Cooperative-Competitive Reinforcement Learning with History-Dependent Rewards

Keyang He
THINC Lab, Computer Science
University of Georgia
keyang@uga.edu

Bikramjit Banerjee
Computing Sciences and Engineering
University of Southern Mississippi
Bikramjit.Banerjee@usm.edu

Prashant Doshi
THINC Lab, Computer Science
University of Georgia
pdoshi@uga.edu

ABSTRACT

Consider a typical organization whose worker agents seek to collectively cooperate for its general betterment. However, each individual agent simultaneously seeks to act to secure a larger chunk than its co-workers of the annual increment in compensation, which usually comes from a fixed pot. As such, the agents in an organization must cooperate and compete. Another feature of many organizations is that a worker receives a bonus, which is often a fraction of previous year's total profit. As such, the agent derives a reward that is also partly dependent on historical performance. How should the individual agent decide to act in this context? Few methods for the mixed cooperative-competitive setting have been presented in recent years, but these are challenged by problem domains whose reward functions additionally depend on historical information. Recent deep multi-agent reinforcement learning (MARL) methods using long short-term memory (LSTM) may be used, but these adopt a joint perspective to the interaction or require explicit exchange of information among the agents to promote cooperation, which may not be possible under competition. In this paper, we first show that the agent's decision-making problem can be modeled as an interactive partially observable Markov decision process (I-POMDP) that captures the dynamic of a history-dependent reward. We present an *interactive* advantage actor-critic method (IA2C$^+$), which combines the independent advantage actor-critic network with a belief filter that maintains a belief distribution over other agents' models. Empirical results show that IA2C$^+$ learns the optimal policy faster and more robustly than several baselines.

KEYWORDS

Actor-critic; Mixed setting; Organization; Reinforcement learning

ACM Reference Format:
Keyang He, Bikramjit Banerjee, and Prashant Doshi. 2021. Cooperative-Competitive Reinforcement Learning with History-Dependent Rewards. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

1 INTRODUCTION

Real-world multi-agent domains involving interactions among agents are often not purely cooperative or competitive. For example, consider a typical organization whose worker agents seek to collectively cooperate for its general betterment. However, each individual agent simultaneously also seeks to act to secure a larger chunk than its co-workers of the annual increment in compensation, which usually comes from a *fixed* pot. Another feature of many organizations is that a worker receives a bonus, which is often a fraction of the previous year's total profit. Thus, the agent derives a reward that partly depends on historical performance. The agent in an organization then faces a context where it must cooperate and compete while collecting history-dependent rewards.

A potential approach to solving the worker agent's decision-making problem is multi-agent reinforcement learning (MARL). While traditional MARL has made significant strides in fields such as game playing (e.g., AlphaStar [14]) and robotics, it struggles in problems involving interactions where the individual and group interests may conflict with each other. In response to this challenge, MARL suited to both cooperative and competitive settings has received attention recently [3, 4, 9]. However, most of these recent methods take a joint perspective to the interaction and require explicit exchange of information among the learning agents, which may not be possible under competition.

In this paper, we first introduce and formalize the Organization problem as a quintessential domain involving mixed cooperation-competition, while exhibiting both partial observability and history-dependent rewards. Next, we show that the individual agent's decision making can be modeled using an interactive partially observable Markov decision process (I-POMDP) [5] despite the presence of history-dependent rewards in the problem, and introduce an approach to MARL for this type of problems. In particular, we introduce an interactive advantage actor-critic algorithm (labeled IA2C$^+$), which combines an independent advantage actor-critic [10] with a belief filter that maintains a belief distribution over the other agents' models, and updates the belief using private observations. We show that even when the set of models attributed to the other agents may not contain a grain of truth, agents using IA2C$^+$ still converge to the optimal policy significantly faster than several relevant baselines, and remain consistent for greater levels of noise in their observations, compared to those baselines.

2 BACKGROUND

In this section, we briefly review the two main components on which this work is built: a well-known model of decision making in a multi-agent environment and actor-critic RL [8].

