. Prosocial learning agents solve generalized stag hunts better than selfish ones. *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, 2018.

[174] Thomy Phan, Fabian Ritz, Lenz Belzner, Philipp Altmann, Thomas Gabor, and Claudia Linnhoff-Popien. Vast: Value function factorization with variable agent sub-teams. *Proceedings of the 35th Conference on Neural Information Processing Systems*, 34:24018–24032, 2021.

[175] Andrew Pilny, Alex Yahja, Marshall Scott Poole, and Melissa Dobosh. A dynamic social network experiment with multi-team systems. In *Proceedings of the 4th IEEE International Conference on Big Data and Cloud Computing*, pages 587–593, 2014.

[176] Flávio L Pinheiro, Jorge M Pacheco, and Francisco C Santos. From local to global dilemmas in social networks. *PLoS ONE*, 7, 2012.

[177] Kristin Pogoda, Petra Kameritsch, Mauricio A Retamal, and José L Vega. Regulation of gap junction channels and hemichannels by phosphorylation and redox changes: a revision. *BMC Cell Biology*, 17:137–150, 2016.

[178] Martha E. Pollack. A model of plan inference that distinguishes between the beliefs of actors and observers. In *ACL*, 1986.

[179] Martha E. Pollack. Plans as complex mental attitudes. In *Intentions in Communication*, pages 77–103. MIT Press, 1990.

[180] Julia Poncela, Jesús Gómez-Gardeñes, Arne Traulsen, and Yamir Moreno. Evolutionary game dynamics in a growing structured population. *New Journal of Physics*, 11(8):083031, aug 2009.

[181] Jeanine P Porck, Fadel K Matta, John R Hollenbeck, Jo K Oh, Klodiana Lanaj, and Stephanie M Lee. Social identification in multiteam systems: The role of depletion and task complexity. *Academy of Management Journal*, 62(4):1137–1162, 2019.

[182] David V. Pynadath and Milind Tambe. An automated teamwork infrastructure for heterogeneous software agents and humans. *Proceedings of the 3rd International Conference on Autonomous Agents and Multiagent Systems*, 7:71–100, 2004.

[183] David Radke, Anna Hessler, and Daniel Ellsworth. Firecast: Leveraging deep learning to predict wildfire spread. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

[184] David Radke, Kate Larson, and Tim Brecht. Exploring the benefits of teams in multiagent learning. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence*, 2022.

[185] David Radke, Kate Larson, and Tim Brecht. The importance of credo in multiagent learning. *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

[186] David Radke, Kate Larson, and Tim Brecht. Towards a better understanding of learning with multiagent teams. *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*, 2023.

[187] David Radke and Alexi Orchard. Presenting multiagent challenges in team sports analytics. *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

[188] David Radke, Daniel Radke, Tim Brecht, and Alex Pawelczyk. Passing and pressure metrics in ice hockey. *Workshop of AI for Sports Analytics*, 2021.

[189] David Radke, Daniel Radke, Tim Brecht, and Alex Pawelczyk. Passing and pressure metrics in ice hockey. In *Artificial Intelligence for Sports Analytics Workshop at the the 30th International Joint Conference on Artificial Intelligence*, 2021.

[190] David Radke, Daniel Radke, and John Radke. Beyond measurement: Extracting vegetation height from high resolution imagery with deep learning. *Journal of Remote Sensing*, 12:3797, 2020.

[191] David Radke and Kyle Tilbury. Learning to learn group alignment: A self-tuning credo framework with multiagent teams. *Adaptive and Learning Workshop at the 22nd International Conference on Autonomous Agents and Multiagent Systems*, 2023.

[192] Talal Rahwan, Tomasz P Michalak, Michael Wooldridge, and Nicholas R Jennings. Coalition structure generation: A survey. *Artificial Intelligence*, 229:139–174, 2015.

[193] David G. Rand and Martin A. Nowak. Human cooperation. *Trends in Cognitive Sciences*, 17:413–425, 2013.

[194] Anatol Rapoport. Prisoner's dilemma—recollections and observations. In *Game Theory as a Theory of a Conflict Resolution*, pages 17–34. Springer, 1974.

[195] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *The Journal of Machine Learning Research*, 21(1):7234–7284, 2020.

[196] Manish Ravula, Shani Alkoby, and Peter Stone. Ad hoc teamwork with behavior switching agents. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019.

