INFLUENCING LONG-TERM BEHAVIOR IN MULTIAGENT REINFORCEMENT LEARNING

Dong-Ki Kim[1,3] **Matthew Riemer**[2,3,4] **Miao Liu**[2,3] **Jakob N. Foerster**[5]
Michael Everett[1] **Chuangchuang Sun**[1,3] **Gerald Tesauro**[2,3] **Jonathan P. How**[1,3]
[1]MIT-LIDS [2]IBM-Research [3]MIT-IBM Watson AI Lab [4]Mila [5]University of Oxford
{dkkim93,mfe,ccsun1,jhow}@mit.edu
{mdriemer,miao.liu1,gtesauro}@us.ibm.com
{jakob.foerster}@eng.ox.ac.uk

ABSTRACT

The main challenge of multiagent reinforcement learning is the difficulty of learning useful policies in the presence of other simultaneously learning agents whose changing behaviors jointly affect the environment's transition and reward dynamics. An effective approach that has recently emerged for addressing this non-stationarity is for each agent to anticipate the learning of other interacting agents and influence the evolution of their future policies towards desirable behavior for its own benefit. Unfortunately, all previous approaches for achieving this suffer from myopic evaluation, considering only a few or a finite number of updates to the policies of other agents. In this paper, we propose a principled framework for considering the limiting policies of other agents as the time approaches infinity. Specifically, we develop a new optimization objective that maximizes each agent's average reward by directly accounting for the impact of its behavior on the limiting set of policies that other agents will take on. Thanks to our farsighted evaluation, we demonstrate better long-term performance than state-of-the-art baselines in various domains, including the full spectrum of general-sum, competitive, and cooperative settings.

1 INTRODUCTION

Learning in multiagent reinforcement learning (MARL) is fundamentally difficult because an agent interacts with other simultaneously learning agents in a shared environment (Buşoniu et al., 2010). The joint learning of agents induces non-stationary environment dynamics from the perspective of each agent, requiring an agent to adapt its behavior with respect to potentially unknown changes in the policies of other interacting agents (Papoudakis et al., 2019). Notably, non-stationary policies will converge to steady-state behaviors by the end of learning in which agents alternate through a recurrent set of joint policies. In practice, this converged joint policy can correspond to a game-theoretic solution concept, such as a Nash equilibrium (Nash, 1950) or more generally a cyclic correlated equilibrium (Zinkevich et al., 2006), but this convergence relies on all agents behaving and updating their policies rationally. Indeed, even when agents do act rationally, multiple equilibria can exist for a single game with some of these Pareto dominating others (Nowé et al., 2012). Hence, a critical question in addressing this non-stationarity is how individual agents should behave to influence convergence of the recurrent set of joint policies towards desirable steady-state behavior.

Our key idea in answering this question is to consider the *limiting policies of other agents* as time approaches infinity. Specifically, the converged behavior of this dynamic multiagent system is not due to some arbitrary stochastic processes, but rather each agent's underlying learning process, which also depends on behaviors of other interacting agents. As such, effective agents should model how their actions affect each other's limiting behaviors and leverage these dependencies to converge jointly to a preferred equilibrium. This farsighted perspective contrasts with other related works that also consider the learning of other agents (Foerster et al., 2018a; Letcher et al., 2019; Xie et al., 2020; Kim et al., 2021; Wang et al., 2021). While those frameworks show improved adaptation performance over methods that neglect the learning of other agents (Lowe et al., 2017; Foerster et al., 2018b; Iqbal & Sha, 2019), they suffer from *myopic evaluation*: an agent only considers a few anticipated updates to the policies of other agents or optimizes for the discounted return, which only can consider a finite

horizon time of $1/(1-\gamma)$ with the discount factor γ (Kearns & Singh, 2002). As a result of this myopic view, we find that these methods often converge to an undesirable joint policy. However, it is non-trivial to achieve farsighted evaluation in practice as increasing the number of anticipated updates to the limit or mixing time is computationally intractable and setting the discount factor $\gamma \to 1$ results in unstable learning (Naik et al., 2019).

Our contribution. With this insight, we provide a principled MARL framework, FUlly Reinforcing acTive influence witH averagE Reward (FURTHER), that considers an agent's impact on limiting policies of other agents for improved converged performance. Specifically, we first introduce an active Markov game formulation that extends the standard Markov game (Littman, 1994) by modeling each agent's impact on the future policies of other agents. Indeed, this setting is more general than the standard formulation and addresses a large portion of realistic settings where it is unreasonable to assume that other agents will keep their policies fixed or even ever truly stop updating their policies in the presence of new information. We then develop a novel optimization objective that for the first time maximizes an agent's average reward per step within this new active Markov game setting. By having each agent optimize this objective, we demonstrate the following benefits of our framework:

- **Converged performance.** We evaluate FURTHER across various general-sum, competitive, and cooperative domains and settings, including self-play. We demonstrate that our method consistently converges to a more desirable equilibrium than baseline methods that either neglect the learning of others (Iqbal & Sha, 2019) or consider their learning with a myopic perspective (Xie et al., 2020).
- **Decentralized learning.** Several prior methods require white-box access to the learning algorithm of other agents for adaptation (Foerster et al., 2018a; Letcher et al., 2019; Kim et al., 2021). Instead, FURTHER employs variational inference for predicting both the unknown policies and learning dynamics of other agents, providing a practical method for learning policies in a decentralized manner.

2 Problem Statement: Active Markov Game

This work studies a general-sum and decentralized multiagent learning setting, where each agent interacts with other independently learning agents in a shared environment. In MARL, each agent updates its policy from recent experiences affected by the joint actions. As such, while an agent cannot directly modify the future policies of other interacting agents, the agent can actively impact them by changing its own actions. We formalize the presence of this causal influence in multiagent interactions by introducing a new framework that formalizes the notion of an active Markov game.

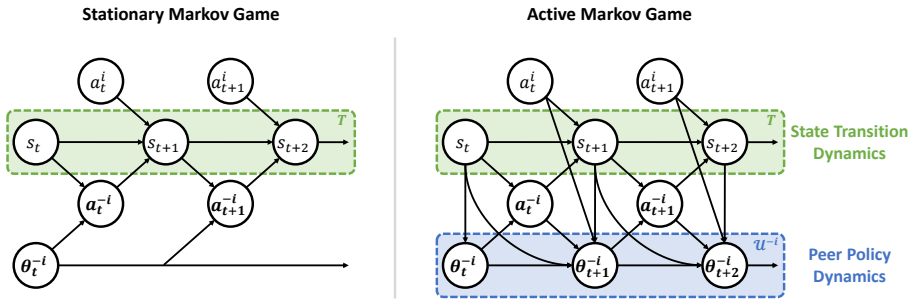


Figure 1: **(Stationary Markov game)** Each agent i assumes that other agents $-i$ have stationary policies in the future. **(Active Markov game)** Each agent i considers that other agents have non-stationary policies in which their policies are updated leveraging a Markovian policy update function.

Active Markov game definition. We define an active Markov game as a tuple $\langle \boldsymbol{\mathcal{I}}, \mathcal{S}, \boldsymbol{\mathcal{A}}, \mathcal{T}, \boldsymbol{\mathcal{R}}, \boldsymbol{\Theta}, \boldsymbol{\mathcal{U}} \rangle$; $\boldsymbol{\mathcal{I}} = \{1, \ldots, n\}$ is the set of n agents; $\mathcal{S}$ is the state space; $\boldsymbol{\mathcal{A}} = \times_{i \in \mathcal{I}} \mathcal{A}^i$ is the set of action spaces for each agent; $\mathcal{T} : \mathcal{S} \times \boldsymbol{\mathcal{A}} \mapsto \mathcal{S}$ is the state transition function; $\boldsymbol{\mathcal{R}} = \times_{i \in \mathcal{I}} \mathcal{R}^i$ is the set of reward functions; $\boldsymbol{\Theta} = \times_{i \in \mathcal{I}} \Theta^i$ is the set of policy parameter spaces for each agent; and $\boldsymbol{\mathcal{U}} = \times_{i \in \mathcal{I}} \mathcal{U}^i$ is the set of policy update functions for each agent. We typeset sets in bold for clarity. Compared to the stationary Markov game that effectively represents MARL with a stationary opponent assumption, the active Markov game considers how other agents' underlying policies change over time (see Figure 1). Specifically, at each timestep t, each agent i executes an

action at a current state $s_t \in \mathcal{S}$ and current policy parameters of other agents $\boldsymbol{\theta}_t^{-i} \in \boldsymbol{\Theta}^{-i}$ according to its stochastic policy $a_t^i \sim \pi^i(\cdot|s_t, \boldsymbol{\theta}_t^{-i}; \theta_t^i)$ parameterized by θ_t^i, where the notation $-i$ indicates all other agents except i. A joint action $\boldsymbol{a_t} = \{a_t^i, \boldsymbol{a_t^{-i}}\}$ yields a transition from s_t to s_{t+1} with probability $\mathcal{T}(s_{t+1}|s_t, a_t^i, \boldsymbol{a_t^{-i}})$. Agent i then obtains a reward according to its reward function $r_t^i = \mathcal{R}^i(s_t, a_t^i, \boldsymbol{a_t^{-i}})$ and considers how policy parameters of other agents will be updated according to probability $\boldsymbol{\mathcal{U}}^{-i}(\boldsymbol{\theta_{t+1}^{-i}}|\boldsymbol{\theta_t^{-i}}, s_t, a_t^i, \boldsymbol{a_t^{-i}}, \boldsymbol{r_t^{-i}}, s_{t+1})$. Importantly, $\boldsymbol{\mathcal{U}}^{-i}$ are a function of a_t^i, which affects the state transition and rewards of other agents. Therefore, i can actively influence their future policies by changing its own behavior. Modeling this influence rather than ignoring it is the main advantage compared to the stationary Markov game formalism.