2.1 Overview of Interactive POMDPs

Interactive partially observable Markov decision processes are a generalization of POMDPs [6] to sequential decision-making in multi-agent environments [2, 5]. Formally, an I-POMDP for agent i in an environment with one other agent j is defined as,

Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.

$$\text{I-POMDP}_i = \langle IS_i, A, T_i, O_i, Z_i, R_i, OC_i \rangle$$

• IS_i denotes the interactive state space. This includes the physical state S as well as models of the other agent M_j, which may be intentional (ascribing beliefs, capabilities and preferences) or subintentional [1]. Examples of the latter are probability distributions and finite state machines. In this paper, we ascribe subintentional models to the other agent.
• $A = A_i \times A_j$ is the set of joint actions of both agents.
• T_i represents the transition function, T_i: $S \times A \times S \longrightarrow [0, 1]$. The transition function is defined over the physical states and excludes the other agent's models. This is a consequence of the model non-manipulability assumption, which states that an agent's actions do not directly influence the other agent's models.
• O_i is the set of agent i's observations.
• Z_i is the observation function, Z_i: $A \times S \times O \longrightarrow [0, 1]$. The observation function is defined over the physical state space only as a consequence of the model non-observability assumption, which states that other's model parameters may not be observed directly.
• R_i defines the reward function for agent i, R_i: $S \times A \longrightarrow \mathbb{R}$. The reward function for I-POMDPs usually assigns preferences over the physical states and actions only.
• OC_i is the subject agent's optimality criterion, which may be a finite horizon H or a discounted infinite horizon where the discount factor $\gamma \in (0, 1)$.

Without loss of generality, let the subintentional model ascribed to j take the form $m_j = (h_j, \pi_j, Z_j)$ where h_j is agent j's action-observation history, π_j is a candidate policy, and Z_j is its observation function. Given agent i's belief over interactive states b_i, on action a_i and receiving observation o_i, the belief is updated as:

$$\begin{aligned} b'_i(is') \propto & \sum_{is} b_i(is) \sum_{a_j} Pr(a_j|m_j) Z_i(a, s', o'_i) T_i(s, a, s') \\ & \times \sum_{o'_j} \delta_K(APPEND(h_j, o'_j), h'_j) Z_j(a, s', o'_j) \end{aligned} \tag{1}$$

where α is a normalizing constant, h_j and h'_j are part of m_j and m'_j, respectively. δ_K is the Kronecker-delta function, and APPEND returns a string with the second argument appended to the first.

Each belief state b_i is associated with a value given by:

$$\begin{aligned} V(b_i) = \max_{a_i \in A_i} \Big\{ & \sum_{is} \sum_{a_j} R_i(s, a_i, a_j) Pr(a_j|m_j) b_i(is) \\ & + \gamma \sum_{o_i \in O_i} Pr(o_i|a_i, b_i) V(b'_i(is')) \Big\} \end{aligned} \tag{2}$$

where $b'_i(is')$ is obtained as shown in Equation 1.

2.2 Actor-Critic RL

RL problems are typically modeled using *Markov decision processes* or MDPs [12], which is defined by the tuple $\langle S, A, T, R \rangle$. The parameters in the tuple have their standard semantics for single-agent contexts. The agent's goal is to learn a policy $\pi : S \mapsto A$ that maximizes the sum of current and future rewards from any state s, given by,

$$V^\pi(s) = \mathbb{E}_T[R(s, \pi(s)) + \gamma R(s', \pi(s')) + \gamma^2 \ldots]$$

where $s, s', \ldots$ are successive samplings from the distribution T following the Markov chain with policy π, and $\gamma \in (0, 1)$ is a discount factor. This is sometimes facilitated by learning an action value function, $Q(s, a)$ given by

$$Q(s, a) = R(s, a) + \max_\pi \gamma \sum_{s'} T(s, a, s') V^\pi(s'). \tag{3}$$

There are two main categories of reinforcement learning: value-based and policy-based RL. Value-based RL learns an optimal value function (e.g., V given above) that maps each state (or state-action pair) to a value. It is more sample efficient and stable compared to policy-based RL, but it usually requires that the action space be finite. Policy-based RL learns the optimal policy directly, sometimes without using a value function. It is useful when the action space is continuous, and it has a faster convergence due to directly searching the policy space.