[197] Paul Resnick, Ko Kuwabara, Richard Zeckhauser, and Eric Friedman. Reputation systems. *Commun. ACM*, 43:45–48, 2000.

[198] Peter Richerson. *The Evolution of Human Ultra-sociality*. University of California, Davis, Davis, California, 2.01 edition, 1998.

[199] John E. Roemer. A theory of cooperation in games with an application to market socialism. *Review of Social Economy*, 77:1 – 28, 2019.

[200] Joshua Romoff, Peter Henderson, Alexandre Piché, Vincent Francois-Lavet, and Joelle Pineau. Reward estimation for variance reduction in deep reinforcement learning. *arXiv preprint arXiv:1805.03359*, 2018.

[201] Ariel Rubinstein, Harold William Kuhn, Oskar Morgenstern, and John Von Neumann. *Theory of Games and Economic Behavior: 60th Anniversary Commemorative Edition*. Princeton University Press, 2007.

[202] Heechang Ryu, Hayong Shin, and Jinkyoo Park. Multi-agent actor-critic with hierarchical graph attention network. In *The AAAI Conference on Artificial Intelligence*, 2020.

[203] Heechang Ryu, Hayong Shin, and Jinkyoo Park. Cooperative and competitive biases for multi-agent reinforcement learning. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[204] Fernando P. Santos, Samuel Mascarenhas, Francisco C. Santos, Filipa Correia, Samuel Gomes, and A. Paiva. Picky losers and carefree winners prevail in collective risk dilemmas with partner selection. *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 34:1–29, 2020.

[205] Fernando P Santos, Francisco C Santos, and Jorge M Pacheco. Social norm complexity and past reputations in the evolution of cooperation. *Nature*, 555(7695):242–245, 2018.

[206] Fernando P Santos, Francisco C Santos, Jorge M Pacheco, and Simon A Levin. Social network interventions to prevent reciprocity-driven polarization. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[207] Fernando P Santos, Francisco C Santos, Ana Paiva, and Jorge M Pacheco. Evolutionary dynamics of group fairness. *Journal of Theoretical Biology*, 378:96–102, 2015.

[208] Francisco C Santos, Jorge M Pacheco, and Tom Lenaerts. Cooperation prevails when individuals adjust their social ties. *PLoS Computational Biology*, 2, 2006.

[209] Alexander Scheerer and Thomas Kude. Coordination in large-scale agile software development: A multiteam systems perspective. *Proceedings of the 47th Hawaii International Conference on System Sciences*, pages 4780–4788, 2014.

[210] Eric Schnell, Robin Schimmelpfennig, and Michael Muthukrishna. The size of the stag determines the level of cooperation. *bioRxiv*, 2021.

[211] Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, et al. Mastering Atari, Go, Chess and Shogi by planning with a learned model. *Nature*, 588(7839):604–609, 2020.

[212] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1889–1897. PMLR, 2015.

[213] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[214] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *Computing Research Repository*, 2017.

[215] Nicolas Schwind, Emir Demirovic, Katsumi Inoue, and Jean-Marie Lagniez. Partial robustness in team formation: Bridging the gap between robustness and resilience. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[216] Dominique Segretain and Matthias M Falk. Regulation of connexin biosynthesis, assembly, gap junction formation, and removal. *Biochimica et Biophysica Acta (BBA)-Biomembranes*, 1662(1-2):3–21, 2004.

[217] Sven Seuken and Shlomo Zilberstein. Formal models and algorithms for decentralized decision making under uncertainty. *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, 17:190–250, 2008.

[218] Rohin Shah, Pedro Freire, Neel Alex, Rachel Freedman, Dmitrii Krasheninnikov, Lawrence Chan, Michael D Dennis, Pieter Abbeel, Anca Dragan, and Stuart Russell. Benefits of assistance over reward learning. *In the 34th Conference Neural Information Processing Systems Workshop on Cooperative AI*, 2020.

[219] Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948.

[220] Lloyd S Shapley. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953.

[221] Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence*, 73:231–252, 1995.

[222] Karl Sigmund. The calculus of selfishness. *Princeton Series in Theoretical and Computational Biology*, 6, 2010.

[223] David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, Timothy Lillicrap, Karen Simonyan, and Demis Hassabis. A general reinforcement learning algorithm that masters Chess, Shogi, and Go through self-play. *Science*, 362:1140 – 1144, 2018.