3 Active Average Reward MARL

The active Markov game provides a principled framework for each agent to model its impact on future policies of other agents. In this section, we develop a new multiagent optimization objective by integrating the average reward formulation (Puterman, 1994; Sutton & Barto, 2018) with the active Markov game framework to maximize the agent's average reward per step while considering its influence on the limiting behaviors of others. We first outline our new objective and derive its policy gradient. We then detail our model-free implementation that builds on top of soft actor-critic (Haarnoja et al., 2018) to learn policies that efficiently optimize for the average reward objective. Our implementation also employs variational inference (Blei et al., 2017) to predict the hidden policies and policy dynamics of other agents for partially observable settings, enabling each agent to select actions and learn its policy in a decentralized manner.

3.1 Formulation of Active Average Reward MARL

Our key finding is that the average reward formulation, developed for single-agent learning (Puterman, 1994), synergizes with our goal of considering limiting behaviors of other interacting agents in multi-agent learning. In particular, the average reward formulation maximizes the agent's average reward per step with equal weight given to immediate and delayed rewards, unlike the discounted return objective. In MARL, once the joint policy arrives at a steady-state behavior, rewards experienced by this recurrent set of policies govern each agent's average reward under the limit of time. Thus, optimizing the average reward in an active Markov game encourages agents to consider their impact on the limiting policies of other agents, as we detail below.

Definition. (Active Average Reward Objective). *Each agent i optimizes its policy parameters θ^i to maximize its expected long-term average reward $\rho_{\theta^i}^i \in \mathbb{R}$ at a state s and policy parameters of other agents $\boldsymbol{\theta}^{-i}$ by considering their Markovian policy update functions $\boldsymbol{\mathcal{U}}^{-i}$:*

$$\max_{\theta^i} \rho_{\theta^i}^i(s, \boldsymbol{\theta}^{-i}) := \max_{\theta^i} \lim_{T\to\infty} \mathbb{E}\Big[\frac{1}{T}\sum_{t=0}^{T} \mathcal{R}^i(s_t, a_t^i, \boldsymbol{a_t^{-i}}) \Big| \begin{smallmatrix} s_0=s,\, \boldsymbol{\theta_0^{-i}}=\boldsymbol{\theta^{-i}}, \\ a_{0:T}^i \sim \pi(\cdot|s_{0:T}, \boldsymbol{\theta_{0:T}^{-i}}; \theta^i), \\ \boldsymbol{a_{0:T}^{-i}} \sim \pi(\cdot|s_{0:T}; \boldsymbol{\theta_{0:T}^{-i}}) \end{smallmatrix}\Big], \tag{1}$$

where T denotes the time horizon. As $T \to \infty$, agent i maximizes its average reward while accounting for the limiting behavior of others $\boldsymbol{\theta_\infty^{-i}}$. We represent $\boldsymbol{\theta_\infty^{-i}}$ as steady-state behavior of other agents under the limit of time, in which they alternate through a recurrent set of policies. As such, the combination of policies and update functions induces a particular type of non-stationary policy, called a cyclic policy (Zinkevich et al., 2006). As a result, our average reward objective effectively considers the converged behaviors of others at a cyclic correlated equilibrium (Zinkevich et al., 2006) under the assumption that agents behave and update their policies rationally. It is important to note that this is a strict generalization of the standard view of MARL: convergence to a fixed point is indeed a recurrent set of size one and Nash equilibria convergence is a special case of a cyclic correlated equilibria.

The active average reward objective in Equation (1) contrasts with an alternative objective of maximizing the discounted return $v_{\theta^i}^i$ in the active Markov game:

$$\max_{\theta^i} v_{\theta^i}^i(s, \boldsymbol{\theta}^{-i}) := \max_{\theta^i} \mathbb{E}\Big[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}^i(s_t, a_t^i, \boldsymbol{a_t^{-i}}) \Big| \begin{smallmatrix} s_0=s,\, \boldsymbol{\theta_0^{i}}=\boldsymbol{\theta^{i}}, \\ a_{0:T}^i \sim \pi(\cdot|s_{0:T}, \boldsymbol{\theta_{0:T}^{-i}}; \theta^i), \\ \boldsymbol{a_{0:T}^{-i}} \sim \pi(\cdot|s_{0:T}; \boldsymbol{\theta_{0:T}^{i}}) \end{smallmatrix}\Big]. \tag{2}$$

As pointed out in Naik et al. (2019), the discounted return objective is bounded by γ, and a learned policy by optimizing the discounted value generally does not correspond to a policy that maximizes

the average reward. Further, $\gamma \to 1$ causes increasingly unstable learning (see Section 5), resulting in the average reward formulation being a more desirable objective to optimize (Nota & Thomas, 2020).

To derive a policy gradient that optimizes Equation (1), we must account for the underlying structure of the state transition $\mathcal{T}$ and policy dynamics $\boldsymbol{\mathcal{U}}$ because the average reward generally depends on the initial state and joint policy. Regarding the state transition $\mathcal{T}$, we follow the single-agent average reward literature (Mahadevan, 1996; Wei et al., 2019; Wan et al., 2021) and assume communicating states: for every pair of states, there exists a joint policy that transitions from one state to the other state in a finite number of steps with non-zero probability. Regarding the policy dynamics $\boldsymbol{\mathcal{U}}$, which is a unique factor for multiagent learning settings, we focus on the unichain interactions:

Assumption. (Unichain policy dynamics). *The policy dynamics of other agents $\boldsymbol{\mathcal{U}}^{-i}$ corresponding to every agent i's policy contain a single recurrent class of policies (i.e., policies of other agents that are visited infinitely often) and a possibly empty set of transient policies (i.e., policies of other agents that are visited only finitely often).*

We note that this unichain assumption is valid for many cases of interest to MARL, including when the policies of other agents satisfy the Greedy in the Limit with Infinite Exploration (GLIE) property generally needed for RL algorithms to provably converge (Sutton & Barto, 2018): 1) all state-action pairs are visited infinitely often and 2) as $t \to \infty$, the behavior policy converges to the greedy policy. More generally, a broad class of noisy update functions can lead to a notion of *stochastic stability* (Foster & Young, 1990; Freidlin et al., 2012; Chasparis, 2019), where multiagent learning with the perturbed learning dynamics has a unique stationary distribution. We refer to Appendix A for a more in-depth discussion of this assumption as well as an analysis of a multi-chain case. A convenient result under these assumptions is that the average reward becomes independent of the initial state and latent strategies (Puterman, 1994):

$$\rho^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}) = \rho^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}}) = \rho^i_{\theta^i} \quad \forall s \neq s', \boldsymbol{\theta^{-i}} \neq \boldsymbol{\theta^{-i\prime}}. \tag{3}$$

Having defined the underlying structure of $\mathcal{T}$ and $\boldsymbol{\mathcal{U}}$, we now derive the Bellman equation in the active Markov game that defines the relationship between the value function and average reward.

Proposition 1. (Active Differential Bellman Equation). *The differential value function $v^i_{\theta^i}$ (Sutton & Barto, 2018) represents the expected total difference between the accumulated rewards from s and $\boldsymbol{\theta^{-i}}$ and the average reward $\rho^i_{\theta^i}$. The differential value function inherently includes the recursive relationship with respect to $v^i_{\theta^i}$ at the next state s' and the updated policies of other agents $\boldsymbol{\theta^{-i\prime}}$:*

$$\begin{aligned} v^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}) &= \lim_{T\to\infty} \mathbb{E}\Big[\sum_{t=0}^{T} \big(\mathcal{R}^i(s_t, a^i_t, \boldsymbol{a^{-i}_t}) - \rho^i_{\theta^i}\big) \Big| \substack{s_0=s,\, \boldsymbol{\theta^{-i}_0}=\boldsymbol{\theta^{-i}}, \\ a^i_{0:T}\sim\pi(\cdot|s_{0:T}, \boldsymbol{\theta^{-i}_{0:T}}; \theta^i), \\ \boldsymbol{a^{-i}_{0:T}}\sim\pi(\cdot|s_{0:T}; \boldsymbol{\theta^{-i}_{0:T}})} \Big] \\ &= \sum_{a^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{\theta^{-i\prime}}} \boldsymbol{\mathcal{U}}^{-i}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, \boldsymbol{r^{-i}}, s') \\ &\quad \Big[\mathcal{R}^i(s, a^i, \boldsymbol{a^{-i}}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}})\Big]. \end{aligned} \tag{4}$$

Proof. See Appendix B for a derivation. □

Finally, we derive the policy gradient based on the differential Bellman equation in Equation (4):

Proposition 2. (Active Average Reward Policy Gradient Theorem). *The gradient of active average reward objective in Equation (1) with respect to agent i's policy parameters θ^i is:*

$$\begin{aligned} \nabla_{\theta^i} J^i_\pi(\theta^i) &= \sum_{s, \boldsymbol{\theta^{-i}}} \boldsymbol{\mu^i_{\theta^i}}(s, \boldsymbol{\theta^{-i}}) \sum_{a^i} \nabla_{\theta^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}), \\ \text{with} \quad q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}) &= \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{z^{i\prime}}} \boldsymbol{\mathcal{U}}^{-i}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, \boldsymbol{r^{-i}}, s') \\ &\quad \Big[\mathcal{R}^i(s, a^i, \boldsymbol{a^{-i}}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}})\Big], \end{aligned} \tag{5}$$

where $\boldsymbol{\mu^i_{\theta^i}}$ denotes i's steady distribution under θ^i with respect to s and $\boldsymbol{\theta^{-i}}$.