Actor-critic methods take advantage of both value-based and policy-based RL while eliminating some drawbacks. It splits the model into two components, an actor and a critic, where the actor controls how the agent acts by learning the optimal policy, and the critic evaluates the actor's actions by computing the action value (Q value in Equation 3) function, or directly the value function (V). The actor and the critic are optimized separately during the training. The interaction and complement allow the architecture to be more robust than if the two models were used individually.

Value-based RL methods typically display a high variance due to the uncertainty embedded in the agent's experience. To mitigate this, instead of using the Q value for the critic, advantage actor-critic (A2C) uses *advantage* values, given for a policy π by

$$\begin{aligned} A(s, a) &= Q(s, a) - V^\pi(s) \\ &= R(s, a) + \gamma \sum_{s'} T(s, a, s') V^\pi(s') - V^\pi(s). \end{aligned}$$

Advantage represents how much better a particular action is at a state compared to the value of the state. While the critic can be optimized by reducing the mean square of the advantages estimated from samples, the actor, $\pi_{\boldsymbol{\theta}}$ (parametrized by $\boldsymbol{\theta}$) can be optimized by gradient descent using gradients:

$$\mathbb{E}_{s \sim d^{\pi_\theta}, a \sim \pi_\theta} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|s) A(s, a),$$

where $d^{\pi_\theta}(s) = \sum_{t=0}^{\infty} \gamma^t Pr(s_t = s|s_0, \pi_{\boldsymbol{\theta}})$ is the discounted state distribution that results from following policy $\pi_{\boldsymbol{\theta}}$.

In an I-POMDP setting, the state is not directly observable; instead the learner receives an observation that is usually (noisily) correlated with the (hidden) state and other agents' actions. The advantage function can be reformulated in terms of belief as

$$\begin{aligned} A(b_i, a_i) = \sum_{is} \Big\{ & \sum_{a_j} R_i(s, a_i, a_j) Pr(a_j|m_j) b_i(is) \\ & + \gamma \sum_{o_i} Pr(o_i|a_i, b_i) V^\pi(b'_i(is')) - V^\pi(b_i(is)) \Big\} \end{aligned} \tag{4}$$

Assuming $\pi_{\boldsymbol{\theta}}$ maps beliefs to actions, its gradients are

$$\mathbb{E}_{b \sim b^{\pi_\theta}, a \sim \pi_\theta} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a|b) A(b, a). \tag{5}$$

where $b^{\pi_\theta}(is) = \sum_{t=0}^{\infty} \gamma^t Pr(is_t = is|is_0, \pi_{\boldsymbol{\theta}})$ is the discounted belief distribution that results from following policy $\pi_{\boldsymbol{\theta}}$.

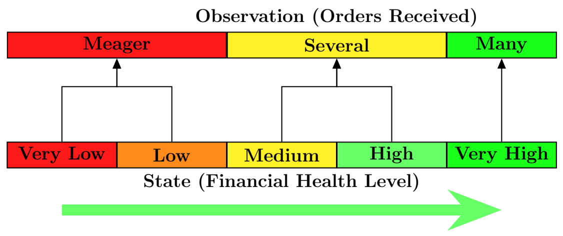


Figure 1: States of the organization domain and the non-deterministic public observations.

3 THE ORGANIZATION DOMAIN

We introduce the new Organization domain, which models a typical business organization featuring a mix of cooperation toward the overall improvement of the organization and individual competition. Notably, the reward function may not have the Markovian property. For example, a proportion of the rewards from the past is added to the current reward as a bonus to the worker if the organization operated well in the past year. We call this a *history-dependent* reward. The goal of each agent is to maximize the sum of all reward signals that it receives, including the history-dependent reward.