[224] J. A. Simpson and E. S. C. Weiner. *The Oxford English Dictionary*. Oxford University Press, 1989.

[225] Satinder Singh, Michael J Kearns, and Yishay Mansour. Nash convergence of gradient dynamics in general-sum games. In *Uncertainty in Artificial Intelligence*, pages 541–548, 2000.

[226] Peter Stone, Gal Kaminka, Sarit Kraus, and Jeffrey Rosenschein. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *The AAAI Conference on Artificial Intelligence*, 2010.

[227] Peter Stone and Manuela Veloso. Multiagent systems: A survey from a machine learning perspective. *Autonomous Robots*, 8:345–383, 2000.

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

[229] Eric Sundstrom, Kenneth De Meuse, and David Futrell. Work teams: Applications and effectiveness. *American Psychologist*, 45(2):120, 1990.

[230] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT Press, 2018.

[231] Richard S Sutton, Doina Precup, and Satinder Singh. Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999.

[232] Gyorgy Szabó and Gabor Fáth. Evolutionary games on graphs. *Physics Reports*, 446:97–216, 2007.

[233] Milind Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[234] Milind Tambe. Implementing agent teams in dynamic multiagent environments. *Journal on Applied Artificial Intelligence*, 12:189–210, 1998.

[235] Milind Tambe and Weixiong Zhang. Towards flexible teamwork in persistent teams. *Proceedings of the 4th International Conference on Autonomous Agents and Multiagent Systems*, 3:159–183, 2000.

[236] Ming Tan. Multi-agent reinforcement learning: Independent vs. cooperative agents. In *Proceedings of the 10th International Conference on Machine Learning*, pages 330–337, 1993.

[237] Andreia Sofia Teixeira, Francisco C Santos, Alexandre P Francisco, and Fernando P Santos. Eliciting fairness in n-player network games through degree-based role assignment. *Complexity*, 2021:1–11, 2021.

[238] Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. *Learning to Learn*, pages 3–17, 1998.

[239] Sebastian B Thrun. Efficient exploration in reinforcement learning. *Technical Report, Carnegie Mellon University*, 1992.

[240] Michael Tomasello, Alicia Melis, Claudio Tennie, Emily Wyman, and Esther Herrmann. Two key steps in the evolution of human cooperation. *Current Anthropology*, 53:673 – 692, 2012.

[241] Michael Tomasello and Amrisha Vaish. Origins of human cooperation and morality. *Annual Review of Psychology*, 64:231–55, 2013.

[242] Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem*. Society for Industrial and Applied Mathematics, 2002.

[243] Mikhail L'vovich Tsetlin. *Automaton theory and modeling of biological systems*, volume 102. Academic Press New York, 1973.

[244] Sjir Uitdewilligen and Mary J Waller. Adaptation in multiteam systems: The role of temporal semistructures. In *Multiteam Systems*, pages 376–405. Routledge, 2012.

[245] Debra Umberson and Mieke Beth Thomeer. Family matters: Research on family ties and health, 2010 to 2020. *Journal of Marriage and Family*, 82(1):404–419, 2020.

[246] Paul AM Van Lange and Daniel Balliet. Interdependence theory. *APA Handbook of Personality and Social Psychology*, 3:65–92, 2014.

[247] Andrew V. D. Ven, Andre Delbecq, and Richard Koenig. Determinants of coordination modes within organizations. *American Sociological Review*, 41:322, 1976.

[248] Daniel Villatoro, Sandip Sen, and Jordi Sabater-Mir. Of social norms and sanctioning: A game theoretical overview. *International Journal of Agent Technologies and Systems*, 2:1–15, 2010.

[249] Eugene Vinitsky, Natasha Jaques, Joel Leibo, Antonio Castenada, and Edward Hughes. An open source implementation of sequential social dilemma games. https://github.com/eugenevinitsky/sequential_social_dilemma_games/issues/182, 2019. GitHub repository.

[250] Eugene Vinitsky, Raphael Koster, John Agapiou, Edgar A. Duéñez-Guzmán, Alexander Vezhnevets, and Joel Z. Leibo. A learning agent that acquires social norms from public sanctions in decentralized multi-agent settings. *Collective Intelligence*, 2021.