Proof. See Appendix C for a derivation. □

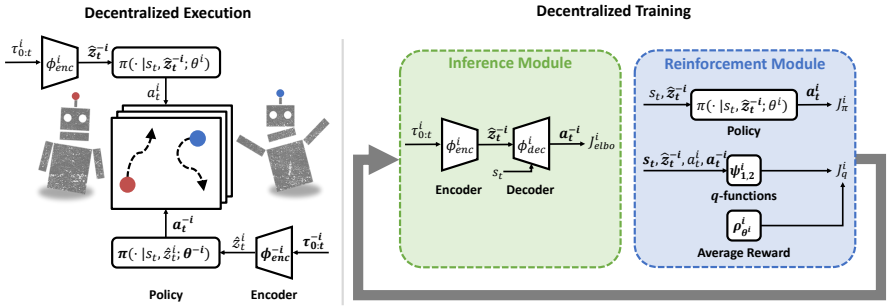


Figure 2: During execution, each agent infers the current hidden policies of other agents from previous interactions and then selects its action. After collecting experiences, each agent updates its inference and reinforcement learning modules in a decentralized manner.

3.2 Practical Implementation of Active Average Reward MARL

Algorithm overview. FURTHER broadly consists of inference and reinforcement learning modules (see Figure 2). In practice, each agent has partial observations about others and cannot directly observe their true policy parameters $\boldsymbol{\theta}^{-i}$ and policy dynamics $\boldsymbol{\mathcal{U}}^{-i}$. The inference learning module predicts this hidden information about other agents via the variational inference (Blei et al., 2017) modified for sequential prediction. The inferred information becomes the input to the reinforcement learning module, which extends the policy gradient theorem in Equation (5) and learns active average reward policies sample efficiently by building on the multiagent soft actor-critic (MASAC) framework for discrete action spaces (Haarnoja et al., 2018; Christodoulou, 2019; Iqbal & Sha, 2019). We note that each agent interacts and learns these modules by only observing the actions of other agents, so our implementation supports decentralized execution and training. We provide further implementation details and psuedocode in Appendix D.

Inference learning module. This module aims to infer the current policies of other agents and their learning dynamics. One approach to achieve this is model-based, where an agent fits an explicit model of the learning strategies of other agents based on the observed data (Kim et al., 2021). However, a model-based approach has difficulties in addressing the infinite recursion problem: if an agent attempts to take into account its opponent's model of the agent itself (Tesauro, 2004). As a result, we study a model-free approach to predict the hidden information of other agents based on an approximate variational inference (Blei et al., 2017). Specifically, we optimise a tractable evidence lower bound (ELBO), defined together with an encoder $p(\hat{\boldsymbol{z}}_{\boldsymbol{t}}^{-\boldsymbol{i}}|\tau_{0:t}^{i};\phi_{\text{enc}}^{i})$ and a decoder $p(\boldsymbol{a}_{\boldsymbol{t}}^{-\boldsymbol{i}}|s_t,\hat{\boldsymbol{z}}_{\boldsymbol{t}}^{-\boldsymbol{i}};\phi_{\text{dec}}^{i})$, parameterised by ϕ_{enc}^{i} and ϕ_{dec}^{i}, respectively:

$$\mathcal{J}_{\text{elbo}}^{i}=\mathbb{E}_{p(\tau_{0:t}^{i}),p(\hat{\boldsymbol{z}}_{\mathbf{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^{i};\phi_{\text{enc}}^{i})}\Big[\sum_{k=0}^{t-1}\underbrace{\log p(\boldsymbol{a}_{\boldsymbol{k}}^{-\boldsymbol{i}}|s_k,\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}};\phi_{\text{dec}}^{i})}_{\text{Reconstruction loss}}-\underbrace{D_{\text{KL}}\big(p(\hat{\boldsymbol{z}}_{\boldsymbol{k+1}}^{-\boldsymbol{i}}|\tau_{0:k}^{i};\phi_{\text{enc}}^{i})||p(\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}})\big)}_{\text{KL divergence}}\Big], \quad (6)$$

where latent strategies $\hat{\boldsymbol{z}}_{\boldsymbol{t}}^{-\boldsymbol{i}}$ represents inferred policy parameters of other agents $\boldsymbol{\theta}_{\boldsymbol{t}}^{-\boldsymbol{i}}$ and $\tau_{0:t}^{i}=\{s_0,a_0^i,\boldsymbol{a}_{\mathbf{0}}^{-\boldsymbol{i}},r_0^i,...,s_t\}$ denotes i's trajectories up to timestep t. We refer to Appendix E for a detailed ELBO derivation. By optimizing the reconstruction term, the encoder aims to infer accurate latent strategies of others. Further, by imposing the prior through the KL divergence, where we set the prior to the previous posterior with initial prior $p(\hat{\boldsymbol{z}}_{\mathbf{0}}^{-\boldsymbol{i}})=\mathcal{N}(0,I)$, the inferred policies from the encoder are encouraged to be sequentially consistent across time (i.e., no abrupt changes in policies of others).

Reinforcement learning module. This module aims to learn a policy that can maximize the agent's average reward based on the inferred information about other agents. Each agent maintains its policy $\pi(\cdot|s,\hat{\boldsymbol{z}}^{-\boldsymbol{i}};\theta^i)$ parameterized by θ^i, two q-functions $q_{\theta^i}^{i}(s,\hat{\boldsymbol{z}}^{-\boldsymbol{i}},a^i,\boldsymbol{a}^{-\boldsymbol{i}};\psi_1^i)$ and $q_{\theta^i}^{i}(s,\hat{\boldsymbol{z}}^{-\boldsymbol{i}},a^i,\boldsymbol{a}^{-\boldsymbol{i}};\psi_2^i)$ parameterized by ψ_1^i,ψ_2^i, and learnable average reward $\rho_{\theta^i}^{i}\in\mathbb{R}$. We train the q-functions and $\rho_{\theta^i}^{i}$ by minimizing the soft Bellman residual:

$$\begin{aligned}J_q^i(\psi_\beta^i,\rho_{\theta^i}^{i})&=\mathbb{E}_{(s,\hat{\boldsymbol{z}}^{-\boldsymbol{i}},a^i,\boldsymbol{a}^{-\boldsymbol{i}},r^i,s',\hat{\boldsymbol{z}}^{-\boldsymbol{i}\prime})\sim\mathcal{D}^i}\Big[\Big(y-q_{\theta^i}^{i}(s,\hat{\boldsymbol{z}}^{-\boldsymbol{i}},a^i,\boldsymbol{a}^{-\boldsymbol{i}};\psi_\beta^i)\Big)^2\Big],\\ \text{with}\quad y&=r^i-\rho_{\theta^i}^{i}+v_{\theta^i}^{i}(s',\hat{\boldsymbol{z}}^{-\boldsymbol{i}\prime};\bar{\psi}_\beta^i),\end{aligned} \quad (7)$$

where $\beta = 1, 2$, $\mathcal{D}^i$ denotes i's replay buffer, and $\bar{\psi}^i_\beta$ denotes the target q-network parameters. The soft value function $v^i_{\theta^i}$ calculates the state value with the policy entropy $\mathcal{H}$ and entropy weight α:

$$v^i_{\theta^i}(s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}; \psi^i) = \sum_{a^i} \pi(\cdot|s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}; \theta^i) \sum_{\boldsymbol{a}^{-\boldsymbol{i}}} \pi(\cdot|s; \hat{\boldsymbol{z}}^{-\boldsymbol{i}}) \min_{\beta=1,2} q^i_{\theta^i}(s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}, a^i, \boldsymbol{a}^{-\boldsymbol{i}}; \psi^i_\beta) + \alpha \mathcal{H}(\pi(\cdot|s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}; \theta^i)). \quad (8)$$

Finally, the policy is trained to maximize:

$$J^i_\pi(\theta^i) = \mathbb{E}_{(s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}, \boldsymbol{a}^{-\boldsymbol{i}}) \sim \mathcal{D}^i} \Big[\sum_{a^i} \pi(\cdot|s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}; \theta^i) \min_{\beta=1,2} q^i_{\theta^i}(s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}, a^i, \boldsymbol{a}^{-\boldsymbol{i}}; \psi^i_\beta) + \alpha \mathcal{H}(\pi(\cdot|s, \hat{\boldsymbol{z}}^{-\boldsymbol{i}}; \theta^i)) \Big]. \quad (9)$$