3.1 Specification

3.1.1 State and Observation Sets. The state space of the problem represents the organization's financial health level, which is discretized into five states: very low (denoted as s_{vl}), low (s_l), medium (s_m), high (s_h), and very high (s_{vh}). Agents cannot directly observe the financial health level of the organization; instead, they only receive observations of the number of orders received by the organization as well as other agents' actions. We assume that the observation is decomposed into *public* and *private* observations. While public observation is common to all agents and depends on the underlying state only, the private observation informs about other agents' actions and is perceived by the corresponding agent only. There are three possible public observations pertaining to the number of orders: *meager* (o_e), *several* (o_s), and *many* (o_m). o_e indicates the organization is in either s_{vl} or s_l, o_s indicates the organization is in either s_m or s_h, and o_m indicates the organization is in s_{vh}. Three private observations represent the agents' three possible actions, respectively, albeit noisily. Agents have 0.8 probability of perceiving the actual action of the other agent, and 0.2 probability of receiving either private observation corresponding to actions which the other agent did not take. Figure 1 illustrates the relationship between states and public observations.

3.1.2 Action Space and Transition Function. Each agent has three possible actions: *self*, *balance*, and *group*. The *self* action solely benefits the agent while the *group* action benefits the organization at the expense of self. The *balance* action benefits both self and the organization. A joint action is determined by the distribution of the individual agent's actions. If the number of agents who picked *self* equals the number of agents who picked *group*, then joint action is a *balance* action. *Balance* joint action does not change the underlying state. If the number of agents who pick *self* is greater than those who pick *group*, the joint action will be a *self* action. *Self* joint action brings down the health level (the current state) by one when the current state is higher than s_{vl}; otherwise, the state remains unchanged. If the number of agents who pick *self* is smaller than the number who pick *group*, the joint action will be a *group* action. *Group* joint action increases the current health level by one when the current state is lower than s_{vh}, otherwise, the state remains unchanged. Besides, if all agents performed *group* individual action, the state increases by two. Figure 2 demonstrates state transitions of the organization domain.

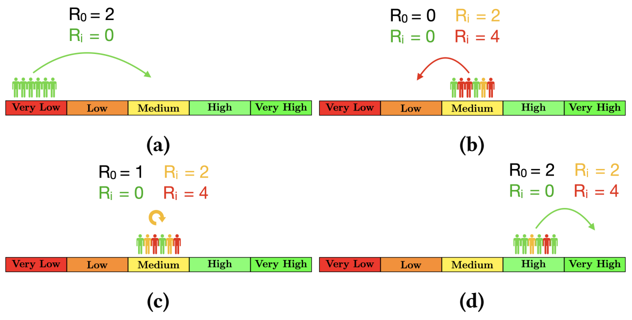


Figure 2: Agents picking individual, balance, and group actions are colored in red, yellow, and green, respectively. (a) If all agents pick group action, the state increases by two. (b) The majority picks individual action, state decreases by one. (c) There is no majority action, state remains unchanged. (d) Unanimously picking group action can only increase the state by one when the current state is s_h; therefore, minority agents may pick other actions to receive higher individual rewards.

3.1.3 Reward Function. At time step t, each agent receives a sum of individual and group rewards, and the bonus. The agent i's individual reward, R_i, depends on the state and its own action:

$$R_i^t \leftarrow R_i(s^t, a_i^t).$$

The group reward, common to all agents, depends on the current state and joint action:

$$R_0^t \leftarrow R(s^t, a^t).$$

The bonus or history-dependent reward is a proportion of the total reward from the previous time step. For $\phi \in (0, 1)$,

$$R_{-1}^t = \phi(\sum\nolimits_i R_i^{t-1} + R_0^{t-1}).$$

The goal of agent i is to optimize the expected sum of individual, group, and history rewards, $\mathbb{E}_{trajectories}\left[\sum_t \gamma^t (R_0^t + R_i^t + R_{-1}^t)\right]$.

The *Group* action is cooperative giving $R_i = 0$ and $R_0 = r$, where $r \in \mathbb{R}$. The *Self* action benefits an agent giving reward $R_i = \beta r$ to the agent, where $\beta > 1$, and $R_0 = 0$. *Balance* action gives $R_i = c\frac{1+\beta}{\alpha}r$, where $0 < c < 1$ and $r < \frac{1+\beta}{\alpha} < \beta$ and $R_0 = (1-c)\frac{1+\beta}{\alpha}r$, thereby benefiting both the agent and the business. If the organization reaches state s_{vl}, each agent receives a penalty of p no matter which individual action was picked.