[251] Oriol Vinyals, I. Babuschkin, Wojciech Czarnecki, Michaël Mathieu, Andrew Dudzik, J. Chung, D. Choi, Richard Powell, Timo Ewalds, P. Georgiev, Junhyuk Oh, Dan Horgan, M. Kroiss, Ivo Danihelka, Aja Huang, L. Sifre, Trevor Cai, J. Agapiou, Max Jaderberg, A. Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, D. Budden, Yury Sulsky, James Molloy, T. Paine, Caglar Gulcehre, Ziyu Wang, T. Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019.

[252] Thomas De Vries, John Hollenbeck, Robert Davison, Frank Walter, and Gerben V. D. Vegt. Managing coordination in multiteam systems: Integrating micro and macro perspectives. *Academy of Management Journal*, 59:1823–1844, 2016.

[253] Caroline Wang, Ishan Durugkar, Elad Liebman, and Peter Stone. DM^2: Distributed multi-agent reinforcement learning for distribution matching. *The AAAI Conference on Artificial Intelligence*, 2023.

[254] Jane X. Wang, Edward Hughes, Chrisantha Fernando, Wojciech M. Czarnecki, Edgar A. Duéñez-Guzmán, and Joel Z. Leibo. Evolving intrinsic motivations for altruistic behavior. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 683–692, 2019.

[255] Rose E. Wang, Sarah A. Wu, James A. Evans, J. Tenenbaum, D. Parkes, and Max Kleiman-Weiner. Too many cooks: Coordinating multi-agent collaboration through inverse planning. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, 2020.

[256] Xin Wang, Yudong Chen, and Wenwu Zhu. A survey on curriculum learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[257] Christopher Watkins and Peter Dayan. Q-learning. *Machine learning*, 8:279–292, 1992.

[258] Théophane Weber, Nicolas Heess, Lars Buesing, and David Silver. Credit assignment techniques in stochastic computation graphs. In *Proceedings of the 24th International Conference on Artificial Intelligence and Statistics*, pages 2650–2660. PMLR, 2019.

[259] Jorgen Weibull. Evolutionary game theory. *Journal of Artificial Societies and Social Simulation*, 1994.

[260] Matthias Weiss and Martin Hoegl. The history of teamwork's societal diffusion: A multi-method review. *Small Group Research*, 46(6):589–622, 2015.

[261] Eric Wiewiora, Garrison W Cottrell, and Charles Elkan. Principled methods for advising reinforcement learning agents. In *Proceedings of the 20th International Conference on Machine Learning*, pages 792–799, 2003.

[262] Julia Wijnmaalen, Hans Voordijk, Sebastiaan Rietjens, and Geert Dewulf. Intergroup behavior in military multiteam systems. *Human Relations*, 72:1081 – 1104, 2019.

[263] Mason Wright and Yevgeniy Vorobeychik. Mechanism design for team formation. In *The AAAI Conference on Aritificial Intelligence*, 2015.

[264] Jason Xu, Julián García, and Toby Handfield. Cooperation with bottom-up reputation dynamics. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, 2019.

[265] Tom Yan, Christian Kroer, and Alex Peysakhovich. Evaluating and rewarding teamwork using cooperative game abstractions. *Proceedings of the 34th Conference on Neural Information Processing Systems*, 2020.

[266] Jiachen Yang, Ang Li, Mehrdad Farajtabar, Peter Sunehag, Edward Hughes, and Hongyuan Zha. Learning to incentivize other learning agents. *Proceedings of the 34th Conference on Neural Information Processing Systems*, 33:15208–15219, 2020.

[267] Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, B. Liu, Jia Chen, Z. Liu, Fuhao Qiu, Hongsheng Yu, Yinyuting Yin, Bei Shi, L. Wang, Tengfei Shi, Qiang Fu, Wei Yang, Lanxiao Huang, and Wei Liu. Towards playing full moba games with deep reinforcement learning. *ArXiv*, abs/2011.12692, 2020.

[268] Yuxuan Yi, Ge Li, Yaowei Wang, and Zongqing Lu. Learning to share in multi-agent reinforcement learning. *arXiv preprint arXiv:2112.08702*, 2021.

[269] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. *Proceedings of the 36th Conference on Neural Information Processing Systems*, 35:24611–24624, 2022.