4 Related Work

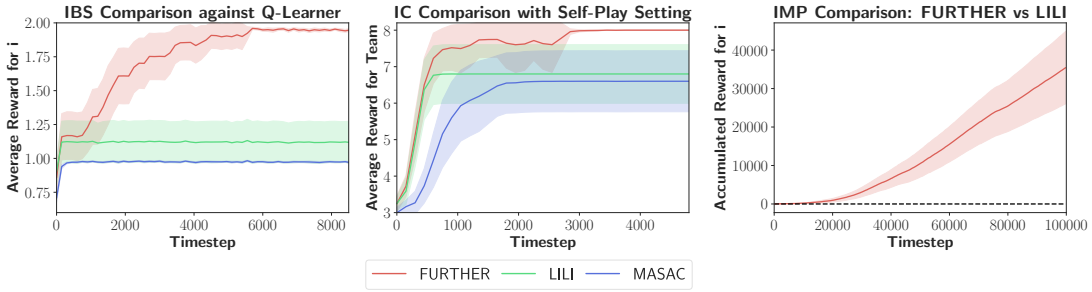
Stationary MARL. The standard approach for addressing the non-stationarity problem in MARL is to consider information about other agents and reason about joint action effects (Hernandez-Leal et al., 2017). Example frameworks include the studies regarding centralized training with decentralized execution, which account for other agents' actions through centralized critics (Lowe et al., 2017; Foerster et al., 2018b; Yang et al., 2018; Omidshafiei et al., 2019; Iqbal & Sha, 2019; Kim et al., 2020). Other related works are the opponent modeling frameworks that infer opponents' policies and condition an agent's policy on the inferred information about others (He et al., 2016; Raileanu et al., 2018; Grover et al., 2018; Wen et al., 2019). While these works alleviate non-stationarity, each agent learns its policy by assuming that other agents will have *stationary* policies in the future. This assumption is incorrect because other agents can have different behaviors in the future due to their learning (Foerster et al., 2018a), resulting in improper adaptation with respect to their changing behaviors. In contrast, FURTHER models the learning processes of other agents and considers how to actively influence their limiting behaviors.

Learning-aware MARL. Our framework is closely related to prior works that consider the learning of other agents in the environment. The framework by Zhang & Lesser (2010), for instance, learns the best response adaptation to the other agent's anticipated updated policy. Notably, LOLA (Foerster et al., 2018a) and its more recent improvements (Foerster et al., 2018c; Letcher et al., 2019) study the impact of behavior on one or a few of another agent's policy updates. Our work is also related to frameworks that leverage the inferred policy dynamics of other agents to impact their future policies by maximizing the discounted return objective (Jaques et al., 2019; Xie et al., 2020; Wang et al., 2021). Lastly, meta-learning frameworks are related that directly account for the non-stationary policy dynamics in multiagent settings based on the inner-loop and outer-loop optimization (Al-Shedivat et al., 2018; Kim et al., 2021; Balaguer et al., 2022). However, all of these approaches only account for a finite number of updates to the policies of other agents, so we observe that these methods can converge to a less desirable equilibrium. FURTHER addresses this issue in these related methods by optimizing for the average reward objective in a novel active Markov game setting.

Game-theoretic MARL. Another effective approach to addressing the non-stationarity is learning equilibrium policies that correspond to game-theoretic solution concepts (Littman, 1994; 2001; Wang & Sandholm, 2002; Greenwald & Hall, 2003; Zinkevich et al., 2006). These frameworks predict stationary joint action values by the end of learning and can guarantee convergence to Nash (Nash, 1950) or correlated (Aumann, 1987) equilibrium values under certain assumptions. However, as noted in Bowling (2005), this convergence is guaranteed only while ignoring the actual learning dynamics of other agents, and each agent assumes all agents will play the same joint equilibrium strategy. As such, equilibrium learners can fail to learn best-response policies when other agents choose to play different equilibrium strategies in the future as a result of their learning. By contrast, FURTHER considers convergence to a recurrent set of joint policies by inferring the true policy dynamics of other agents. We note that this recurrent set of joint policies can arrive at more general game theoretic concepts than Nash equilibria such as cyclic correlated equilibria (Zinkevich et al., 2006) in the case that all agents behave and update their policies rationally.

5 Evaluation

We demonstrate FURTHER's efficacy on a diverse suite of domains, including general-sum, competitive, and cooperative settings. We refer to appendix F for hyperparameters used in experiments. Each figure shows the mean and 95% confidence interval computed across 20 random seeds.



(a) (b) (c)

Figure 3: **(a)** Convergence in a general-sum game of IBS. The FURTHER agent achieves convergence to its optimal pure strategy Nash equilibrium. **(b)** Convergence in a cooperative game of IC with self-play. The FURTHER team shows better converged performance than baselines. **(c)** A competitive play between FURTHER and LILI in IMP. FURTHER receives higher rewards than LILI over time.

Baselines. We compare our method with the following baseline adaptation strategies:

• **LILI (Xie et al., 2020):** An approach that considers the learning dynamics of other agents but suffers from myopic evaluation bias by optimizing the discounted return objective in Equation (2).

• **MASAC (Iqbal & Sha, 2019):** An approach that extends SAC (Haarnoja et al., 2018) to a multiagent learning setting by having centralized critics (Lowe et al., 2017). This baseline assumes other agents will have stationary policies in the future and thus neglects their learning.

Question 1. *How do methods perform when playing against a q-learning agent?*

We consider playing the iterated Bach or Stravinsky game (IBS; see Table 1). This general-sum game involves conflicting elements with two pure strategy Nash equilibria, where convergence to (A,A) and (B,B) equilibrium are more preferable from i's and j's perspective, respectively. Suppose agent i plays against a naive learner j, such as q-learner (Watkins & Dayan, 1992), whose initial q-values are set to prefer action (B). In this experimental setting, it is ideal for agent i to change j's influence behavior to select (A) such that they converge to i's optimal pure strategy Nash equilibrium of (A,A). As in Foerster et al. (2018a), we model the state space as $s_0 = \varnothing$ and $s_t = \boldsymbol{a_{t-1}}$ for $t \geq 1$.

		Agent j	
		B	S
Agent i	B	$(2,1)$	$(0,0)$
	S	$(0,0)$	$(1,2)$

Table 1: Bach or Stravinsky game payoff matrix.

The average reward performance when an agent i, trained with either FURTHER or the baseline methods, interacts with the q-learner j is shown in Figure 3a. There are two notable observations. First, the FURTHER agent i consistently converges to its optimal equilibrium of (A,A), while the LILI agent often converges to the sub-optimal equilibrium of (B,B). The FURTHER agent i learns to select (A) while j selects (B), receive the worst rewards of zero, and wait until j's q-value for (B) is updated to be lower than the q-value for (A). With the limiting view, i learns that the waiting process is only temporary, and receiving the eventual rewards of two by converging to (A,A) is optimal. By contrast, LILI suffers from myopic evaluation and shows decreased convergence performance because the agent prefers simply converging to the sub-optimal equilibrium rather than waiting indefinitely. Second, FURTHER and LILI outperform the other approach of MASAC, showing the benefit of considering active influence on future policies of other agents.

Question 2. *Which equilibrium do methods converge to when there are multiple equilibria in a self-play setting?*

We now experiment with a self-play setting, where all agents learn with the same algorithm. We consider evaluation in an iterated cooperative (IC) game with identical payoffs (see Table 2). Note that this game has two pure strategy Nash equilibria of (A,A) and (B,B), in which the (B,B) equilibrium Pareto dominates the other. For the experimental setting, we compare when both agents are trained with either ours or the baseline methods. Figure 3b shows the average reward performance as the train iteration increases. Similar to the IBS results, we observe

		Agent j	
		A	B
Agent i	A	$(4,4)$	$(0,0)$
	B	$(0,0)$	$(8,8)$

Table 2: Cooperative game payoff matrix.

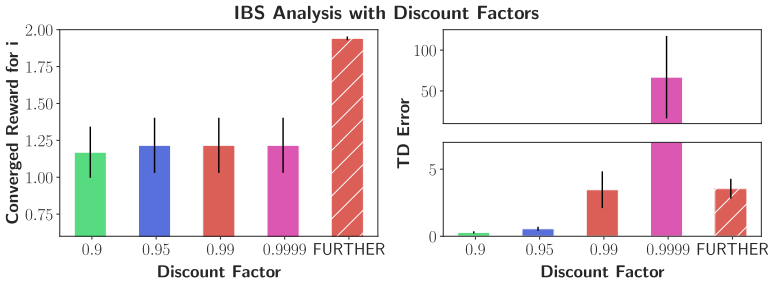


Figure 4: Convergence performance and corresponding TD errors with varying γ in LILI when agent i interacts with a q-learner j.

that FURTHER consistently converges to the best equilibrium of (B,B) while the baseline methods can converge to the sub-optimal equilibrium, and LILI performs better than MASAC.