3.2 Importance of History-Dependent Reward

In this subsection, we demonstrate the importance of history-dependent reward in the Organization domain. Suppose that policy π_0 leads to the reward sequence $\{\beta r, \beta r, r, \beta r, \ldots\}$ for an agent, and policy π_1 leads to the reward sequence $\{\beta r, \beta r, \frac{1+\beta}{\alpha} r, \frac{1+\beta}{\alpha} r, \ldots\}$. For convenience, we set $d = \frac{1+\beta}{\alpha}$. For horizon $H = 4$, the total reward from performing policy π_0 is:

$$\begin{aligned}&\beta r + (\phi\beta r + \beta r) + (\phi^2\beta r + \phi\beta r + r) + (\phi^3\beta r + \phi^2\beta r + \phi r + \beta r)\\&= \phi^3\beta r + 2\phi^2\beta r + 2\phi\beta r + \phi r + 3\beta r + r\end{aligned}$$

The total reward from performing policy π_1 is:

$$\begin{aligned}&\beta r + (\phi\beta r + \beta r) + (\phi^2\beta r + \phi\beta r + dr) + (\phi^3\beta r + \phi^2\beta r + \phi dr + dr)\\&= \phi^3\beta r + 2\phi^2\beta r + 2\phi\beta r + \phi dr + 2\beta r + 2dr\end{aligned}$$

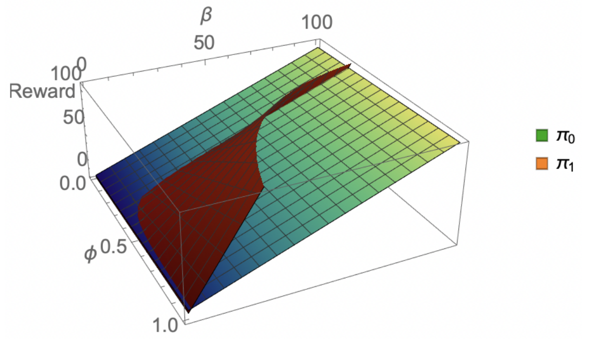


Figure 3: The total reward from policy π_0 and π_1 with varying β and ϕ for horizon of 4.

Then we can compare the total rewards from these two policies with varying choices of β and ϕ. We use a simple program to check if ϕ can affect which of the two policies is optimal for horizon $H \geq 4$ when d is set to $\frac{9}{4}$. The result shows that for every horizon from 4 to 100, the history-dependent parameter ϕ is always a deciding factor in finding the optimal policy. Figure 3 shows the total reward from policy π_0 and π_1 with varying β and ϕ for horizon 4. Notice that ϕ influences when each surface has the higher total reward.

3.3 Modeling the Org Domain as an I-POMDP

The I-POMDP framework is suited to modeling Organization from an individual agent i's perspective. We formulate such an I-POMDP in this section. Suppose that if we remove the history-dependent rewards, then the residual environment has perfectly Markovian dynamics, with functions $T(s_f, a_i, a_j, s'_f)$, $Z(a_i, a_j, s_f, s'_f, o'_f)$, and $R(s_f, a_i, a_j)$, where s_f is an ordinary physical state and o_f is an observation related to this state. (Note that we assume Z depends on both s_f and s'_f, unlike in Section 2.1, as this is needed for our formulation.) To preserve the Markov property even when R_{-1} is introduced, we add a continuous-valued feature, $s_r \in \mathbb{R}$, to the state space that memorizes the reward from the last step. The new I-POMDP for agent i in the Organization domain with one other agent j has an expanded definition:

$$\text{I-POMDP}_i = \langle IS_i, A, T_i, \Omega_i, W_i, Z_i, O_i, R_i, OC_i \rangle$$

• The interactive state space IS_i now includes the physical state S_f, the history-reward state S_r, as well as models of the other agent M_j. We let the latter be subintentional in this domain.
• $A = A_i \times A_j$ is the set of joint actions of both agents.
• T_i represents the transition function, now defined as:

$$\begin{aligned}&T_i(\langle s_f, s_r\rangle, a_i, a_j, \langle s'_f, s'_r\rangle)\\&= \begin{cases} T(s_f, a_i, a_j, s'_f), & \text{if } s'_r = R(s_f, a_i, a_j) + \phi \cdot s_r \\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{6}$$