[270] Stephen J. Zaccaro, Samantha Dubrow, Elisa M. Torres, and Lauren N.P. Campbell. Multiteam systems: An integrated review and comparison of different forms. *Annual Review of Organizational Psychology and Organizational Behavior*, 7(1):479–503, 2020.

[271] Stephen J. Zaccaro, Michelle A. Marks, and Leslie A. DeChurch. Multiteam systems: An introduction. In *Multiteam Systems*, pages 18–47. Routledge, 2012.

[272] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of Reinforcement Learning and Control*, pages 321–384, 2021.

[273] Ke Zhang, Fang He, Zhengchao Zhang, Xi Lin, and Meng Li. Multi-vehicle routing problems with soft time windows: A multi-agent reinforcement learning approach. *Transportation Research Part C: Emerging Technologies*, 121:102861, 2020.

[274] X. Zhang, Hui hua Liu, Yushen Shi, and D. Tjosvold. Conflict management for coordination between shift teams in Shanghai subway stations. *Asia Pacific Journal of Human Resources*, 57:399–417, 2019.

[275] Stephan Zheng, Alexander Trott, Sunil Srinivasa, Nikhil Naik, Melvin Gruesbeck, David Parkes, and R. Socher. The AI economist: Improving equality and productivity with AI-driven tax policies. *ArXiv*, abs/2004.13332, 2020.

[276] Simon Zhuang and Dylan Hadfield-Menell. Consequences of misaligned AI. *Proceedings of the 34th Conference on Neural Information Processing Systems*, 33:15763–15773, 2020.

APPENDICES

Appendix A

Counterpart Sampling in the Iterated Prisoner's Dilemma

A.1 Training Samples Theory

Chapter 4 shows how agents with teammates are able to learn cooperative policies in the Iterated Prisoner's Dilemma (IPD) in various environmental conditions. One assumption we make when designing the IPD scenario is that every team is the same size and agents sample a counterpart from any team with equal probability (i.e., uniform sampling over teams). When using off-policy Deep Q-Network reinforcement learning agents, these assumptions ensured that each agent was training on the same number of samples in the limit. We prove this aspect of our design here.

In each episode, agents all play a round of the Prisoner's Dilemma stage game against a counterpart for a total of N pairings per-episode. Agents are given a counterpart to interact with, but could also be selected as the counterpart for another agent. Since agents are individual learners and only learn through their own direct interactions, we must ensure that the particular matching process does not disproportionally bias a subset of the agents. In particular, we need to be confidant that the underlying team structure in which agents are embedded in no way influences the agent training through under- or over-sampling or providing disproportionate opportunities to be matched and play an iteration of the IPD.

Proposition 2. *If* $|T_i| = |T_j| \; \forall i, j \in N$ *and agents are randomly paired from any team with uniform probability, each agent will have the same expected number of IPD interactions for any value of* $|T|$ *or* N*.*

Proof. Let a population of N agents be split up into $|\mathcal{T}|$ teams of size n, so that $N = |\mathcal{T}|n$. Since agents are paired with an agent from any team with equal probability, $p(IN) = 1 - \frac{1}{|\mathcal{T}|(n-1)}$ and $p(OUT) = 1 - \frac{1}{|\mathcal{T}|n}$ represents the probability of **not** being matched with a teammate or non-teammate respectively. These are different since an agent is unable to be paired with themselves, leaving $n-1$ agents to possibly be paired with from their own team. The probability of agent i not being chosen as the matching agent is defined as:

$$p(\neg i)_{|\mathcal{T}|n} = p(IN)^{n-1} + p(OUT)^{n(|\mathcal{T}|-1)}.$$

Suppose m agents are added to each team so that $N' = |\mathcal{T}|n + |\mathcal{T}|m$ and $n := n + m$. In this new setting, the probability of i not being chosen in a population of $|\mathcal{T}|(n+m)$ agents becomes:

$$p(\neg i)_{|\mathcal{T}|(n+m)} = \\ p(IN)^{(n-1)+m} + p(OUT)^{n(|\mathcal{T}|-1)+(|\mathcal{T}|m-m)}.$$

We can derive that $p(\neg i)_{|\mathcal{T}|(n+m)} - p(\neg i)_{|\mathcal{T}|n} = (|\mathcal{T}|m - m) + m$, which simplifies to $|\mathcal{T}|m$. Note that also $N' - N = |\mathcal{T}|m$. While the probability of not being chosen increases by $|\mathcal{T}|m$, the total interactions in each episode also increases by $|\mathcal{T}|m$. Thus, agents have the same number of expected interactions. □

Appendix B

Equilibrium Analysis with Credo

B.1 Expanded Equilibrium Analysis with Credo

As review, we re-define the IPD payoff matrices for the cases where agents have no common interest (Table B.1) and where agents have full common interest (Table B.2). These tables are the same as in Chapter 3.