Question 3. *How does FURTHER's limiting optimization perform against LILI's myopic discounted optimization?*

To answer this question, we consider the FURTHER agent i directly competing against the LILI opponent j in the iterated matching pennies (IMP) game (see Table 3). We observe that showing the average reward is noisy and hard to interpret in this zero-sum game, so we show an alternative metric of i's accumulated reward summed up to the current timestep. Figure 3c shows that the accumulated reward for i is positive, meaning that FURTHER achieves higher rewards than LILI over time. This result concludes that FURTHER is more effective than LILI by employing the limiting view via the average reward formulation.

Agent i \ Agent j	H	T
H	(1, -1)	(-1, 1)
T	(-1, 1)	(1, -1)

Table 3: Matching pennies game payoff matrix.

Question 4. *Is it beneficial to set the discount factor γ close to one?*

In Section 3, we noted that the discounted return objective does not maximize the average reward and has unstable learning when $\gamma \to 1$ (Naik et al., 2019). We empirically validate this statement in this question. In particular, we evaluate with varying discount factors $\gamma \in \{0.9, 0.95, 0.99, 0.9999\}$ for LILI in the IBS scenario (see Question 1). Figure 4 shows the converged performance at the end of learning and corresponding temporal different (TD) errors in the q updates, respectively. We observe a slight increase in the converged performance when γ increases from 0.9 to higher values, but the performance with the discounted return is still much less than the performance by the average reward objective. Notably, the TD error increases exponentially with higher γ and causes unstable learning, which makes optimizing the average reward objective more desirable than the discounted return.

6 Conclusion

In this paper, we have introduced FURTHER, a principled algorithm to address non-stationarity by considering each agent's impact on the converged policies of other agents. The key idea is to consider the limiting policies of other agents through the average reward formulation in a newly proposed active Markov game framework, and we have developed a practical model-free and decentralized approach to address this setting. We evaluated our method on various multiagent settings and showed that FURTHER consistently converges to more desirable long-term behavior for agents that use it than state-of-the-art baseline approaches.

Acknowledgments

Research funded by IBM (as part of the MIT-IBM Watson AI Lab initiative).

References

Maruan Al-Shedivat, Trapit Bansal, Yura Burda, Ilya Sutskever, Igor Mordatch, and Pieter Abbeel. Continuous adaptation via meta-learning in nonstationary and competitive environments. In

International Conference on Learning Representations, 2018. URL https://openreview.net/forum?id=Sk2u1g-0-.

Robert J. Aumann. Correlated equilibrium as an expression of bayesian rationality. *Econometrica*, 55(1):1–18, 1987. ISSN 00129682, 14680262. URL http://www.jstor.org/stable/1911154.

Jan Balaguer, Raphael Koster, Christopher Summerfield, and Andrea Tacchetti. The good shepherd: An oracle agent for mechanism design. *arXiv preprint arXiv:2202.10135*, 2022.

David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, Apr 2017. ISSN 1537-274X. doi: 10.1080/01621459.2017.1285773. URL http://dx.doi.org/10.1080/01621459.2017.1285773.

Michael Bowling. Convergence and no-regret in multiagent learning. In *Neural Information Processing Systems (NeurIPS)*, pp. 209–216. MIT Press, 2005. URL http://papers.nips.cc/paper/2673-convergence-and-no-regret-in-multiagent-learning.pdf.

Lucian Buşoniu, Robert Babuška, and Bart De Schutter. *Multi-agent Reinforcement Learning: An Overview*, pp. 183–221. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010. ISBN 978-3-642-14435-6. doi: 10.1007/978-3-642-14435-6_7. URL https://doi.org/10.1007/978-3-642-14435-6_7.

Georgios C. Chasparis. Stochastic stability of perturbed learning automata in positive-utility games. *IEEE Transactions on Automatic Control*, 64(11):4454–4469, 2019. doi: 10.1109/TAC.2019.2895300.

Petros Christodoulou. Soft actor-critic for discrete action settings. *CoRR*, abs/1910.07207, 2019. URL http://arxiv.org/abs/1910.07207.

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

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. *Association for the Advancement of Artificial Intelligence (AAAI)*, 32(1), Apr. 2018b. URL https://ojs.aaai.org/index.php/AAAI/article/view/11794.

Jakob Foerster, Gregory Farquhar, Maruan Al-Shedivat, Tim Rocktäschel, Eric Xing, and Shimon Whiteson. DiCE: The infinitely differentiable Monte Carlo estimator. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 1524–1533. PMLR, 10–15 Jul 2018c. URL http://proceedings.mlr.press/v80/foerster18a.html.

Dean Foster and Peyton Young. Stochastic evolutionary game dynamics. *Theoretical Population Biology*, 38(2):219–232, 1990. ISSN 0040-5809. doi: https://doi.org/10.1016/0040-5809(90)90011-J. URL https://www.sciencedirect.com/science/article/pii/004058099090011J.

M.I. Freidlin, J. Szücs, and A.D. Wentzell. *Random Perturbations of Dynamical Systems*. Grundlehren der mathematischen Wissenschaften. Springer, 2012. ISBN 9783642258473. URL http://books.google.de/books?id=p8LFMILAiMEC.

Amy Greenwald and Keith Hall. Correlated-Q learning. In *International Conference on Machine Learning (ICML)*, pp. 242–249. AAAI Press, 2003. ISBN 1577351894.

Aditya Grover, Maruan Al-Shedivat, Jayesh Gupta, Yuri Burda, and Harrison Edwards. Learning policy representations in multiagent systems. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 1802–1811, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/grover18a.html.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 1861–1870. PMLR, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/haarnoja18b.html.

He He, Jordan Boyd-Graber, Kevin Kwok, and Hal Daumé III. Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 48, pp. 1804–1813, 20–22 Jun 2016. URL http://proceedings.mlr.press/v48/he16.html.

Pablo Hernandez-Leal, Michael Kaisers, Tim Baarslag, and Enrique Munoz de Cote. A survey of learning in multiagent environments: Dealing with non-stationarity. *CoRR*, abs/1707.09183, 2017. URL http://arxiv.org/abs/1707.09183.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 97, pp. 2961–2970. PMLR, 09–15 Jun 2019. URL http://proceedings.mlr.press/v97/iqbal19a.html.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Caglar Gulcehre, Pedro Ortega, Dj Strouse, Joel Z. Leibo, and Nando De Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 97, pp. 3040–3049. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/jaques19a.html.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. *Machine learning*, 49(2), 2002.

Dong-Ki Kim, Miao Liu, Shayegan Omidshafiei, Sebastian Lopez-Cot, Matthew Riemer, Golnaz Habibi, Gerald Tesauro, Sami Mourad, Murray Campbell, and Jonathan P. How. Learning hierarchical teaching policies for cooperative agents. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, AAMAS '20, pp. 620–628, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

Dong Ki Kim, Miao Liu, Matthew D Riemer, Chuangchuang Sun, Marwa Abdulhai, Golnaz Habibi, Sebastian Lopez-Cot, Gerald Tesauro, and Jonathan How. A policy gradient algorithm for learning to learn in multiagent reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 5541–5550. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/kim21g.html.

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

Michael L. Littman. Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, ICML'94, pp. 157–163. Morgan Kaufmann Publishers Inc., 1994. ISBN 1-55860-335-2. URL http://dl.acm.org/citation.cfm?id=3091574.3091594.

Michael L. Littman. Friend-or-foe q-learning in general-sum games. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 322–328, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Neural Information Processing Systems (NeurIPS)*, pp. 6382–6393, 2017.

Sridhar Mahadevan. Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Mach. Learn.*, 22(1–3):159–195, jan 1996. ISSN 0885-6125. doi: 10.1007/BF00114727. URL https://doi.org/10.1007/BF00114727.

Abhishek Naik, Roshan Shariff, Niko Yasui, Hengshuai Yao, and Richard S. Sutton. Discounted reinforcement learning is not an optimization problem, 2019.

John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. ISSN 0027-8424. doi: 10.1073/pnas.36.1.48. URL https://www.pnas.org/content/36/1/48.

Chris Nota and Philip S. Thomas. Is the policy gradient a gradient? In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, pp. 939–947, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.

Ann Nowé, Peter Vrancx, and Yann-Michaël De Hauwere. *Game Theory and Multi-agent Reinforcement Learning*, pp. 441–470. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27645-3. doi: 10.1007/978-3-642-27645-3_14. URL https://doi.org/10.1007/978-3-642-27645-3_14.

Shayegan Omidshafiei, Dong-Ki Kim, Miao Liu, Gerald Tesauro, Matthew Riemer, Christopher Amato, Murray Campbell, and Jonathan P. How. Learning to teach in cooperative multiagent reinforcement learning. In *Association for the Advancement of Artificial Intelligence (AAAI)*. AAAI Press, 2019. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.33016128. URL https://doi.org/10.1609/aaai.v33i01.33016128.

Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V. Albrecht. Dealing with non-stationarity in multi-agent deep reinforcement learning. *CoRR*, abs/1906.04737, 2019. URL http://arxiv.org/abs/1906.04737.

Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., USA, 1st edition, 1994. ISBN 0471619779.

Roberta Raileanu, Emily Denton, Arthur Szlam, and Rob Fergus. Modeling others using oneself in multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 4257–4266, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/raileanu18a.html.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018. URL http://incompleteideas.net/book/the-book-2nd.html.