• Ω_i is the set of agent i's *private* observations.
• $W_i : A \times \Omega_i \rightarrow [0, 1]$ is the private observation function.
• $O_i = O_f \times O_r$ is the set of agent i's public observations, where O_f informs about the state and $O_r = S_r$, allowing the agent to observe the past reward.
• Z_i is the observation function, defined as:

$$\begin{aligned}&Z_i(a_i, a_j, \langle s_f, s_r\rangle, \langle s'_f, s'_r\rangle, \langle o'_f, o'_r\rangle)\\&= \begin{cases} Z(a_i, a_j, s_f, s'_f, o'_f), & \text{if } (s'_r = R(s_f, a_i, a_j) + \phi \cdot s_r) \wedge (o'_r = s'_r) \\ 0 & \text{otherwise}\end{cases}\end{aligned} \tag{7}$$

• R_i defines the reward function for agent i:

$$R_i(\langle s_f, s_r\rangle, a_i, a_j) = R(s_f, a_i, a_j) + \phi \cdot s_r \tag{8}$$

•OC_i is as defined previously in Section 2. In our experiments, we use the discounted infinite horizon with γ is set to 0.9.

The belief update equation for the new I-POMDP formulation is:

$$\begin{aligned}&b'_i(is'|b_i, a_i, o'_i, \omega'_i) = b'_i(\langle s'_f, s'_r\rangle|b_i, a_i, o'_i, \omega'_i) \times\\&b'_i(m'_j|\langle s'_f, s'_r\rangle, b_i, a_i, o'_i, \omega'_i)\end{aligned} \tag{9}$$

where the first term can be derived as:

$$\begin{aligned}&b'_i(\langle s'_f, s'_r\rangle|b_i, a_i, o'_i, \omega'_i) \propto \sum_{a_j} \sum_{s_f, s_r = \frac{o'_r - R_i(s_f, a_i, a_j)}{\phi}} Pr(a_j|m_j)\\&\times T(s_f, a_i, a_j, s'_f)\ b_i(\langle s_f, s_r\rangle)\ Z; (a_i, a_j, s_f, s'_f, o'_f)\\&\times W_i(a_i, a_j, \omega'_i)\ \delta_k(APPEND(h_j, a_j, o'_f), h'_j).\end{aligned}$$

Recall that o'_f is a public observation received by both agents. We derive the second term of the belief update decomposition in the next subsection. For convenience, we summarize the full update as $\tau(b_i, a_i, o'_i, \omega'_i, b'_i)$.

The Bellman's equation for the new I-POMDP is:

$$\begin{aligned}V(b_i) = \max_{a_i} \Bigg[&\sum_{s_f, s_r} \sum_{a_j} R_i(\langle s_f, s_r\rangle, a_i, a_j) Pr(a_j|m_j) b_i(\langle s_f, s_r\rangle)\\&+ \gamma \sum_{a_j} \sum_{s'_f, s'_r = R(s_f, a_i, a_j) + \phi \cdot s_r} Pr(a_j|m_j) \sum_{o'_i, \omega'_i} T(s_f, a_i, a_j, s'_f)\\&\times b_i(\langle s_f, s_r\rangle) Z(a_i, a_j, s_f, s'_f, o'_f) W_i(a_i, a_j, \omega'_i)\ V(\tau(b_i, a_i, o'_i, \omega'_i, b'_i))\Bigg]\end{aligned} \tag{10}$$

3.4 Model Belief Update

The belief update in Equation 9 updates the belief over states and other agent's model simultaneously. The state belief update and model belief update can be separated in case when the two parts are not handled by a single network. In a two-agent setting, agent j's model set at time step t is denoted as M_j, where M_j contains a finite number of pre-defined models m_j. Given agent i's action a_i, public observation $\langle o'_f, o'_r \rangle$, private observation ω'_i, and previous belief b_i, the model belief update is defined as below:

$$
\begin{aligned}
& b'_i(m'_j | \langle s'_f, s'_r \rangle, b_i, a_i, o'_i, \omega'_i) = \frac{Pr(m'_j, \omega'_i | \langle s'_f, s'_r \rangle, a_i, o'_i, b_i)}{Pr(\omega'_i | \langle s'_f, s'_r \rangle, a_i, o'_i, b_i)} \\
& \propto \sum\nolimits_{m_j} b_i(m_j) Pr(m'_j, \omega'_i | \langle s'_f, s'_r \rangle, a_i, o'_i, m_j) \\
& \propto \sum\nolimits_{m_j} b_i(m_j) \sum\nolimits_{a_j} Pr(m'_j, \omega'_i | \langle s'_f, s'_r \rangle, a_i, o'_i, m_j, a_j) \\
& \times Pr(a_j | \langle s'_f, s'_r \rangle, a_i, o'_i, m_j) \\
& \propto \sum\nolimits_{m_j} b_i(m_j) \sum_{a_j} Pr(a_j | m_j) Pr(m'_j, \omega'_i | \langle s'_f, s'_r \rangle, a_i, a_j, o'_i, m_j) \\
& \propto \sum_{m_j} b_i(m_j) \sum_{a_j} Pr(a_j | m_j) \; Pr(\omega'_i | m'_j, \langle s'_f, s'_r \rangle, a_i, a_j, o'_i, m_j) \\
& \times Pr(m'_j | \langle s'_f, s'_r \rangle, a_i, a_j, o'_i, m_j) \\
& \propto \sum_{m_j} b_i(m_j) \sum_{a_j} Pr(a_j | m_j) W_i(a_i, a_j, \omega'_i) \; Pr(m'_j | a_i, a_j, o'_i, m_j).
\end{aligned}
\tag{11}
$$

The last equation follows because the private observation function does not condition the private observation on the physical state. To simplify the term $Pr(m'_j | a, o'_i, m_j)$, we substitute m'_j with its components: $m'_j = (\pi'_j, h'_j)$, where h'_j is agent j's action-observation history at next time step and π'_j is j's policy.

$$
\begin{aligned}
Pr(m'_j | a_i, a_j, o'_i, m_j) &= Pr(\pi'_j, h'_j | a_i, a_j, o'_i, \pi_j, h_j) \\
&= Pr(h'_j | \pi'_j, a_i, a_j, o'_i, \pi_j, h_j) Pr(\pi'_j | \pi_j, a_i, a_j, o'_i, h_j) \\
&= Pr(h'_j | \pi'_j, a_i, a_j, o'_i, \pi_j, h_j) Pr(\pi'_j | \pi_j, a_i, a_j, h_j) \\
&= \delta_K(APPEND(h_j, a_j, o'_i), h'_j) \delta_K(\pi_j, \pi'_j)
\end{aligned}
\tag{12}
$$

where π^t_j is j's policy contained in m^t_j (note that the action-observation history in a model expands with time steps, but the policy in the model does not change).

By substituting $Pr(m'_j | a_i, a_j, o_i, m_j)$ with Equation 12, we can rewrite Equation 11 as

$$
\begin{aligned}
& b'_i(m'_j | \langle s'_f, s'_r \rangle, b_i, a_i, o'_i, \omega'_i) \propto \sum\nolimits_{m_j} b_i(m_j) \sum\nolimits_{a_j} Pr(a_j | m_j) \\
& W_i(a_i, a_j, \omega'_i) \delta_K(\pi_j, \pi'_j) \sum\nolimits_{o'_i} \delta_K(APPEND(h_j, a_j, o'_i), h'_j).
\end{aligned}
\tag{13}
$$

The second Kronecker-delta function δ_K is 1 if the updated history matches the one in m'_j, it is 0 otherwise. When modeling more than one agent, the set M_j becomes a Cartesian product of the sets of models attributed to each agent being modeled.