Chapter 4 presents an equilibrium analysis with the IPD and team structures under the assumption that agents in a team are fully team-focused. 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 B.1. 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. We show the derivation for team-focused agents and continue with the final equilibrium with credo below.

B.1.1 Team-Focused Agents

Let $\sigma_{T_i} = (\sigma_{ji}, 1 - \sigma_{ji})$, where σ_{ji} is the probability for agent j choosing action C when i and j have common interest (i.e., $j \in T_i$). Let $\sigma_{T_j} = (\sigma_{jj}, 1 - \sigma_{jj})$ be the strategy profile of agent j when i and j do not have common interest (i.e., $j \in T_j$). We now use common interest instead of strictly in the same team to scale to the self- and system-focus settings. 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_i} or σ_{T_j}). First we show the derivation for a fully team-focused agent i's expected utility for choosing C

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

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

	Cooperate	Defect
Cooperate	$b-c,\ b-c$	$\frac{b-c}{2},\ \frac{b-c}{2}$
Defect	$\frac{b-c}{2},\ \frac{b-c}{2}$	$0,\ 0$

Table B.2: An example of the Prisonner's Dilemma when agents are teammates. (C, C) is the unique Nash Equilibrium.

subject to j's strategy. This is the same derivation as the expected value of cooperate in Chapter 4.

$$\mathbb{E}(C, \sigma_T) = \nu \left[\sigma_{ji}(b-c) + (1-\sigma_{ji})\frac{b-c}{2} \right] + (1-\nu)\left[\sigma_{jj}(b-c) + (1-\sigma_{jj}) - c\right] \quad \text{(B.1)}$$

$$\mathbb{E}(C, \sigma_T) = \nu \left[\frac{2\sigma_{ji}(b-c)}{2} + \frac{b-c}{2} - \frac{\sigma_{ji}(b-c)}{2} \right] + (1-\nu)\left[\sigma_{jj}b - \sigma_{jj}c - c + \sigma_{jj}c\right] \quad \text{(B.2)}$$

$$\mathbb{E}(C, \sigma_T) = \nu \left[\frac{\sigma_{ji}b - \sigma_{ji}c}{2} + \frac{b-c}{2} \right] + (1-\nu)\left[\sigma_{jj}b - c\right] \quad \text{(B.3)}$$

$$\mathbb{E}(C, \sigma_T) = \nu \left[\frac{(b-c)(\sigma_{ji}+1)}{2} \right] + (1-\nu)\left[\sigma_{jj}b - c\right] \quad \text{(B.4)}$$

$$\mathbb{E}(C, \sigma_T) = \frac{\nu(b-c)(\sigma_{ji}+1)}{2} + (1-\nu)(\sigma_{jj}b - c) \quad \text{(B.5)}$$

Now we show the derivation for a team-focused agent i's expected utility for choosing

D subject to j's strategy. This is also the same as in Chapter 4.

$$\mathbb{E}(D, \sigma_T) = \nu \left[\sigma_{ji} \frac{(b-c)}{2} \right] + (1-\nu) \left[\sigma_{jj} b \right] \tag{B.6}$$

$$\mathbb{E}(D, \sigma_T) = \frac{\nu \sigma_{ji}(b-c)}{2} + (1-\nu)\sigma_{jj} b \tag{B.7}$$

The terms for playing defection with a counterpart who mutually defects $(1 - \sigma_{ji})$ is zero, and therefore omitted above. Next, we show how the final equilibrium is derived using our parameters which define credo.