Gerald Tesauro. Extending Q-learning to general adaptive multi-agent systems. In S. Thrun, L. Saul, and B. Schölkopf (eds.), *Neural Information Processing Systems (NeurIPS)*, volume 16. MIT Press, 2004. URL https://proceedings.neurips.cc/paper/2003/file/e71e5cd119bbc5797164fb0cd7fd94a4-Paper.pdf.

Yi Wan, Abhishek Naik, and Richard S Sutton. Learning and planning in average-reward markov decision processes. In *International Conference on Machine Learning (ICML)*, volume 139, pp. 10653–10662. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/wan21a.html.

Woodrow Zhouyuan Wang, Andy Shih, Annie Xie, and Dorsa Sadigh. Influencing towards stable multi-agent interactions. In *Conference on Robot Learning (CoRL)*, 2021. URL https://openreview.net/forum?id=n6xYib0irVR.

Xiaofeng Wang and Tuomas Sandholm. Reinforcement learning to play an optimal nash equilibrium in team markov games. In *Neural Information Processing Systems (NeurIPS)*, pp. 1603–1610. MIT Press, 2002.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. In *Machine Learning*, pp. 279–292, 1992.

Chen-Yu Wei, Mehdi Jafarnia-Jahromi, Haipeng Luo, Hiteshi Sharma, and Rahul Jain. Model-free reinforcement learning in infinite-horizon average-reward markov decision processes. *CoRR*, abs/1910.07072, 2019. URL http://arxiv.org/abs/1910.07072.

Ying Wen, Yaodong Yang, Rui Luo, Jun Wang, and Wei Pan. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rkl6As0cF7.

Annie Xie, Dylan Losey, Ryan Tolsma, Chelsea Finn, and Dorsa Sadigh. Learning latent representations to influence multi-agent interaction. In *Conference on Robot Learning (CoRL)*, 2020.

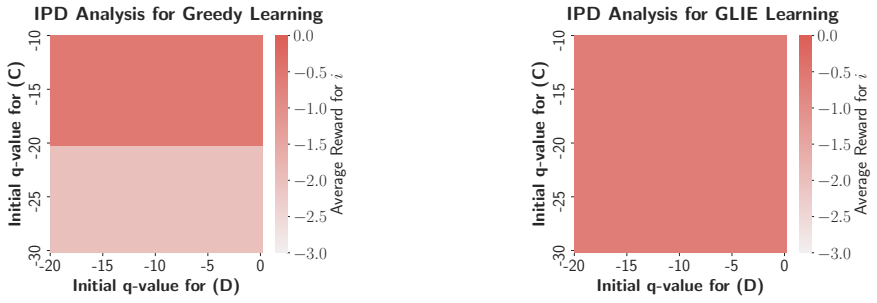
Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 5571–5580, 10–15 Jul 2018. URL http://proceedings.mlr.press/v80/yang18d.html.

Chongjie Zhang and Victor R. Lesser. Multi-agent learning with policy prediction. In *Association for the Advancement of Artificial Intelligence (AAAI)*, 2010.

Martin Zinkevich, Amy Greenwald, and Michael Littman. Cyclic equilibria in markov games. In *Neural Information Processing Systems (NeurIPS)*, volume 18. MIT Press, 2006. URL https://proceedings.neurips.cc/paper/2005/file/9752d873fa71c19dc602bf2a0696f9b5-Paper.pdf.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=Hkl9JlBYvr.

A Discussion: Unichain Policy Dynamics Assumption



(a) (b)

Figure 5: **(a)** A policy iteration analysis in IPD when agent j has a greedy learning algorithm. Depending on $\boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}$, i's possible maximum average reward is affected. **(b)** A policy iteration analysis in IPD when agent j has a GLIE learning algorithm. The possible maximum average reward for agent i is independent to j's initial policy $\boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}$.

In Section 3, we assumed the unichain assumption on policy dynamics, resulting in the average reward independent of initial policies of other agents. We note that this unichain assumption is valid for many cases of interest to MARL, including when the policies of other agents satisfy the Greedy in the Limit with Infinite Exploration (GLIE) property (Sutton & Barto, 2018): 1) all state-action pairs are visited infinitely often and 2) as $t \to \infty$, the behavior policy converges to the greedy policy. In particular, the exploration in GLIE adds noise to the learning dynamics of agents, and agents arrive at the concept of *stochastic stability* (Foster & Young, 1990; Freidlin et al., 2012; Chasparis, 2019), where multiagent learning with the perturbed learning dynamics has a unique stationary distribution. Because most MARL algorithms have exploration in choosing actions, GLIE is satisfied in many practical settings, so the unichain assumption is a reasonable assumption to make.

For example, consider playing the iterated prisoner's dilemma (IPD) game (see Table 4), where agent i plays against a q-learning agent j. We perform a policy iteration analysis (Puterman, 1994) with respect to j's varying initial q-values for each action $\boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}$. Figure 5a and Figure 5b show i's maximum average reward with respect to $\boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}$ when j trains with a greedy and GLIE algorithm, respectively. Interestingly, the analysis with the greedy algorithm shows that i's average reward depends on $\boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}$ in IPD, where there is a set of j's initial policies that i can achieve the high average reward, but there is the other set of initial policies that can result in the worst average reward of -2. By contrast, Figure 5b shows that i's average reward is independent of $\boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}$ when j's learning satisfies GLIE.

Agent i \ Agent j	C	D
C	(-1, -1)	(-3, 0)
D	(0, -3)	(-2, -2)

Table 4: Prisoner's dilemma game payoff matrix.

B Derivation of Active Differential Bellman Equation

Proposition 1. (Active Differential Bellman Equation). *The differential value function $v_{\theta^i}^i$ (Sutton & Barto, 2018) represents the expected total difference between the accumulated rewards from s and $\boldsymbol{\theta}^{-\boldsymbol{i}}$ and the average reward $\rho_{\theta^i}^i$. The differential value function inherently includes the recursive relationship with respect to $v_{\theta^i}^i$ at the next state s' and the updated policies of other agents $\boldsymbol{\theta}^{-\boldsymbol{i}\boldsymbol{\prime}}$:*

$$
\begin{aligned}
v_{\theta^i}^i(s, \boldsymbol{\theta}^{-\boldsymbol{i}}) &= \lim_{T\to\infty} \mathbb{E}\Big[\sum_{t=0}^{T} \big(\mathcal{R}^i(s_t, a_t^i, \boldsymbol{a}_{\boldsymbol{t}}^{-\boldsymbol{i}}) - \rho_{\theta^i}^i\big) \,\Big|\, \substack{s_0=s,\, \boldsymbol{\theta}_{\mathbf{0}}^{-\boldsymbol{i}}=\boldsymbol{\theta}^{-\boldsymbol{i}}, \\ a_{0:T}^i \sim \pi(\cdot|s_{0:T}, \boldsymbol{\theta}_{\mathbf{0}:\boldsymbol{T}}^{-\boldsymbol{i}}; \theta^i), \\ \boldsymbol{a}_{\mathbf{0}:\boldsymbol{T}}^{-\boldsymbol{i}} \sim \pi(\cdot|s_{0:T}; \boldsymbol{\theta}_{\mathbf{0}:\boldsymbol{T}}^{-\boldsymbol{i}})} \Big] \\
&= \sum_{a^i} \pi(\cdot|s, \boldsymbol{\theta}^{-\boldsymbol{i}}; \theta^i) \sum_{\boldsymbol{a}^{-\boldsymbol{i}}} \pi(\cdot|s; \boldsymbol{\theta}^{-\boldsymbol{i}}) \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a}^{-\boldsymbol{i}}) \sum_{\boldsymbol{\theta}^{-\boldsymbol{i}\boldsymbol{\prime}}} \boldsymbol{\mathcal{U}}^{-\boldsymbol{i}}(\boldsymbol{\theta}^{-\boldsymbol{i}\boldsymbol{\prime}}|\boldsymbol{\theta}^{-\boldsymbol{i}}, s, a^i, \boldsymbol{a}^{-\boldsymbol{i}}, \boldsymbol{r}^{-\boldsymbol{i}}, s') \\
&\quad \Big[\mathcal{R}^i(s, a^i, \boldsymbol{a}^{-\boldsymbol{i}}) - \rho_{\theta^i}^i + v_{\theta^i}^i(s', \boldsymbol{\theta}^{-\boldsymbol{i}\boldsymbol{\prime}})\Big].
\end{aligned}
$$

Proof. We seek to derive the recursive relationship between $v_{\theta^i}^i(s, \boldsymbol{\theta}^{-\boldsymbol{i}})$ and $v_{\theta^i}^i(s', \boldsymbol{\theta}^{-\boldsymbol{i}\boldsymbol{\prime}})$. We leverage the general derivation outlined in Sutton & Barto (2018) (page 59) and extend it to our active Markov

game formulation:

$$
\begin{aligned}
v^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}) &= \lim_{T\to\infty} \mathbb{E}\Big[\sum_{t=0}^{T} \big(\mathcal{R}^i(s_t, a^i_t, \boldsymbol{a^{-i}_t}) - \rho^i_{\theta^i}\big) \Big| \substack{s_0=s,\, \boldsymbol{\theta^{-i}_0}=\boldsymbol{\theta^{-i}},\\ a^i_{0:T}\sim\pi(\cdot|s_{0:T},\boldsymbol{\theta^i_{0:T}};\theta^i),\\ \boldsymbol{a^{-i}_{0:T}}\sim\pi(\cdot|s_{0:T};\boldsymbol{\theta^{-i}_{0:T}})} \Big] \qquad (10)\\
&= \lim_{T\to\infty} \mathbb{E}\Big[\mathcal{R}^i(s_0, a^i_0, \boldsymbol{a^{-i}_0}) - \rho^i_{\theta^i} + \sum_{t=1}^{T} \big(\mathcal{R}^i(s_t, a^i_t, \boldsymbol{a^{-i}_t}) - \rho^i_{\theta^i}\big) \Big| \substack{s_0=s,\, \boldsymbol{\theta^{-i}_0}=\boldsymbol{\theta^{-i}},\\ a^i_{0:T}\sim\pi(\cdot|s_{0:T},\boldsymbol{\theta^i_{0:T}};\theta^i),\\ \boldsymbol{a^{-i}_{0:T}}\sim\pi(\cdot|s_{0:T};\boldsymbol{\theta^{-i}_{0:T}})} \Big]\\
&= \sum_{a^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{\theta^{-i\prime}}} \boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, \boldsymbol{r^{-i}}, s')\\
&\quad \Big[\mathcal{R}^i(s, a^i, \boldsymbol{a^{-i}}) - \rho^i_{\theta^i} + \lim_{T\to\infty} \mathbb{E}\Big[\sum_{t=0}^{T} \big(\mathcal{R}^i(s_{t+1}, a^i_{t+1}, \boldsymbol{a^{-i}_{t+1}}) - \rho^i_{\theta^i}\big) \Big| \substack{s_1=s',\, \boldsymbol{\theta^{-i}_1}=\boldsymbol{\theta^{-i\prime}},\\ a^i_{1:T}\sim\pi(\cdot|s_{1:T},\boldsymbol{\theta^i_{1:T}};\theta^i),\\ \boldsymbol{a^{-i}_{1:T}}\sim\pi(\cdot|s_{1:T};\boldsymbol{\theta^{-i}_{1:T}})} \Big]\Big]\\
&= \sum_{a^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{\theta^{-i\prime}}} \boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, \boldsymbol{r^{-i}}, s')\\
&\quad \Big[\mathcal{R}^i(s, a^i, \boldsymbol{a^{-i}}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}})\Big].
\end{aligned}
$$

□

C Derivation of Active Average Reward Policy Gradient

Proposition 2. (Active Average Reward Policy Gradient Theorem). *The gradient of active average reward objective in Equation (1) with respect to agent i's policy parameters θ^i is:*

$$
\begin{aligned}
\nabla_{\theta^i} J^i_\pi(\theta^i) &= \sum_{s, \boldsymbol{\theta^{-i}}} \boldsymbol{\mu^i_{\theta^i}}(s, \boldsymbol{\theta^{-i}}) \sum_{a^i} \nabla_{\theta^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}),\\
\text{with} \quad q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}) &= \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{z^{i\prime}}} \boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, \boldsymbol{r^{-i}}, s')\\
&\quad \Big[\mathcal{R}^i(s, a^i, \boldsymbol{a^{-i}}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}})\Big],
\end{aligned}
$$

where $\boldsymbol{\mu^i_{\theta^i}}$ denotes i's steady distribution under θ^i with respect to s and $\boldsymbol{\theta^{-i}}$.

Proof. We seek to derive an expression for optimizing the active average reward objective in Equation (1) with respect to agent i's policy parameters θ^i. Our derivation leverages the general policy gradient theorem proof for the continuing case in Sutton & Barto (2018) (page 334). We begin by expressing the gradient of the differential value function $v^i_{\theta^i}(s, \boldsymbol{\theta^{-i}})$ for any s and $\boldsymbol{\theta^{-i}}$:

$$
\begin{aligned}
\nabla_{\theta^i} v^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}) &= \nabla_{\theta^i} \Big[\sum_{a^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}) \Big]\\
&= \Big[\sum_{a^i} \nabla_{\theta^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}) +\\
&\quad \sum_{a^i} \pi(\cdot|s, \boldsymbol{\theta^{-i}}; \theta^i) \sum_{\boldsymbol{a^{-i}}} \pi(\cdot|s; \boldsymbol{\theta^{-i}}) \underbrace{\nabla_{\theta^i} q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}})}_{\text{Term A}} \Big].
\end{aligned} \qquad (11)
$$

We continue to derive the Term A in Equation (11):

$$
\begin{aligned}
\nabla_{\theta^i} q^i_{\theta^i}(s, \boldsymbol{\theta^{-i}}, a^i, \boldsymbol{a^{-i}}) &= \nabla_{\theta^i} \Big[\sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{z^{i\prime}}} \boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, s')\\
&\qquad \big[\mathcal{R}^i(s, a^i, \boldsymbol{a^{-i}}) - \rho^i_{\theta^i} + v^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}})\big] \Big] \qquad (12)\\
&= -\nabla_{\theta^i} \rho^i_{\theta^i} + \sum_{s'} \mathcal{T}(s'|s, a^i, \boldsymbol{a^{-i}}) \sum_{\boldsymbol{\theta^{-i\prime}}} \boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}}, s, a^i, \boldsymbol{a^{-i}}, \boldsymbol{r^{-i}}, s') \times\\
&\quad \nabla_{\theta^i} v^i_{\theta^i}(s', \boldsymbol{\theta^{-i\prime}}).
\end{aligned}
$$

Then, we summarize Equation (11) and Equation (12) together and re-arrange terms to obtain:

$$\begin{aligned}\nabla_{\theta^i}\rho^i_{\theta^i} = \Big[&\sum_{a^i}\nabla_{\theta^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})q^i_{\theta^i}(s,\boldsymbol{\theta^{-i}},a^i,\boldsymbol{a^{-i}})+\\ &\sum_{a^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})\sum_{s'}\mathcal{T}(s'|s,a^i,\boldsymbol{a^{-i}})\times\\ &\sum_{\boldsymbol{\theta^{-i\prime}}}\boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}},s,a^i,\boldsymbol{a^{-i}},\boldsymbol{r^{-i}},s')\nabla_{\theta^i}v^i_{\theta^i}(s',\boldsymbol{\theta^{-i\prime}})\Big]-\nabla_{\theta^i}v^i_{\theta^i}(s,\boldsymbol{\theta^{-i}}),\end{aligned} \tag{13}$$

where $\nabla_{\theta^i}\rho^i_{\theta^i}=\nabla_{\theta^i}J^i_\pi(\theta^i)$. We define agent i's steady distribution $\boldsymbol{\mu^i_{\theta^i}}$ under θ^i with respect to s and $\boldsymbol{\theta^{-i}}$ as the special distribution that satisfies:

$$\begin{aligned}\boldsymbol{\mu^i_{\theta^i}}(s',\boldsymbol{\theta^{-i\prime}}) = &\sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\sum_{a^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})\sum_{s'}\mathcal{T}(s'|s,a^i,\boldsymbol{a^{-i}})\times\\ &\sum_{\boldsymbol{\theta^{-i\prime}}}\boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}},s,a^i,\boldsymbol{a^{-i}},\boldsymbol{r^{-i}},s'),\end{aligned} \tag{14}$$

which exists under the communicating and unichain assumptions on $\mathcal{T}$ and $\boldsymbol{\mathcal{U}}$, respectively (Puterman, 1994). We now apply the steady distribution to Equation (13) and derive the final expression for policy gradient:

$$\begin{aligned}\nabla_{\theta^i}J^i_\pi(\theta^i) &= \sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\Big(\Big[\sum_{a^i}\nabla_{\theta^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})q^i_{\theta^i}(s,\boldsymbol{\theta^{-i}},a^i,\boldsymbol{a^{-i}})+\\ &\quad\sum_{a^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})\sum_{s'}\mathcal{T}(s'|s,a^i,\boldsymbol{a^{-i}})\times\\ &\quad\sum_{\boldsymbol{\theta^{-i\prime}}}\boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}},s,a^i,\boldsymbol{a^{-i}},\boldsymbol{r^{-i}},s')\nabla_{\theta^i}v^i_{\theta^i}(s',\boldsymbol{\theta^{-i\prime}})\Big]-\nabla_{\theta^i}v^i_{\theta^i}(s,\boldsymbol{\theta^{-i}})\Big)\\ &= \sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\sum_{a^i}\nabla_{\theta^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})q^i_{\theta^i}(s,\boldsymbol{\theta^{-i}},a^i,\boldsymbol{a^{-i}})+\\ &\quad\sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\sum_{a^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})\sum_{s'}\mathcal{T}(s'|s,a^i,\boldsymbol{a^{-i}})\times\\ &\quad\sum_{\boldsymbol{\theta^{-i\prime}}}\boldsymbol{\mathcal{U}^{-i}}(\boldsymbol{\theta^{-i\prime}}|\boldsymbol{\theta^{-i}},s,a^i,\boldsymbol{a^{-i}},\boldsymbol{r^{-i}},s')\nabla_{\theta^i}v^i_{\theta^i}(s',\boldsymbol{\theta^{-i\prime}})-\\ &\quad\sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\nabla_{\theta^i}v^i_{\theta^i}(s,\boldsymbol{\theta^{-i}})\\ &= \sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\sum_{a^i}\nabla_{\theta^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})q^i_{\theta^i}(s,\boldsymbol{\theta^{-i}},a^i,\boldsymbol{a^{-i}})+\\ &\quad\sum_{s',\boldsymbol{\theta^{-i\prime}}}\boldsymbol{\mu^i_{\theta^i}}(s',\boldsymbol{\theta^{-i\prime}})\nabla_{\theta^i}v^i_{\theta^i}(s',\boldsymbol{\theta^{-i\prime}})-\sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\nabla_{\theta^i}v^i_{\theta^i}(s,\boldsymbol{\theta^{-i}})\\ &= \sum_{s,\boldsymbol{\theta^{-i}}}\boldsymbol{\mu^i_{\theta^i}}(s,\boldsymbol{\theta^{-i}})\sum_{a^i}\nabla_{\theta^i}\pi(\cdot|s,\boldsymbol{\theta^{-i}};\theta^i)\sum_{\boldsymbol{a^{-i}}}\pi(\cdot|s;\boldsymbol{\theta^{-i}})q^i_{\theta^i}(s,\boldsymbol{\theta^{-i}},a^i,\boldsymbol{a^{-i}})\end{aligned} \tag{15}$$

□

D Additional Implementation Details

Our neural networks for the policy, q-functions consist of 3 fully-connected (FC) layers. Regarding the inference module, the encoder consists of a FC input layer followed by a single-layer LSTM and a FC output layer. The encoder outputs the mean and standard deviation for the Gaussian distribution of $p(\hat{\boldsymbol{z}}^{-i}_t|\tau^i_{0:t-1};\phi^i_{\text{enc}})$, in which we sample $\hat{\boldsymbol{z}}^{-i}_t$. Because it is impractical to input the entire interactions from the beginning of the game to the encoder, we limit $\tau^i_{0:t-1}$ to be recent 100 interactions. The decoder consists of 3 FC layers and outputs a probability for the categorical distribution of another agent.

Pseudocode. Algorithm for FURTHER is provided below:

Algorithm 1 FURTHER Pseudocode

Require: Learning rates $\alpha_q, \alpha_\pi, \alpha_\phi$, Soft q-target update rate τ_q

# Agent initialization
for each agent i **do**
  Initialize RL module $\theta^i, \psi_1^i, \psi_2^i, \bar{\psi}_1^i, \bar{\psi}_2^i, \rho_{\theta^i}^i, \mathcal{D}^i$
  Initialize inference module $\phi_{\text{enc}}^i, \phi_{\text{dec}}^i$
  Initialize other agents' latent strategies $\hat{\boldsymbol{z}}_{\mathbf{0}}^{-\boldsymbol{i}}$
end for
for each timestep t **do**
  # Decentralized execution
  For each agent i, select $a_t^i \sim \pi(\cdot|s_t, \hat{\boldsymbol{z}}_{\boldsymbol{t}}^{-\boldsymbol{i}}; \theta^i)$
  Execute joint action $\boldsymbol{a_t}$ and receive next state s_{t+1} and joint rewards $\boldsymbol{r_t}$
  For each agent i, infer next updated policies of other agents $\hat{\boldsymbol{z}}_{\boldsymbol{t+1}}^{-\boldsymbol{i}} \sim p(\cdot|\tau_{0:t+1}^i; \phi_{\text{enc}}^i)$
  For each agent i, add a transition to its replay memory $\mathcal{D}^i \leftarrow \mathcal{D}^i \cup \{s_t, \hat{\boldsymbol{z}}_{\boldsymbol{t}}^{-\boldsymbol{i}}, a_t^i, \boldsymbol{a}_{\boldsymbol{t}}^{-\boldsymbol{i}}, r_t^i, s_{t+1}, \hat{\boldsymbol{z}}_{\boldsymbol{t+1}}^{-\boldsymbol{i}}\}$
  # Decentralized training
  for each agent i **do**
    $\{\psi_\beta^i, \rho_{\theta^i}^i\} \leftarrow \{\psi_\beta^i, \rho_{\theta^i}^i\} - \alpha_q J_q^i(\psi_\beta^i, \rho_{\theta^i}^i)$ for $\beta = 1, 2$
    $\theta^i \leftarrow \theta^i + \alpha_\pi J_\pi^i(\theta^i)$
    $\{\phi_{\text{enc}}^i, \phi_{\text{dec}}^i\} \leftarrow \{\phi_{\text{enc}}^i, \phi_{\text{dec}}^i\} - \alpha_\phi J_{\text{elbo}}^i(\phi_{\text{enc}}^i, \phi_{\text{dec}}^i)$
    $\bar{\psi}_\beta^i \leftarrow \tau_q \psi_\beta^i + (1 - \tau_q)\bar{\psi}_\beta^i$ for $\beta = 1, 2$
  end for
end for

E ELBO DERIVATION

We derive our ELBO optimization in Equation (6) for the inference module. In particular, we follow the ELBO derivation in Zintgraf et al. (2020) (Appendix A) and modify it for our multiagent setting:

$$
\begin{aligned}
\mathbb{E}_{p(\tau_{0:t}^i)}\Big[\log p(\tau_{0:t}^i; \phi_{\text{dec}}^i)\Big] &= \mathbb{E}_{p(\tau_{0:t}^i)}\Big[\log \int p(\tau_{0:t}^i, \hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i) d\hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}\Big] \\
&= \mathbb{E}_{p(\tau_{0:t}^i)}\Big[\log \int p(\tau_{0:t}^i, \hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i) \frac{p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}{p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)} d\hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}\Big] \\
&= \mathbb{E}_{p(\tau_{0:t}^i)}\Big[\log \mathbb{E}_{p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\big[\frac{p(\tau_{0:t}^i, \hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i)}{p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\big]\Big] \\
&\geq \mathbb{E}_{p(\tau_{0:t}^i), p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\Big[\log \frac{p(\tau_{0:t}^i, \hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i)}{p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\Big] \\
&= \mathbb{E}_{p(\tau_{0:t}^i), p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\Big[\log p(\tau_{0:t}^i, \hat{\boldsymbol{z}}_{\boldsymbol{0:t-1}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i) - \log p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)\Big] \\
&= \mathbb{E}_{p(\tau_{0:t}^i), p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\Big[\sum_{k=0}^{t-1} \log p(\boldsymbol{a}_{\boldsymbol{k}}^{-\boldsymbol{i}}|s_k, \hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i) + \sum_{k=0}^{t-1} \log p(\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}}) - \\
&\qquad \sum_{k=1}^{t} \log p(\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}}|\tau_{0:k-1}^i; \phi_{\text{enc}}^i)\Big]. \qquad (16)
\end{aligned}
$$

Finally, we summarize terms to obtain Equation (6):

$$
\mathbb{E}_{p(\tau_{0:t}^i), p(\hat{\boldsymbol{z}}_{\boldsymbol{0:t}}^{-\boldsymbol{i}}|\tau_{0:t}^i; \phi_{\text{enc}}^i)}\Big[\sum_{k=0}^{t-1} \underbrace{\log p(\boldsymbol{a}_{\boldsymbol{k}}^{-\boldsymbol{i}}|s_k, \hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}}; \phi_{\text{dec}}^i)}_{\text{Reconstruction loss}} - \underbrace{D_{\text{KL}}\big(p(\hat{\boldsymbol{z}}_{\boldsymbol{k+1}}^{-\boldsymbol{i}}|\tau_{0:k}^i; \phi_{\text{enc}}^i)||p(\hat{\boldsymbol{z}}_{\boldsymbol{k}}^{-\boldsymbol{i}})\big)}_{\text{KL divergence}}\Big].
$$

F HYPERPARAMETER DETAILS

Hyperparameter	Value
Critic learning rate α_q	2e-3
Actor learning rate α_π	7.5e-4
Inference learning rate α_ϕ	2e-3
Entropy weight α	0.1
Dimension of latent space $\lvert \boldsymbol{z}^{-i} \rvert$	3
Discount factor γ	0.99
Number of hidden units in FC	16
Number of hidden units in LSTM	16

Table 5: IBS Experiment

Hyperparameter	Value
Critic learning rate α_q	1e-4
Actor learning rate α_π	1.5e-4
Inference learning rate α_ϕ	1e-4
Entropy weight α	0.1
Dimension of latent space $\lvert \boldsymbol{z}^{-i} \rvert$	10
Discount factor γ	0.99
Number of hidden units in FC	16
Number of hidden units in LSTM	16

Table 6: IC Experiment

Hyperparameter	Value
Critic learning rate α_q	1.5e-4
Actor learning rate α_π	1e-4
Inference learning rate α_ϕ	1.5e-4
Entropy weight α	0.5
Dimension of latent space $\lvert \boldsymbol{z}^{-i} \rvert$	10
Discount factor γ	0.99
Number of hidden units in FC	16
Number of hidden units in LSTM	16

Table 7: IMP Experiment