B.1.2 Equilibrium with Credo

The 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:

$$\begin{aligned} &\psi \mathbb{E}(C, \sigma_T)_{\mathcal{I}} + \phi \mathbb{E}(C, \sigma_T)_{\mathcal{T}} + \omega \mathbb{E}(C, \sigma_T)_{\mathcal{S}} \geq \\ &\quad \psi \mathbb{E}(D, \sigma_T)_{\mathcal{I}} + \phi \mathbb{E}(D, \sigma_T)_{\mathcal{T}} + \omega \mathbb{E}(D, \sigma_T)_{\mathcal{S}}. \end{aligned}$$

Expanding each term with the derivations above (and those for self- and system-focus), we get:

$$\begin{aligned} &\psi_i \left[\sigma_{jj}(b-c) + (1-\sigma_{jj})(-c) \right] + \\ &\phi_i \left[\frac{\nu(b-c)(\sigma_{ji}+1)}{2} + (1-\nu)(\sigma_{jj} b - c) \right] + \\ &\omega_i \left[\sigma_{ji}(b-c) + (1-\sigma_{ji}) \left(\frac{b-c}{2} \right) \right] \geq \psi_i \left[\sigma_{jj}(b) \right] + \\ &\phi_i \left[\frac{\nu \sigma_{ji}(b-c)}{2} + (1-\nu)\sigma_{jj} b \right] + \omega_i \left[\sigma_{ji} \left(\frac{b-c}{2} \right) \right]. \end{aligned} \tag{B.8}$$

Note that the opponent strategies are always σ_{jj} for the self-focus term (no common inter-

est) and σ_{ji} for the system-focus term (all common interest). We expand and simplify:

$$\psi_i \left[\sigma_{jj} b - \sigma_{jj} c - c + \sigma_{jj} c\right] + \phi_i \left[\frac{\nu(b-c)(\sigma_{ji}+1)}{2} - c + \nu c\right] + \omega_i \left[\sigma_{ji}(b-c) + \frac{b-c}{2} - \sigma_{ji}\left(\frac{b-c}{2}\right)\right] \geq \psi_i \left[\sigma_{jj}(b)\right] + \phi_i \left[\frac{\nu \sigma_{ji}(b-c)}{2}\right] + \omega_i \left[\sigma_{ji}\left(\frac{b-c}{2}\right)\right]. \tag{B.9}$$

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

$$\psi_i \left[-c\right] + \phi_i \left[\frac{\nu(b-c)}{2} - c + \nu c\right] + \omega_i \left[\sigma_{ji}(b-c) + \frac{b-c}{2} - \sigma_{ji}(b-c)\right] \geq 0, \tag{B.10}$$

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

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

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

$$-\psi_i c + \phi_i \left[\nu - \frac{2c}{b+c}\right] + \omega_i \left[\frac{b-c}{2}\right] \geq 0 \tag{B.13}$$

$$\phi_i \left(\nu - \frac{2c}{b+c}\right) + \omega_i \left(\frac{b-c}{2}\right) \geq \psi_i c \tag{B.14}$$

This last step represents the final derivation shown as Equation 5.2 in Chapter 5. This equilibrium signifies the conditions under which an agent has more incentive to cooperate than to defect.

Appendix C

Information Sparsity

C.1 Information with Teams

A fixed behavior policy π_i induces a stationary visitation distribution for agent i over states and state-action pairs, denoted as $d^{\pi_i}(s)$ and $d^{\pi_i}(s,a)$ respectively. Since we are concerned with the progression of how agents learn, our theory assumes agents are initialized with random policies that cover the state space uniformly, consistent with past work [9].

The value of $\text{var}\left[\mathcal{I}^{\pi_i}_{S_i,A_i}(Z_{T_i})\right]$ depends on calculating the KL Divergence for state-action pairs from the distribution of states and actions for π_i, d^{π_i}. Given the distributional support $\mathcal{X}_{s_i,a_i}$ (the distribution of team rewards conditioned on specific state-action pairs that are not mapped to zero), this can be expanded to be:

$$\text{var}\left[\mathcal{I}^{\pi_i}_{S_i,A_i}(Z_{T_i})\right] = \text{var}_{s_i,a_i\sim d^{\pi_i}}\left[\sum_{Z_{T_i}\in\mathcal{X}_{s_i,a_i}} p(Z_{T_i}|s_i,a_i)\log\left(\frac{p(Z_{T_i}|s_i,a_i)}{p(Z_{T_i}|s_i)}\right)\right] \tag{C.1}$$

Note that S_i and A_i are based on agent i's individual observations and policy, but Z_{T_i} is based on their shared team reward.