4 INTERACTIVE A2C$^+$

Most current deep reinforcement learning methods require explicit exchange of information among agents. Some methods use maximum likelihood estimation (MLE) to predict other agents' actions from historical information in scenarios where agents cannot exchange information. However, the result is unsatisfactory in complex domains with large state and action spaces, as comparative evaluations have shown [9]. In this paper, we present IA2C$^+$, which extends advantage actor-critic by maintaining predictions of other agents' actions based on dynamic beliefs over models. We implement A2C with one input layer, one hidden layer with *tanh* activation, followed by one output layer. We published the code at https://github.com/khextendss/IA2C. For simplicity, we assume that both the critic and the actor networks map observations rather than beliefs, and that the critic maps observations to *joint action values*, $Q(\langle o_f, o_r \rangle, a_i, a_j)$. We estimate Equation 4 as:

$$A(\langle o_f, o_r \rangle, a_i, \hat{a}_j) = avg[r + \gamma Q(\langle o'_f, o'_r \rangle, a'_i, \hat{a}'_j) - Q(\langle o_f, o_r \rangle, a_i, \hat{a}j)]$$

while the actor's gradient (Equation 5) is estimated as:

$$avg[\nabla_\theta \log \pi_\theta(a_i | \langle o_f, o_r \rangle) A(\langle o_f, o_r \rangle, a_i, \hat{a}_j)]$$

where r, $\langle o'_f, o'_r \rangle$ and a'_i are samples, $\hat{a}_j$ and $\hat{a}'_j$ are predicted actions (see next section), and the *avg* is taken over sampled trajectories.

4.1 Belief Filter

We implement a Bayesian belief filter, integrated with the critic within the deep RL pipeline, to complete the model belief update. We further decompose the model belief update of Equation 13 into two steps. The first step is the *prediction*, which accounts for other agent's actions:

$$\hat{b}'_i(m'_j) = \sum_{m_j} b_i(m_j) \sum_{a_j} Pr(a_j | m_j) \delta_K(\pi_j, \pi'_j) \delta_K(APPEND(h_j, a_j, o'_i), h'_j).$$

Second step corrects the predictions using perceived observations:

$$b'_i(m'_j) \propto \hat{b}'_i(m'_j) \sum\nolimits_{a_j} W_i(a_i, a_j, \omega'_i).$$

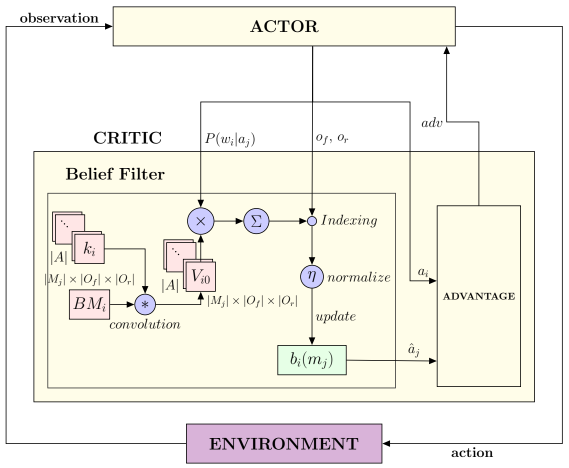


Figure 4: Belief filter samples agent j's action based on model distribution, and passes the predicted action to agent i's interactive A2C network.

Note that at each time step, the belief is represented by a $|M_j| \times |O_f| \times |O_r|$ tensor. To keep the dimension of the belief matrix finite, we round O_r to one decimal place. The prediction step is done by performing convolution on belief tensors using $|A|$-many convolution filters. After the convolution, we obtain $|A|$ belief vectors corresponding to each possible joint action. Then the correction step is executed by multiplying each belief vector with the probability of perceiving its corresponding private observation. Finally, we sum up all result vectors and index all possible models by public observations. This step represents the Kronecker-delta function.

We use model leveraging to sample a predicted action of the other agent. Figure 4 shows the network architecture used by each agent.

5 EXPERIMENTS

We evaluate the performance of the A2C method with belief filtering, labeled as IA2C$^+$, on the cooperative-competitive Organization domain using four sets of experiments. First, we show that a method that does not account for the history-dependent rewards fails to reach optimality. Second, we evaluate the need for cooperation in Organization and that agents using the IA2C$^+$ can learn to cooperate. Third, we explore the performances of other recent deep MARL methods on Organization and compare them with IA2C$^+$. Finally, we explore the impact of increasing noise levels in the observations on the convergence to the optimal policy by the various methods.

In order to model the partial observability of the problem, we extend the public observation $\langle o_f, o_r \rangle$ used in the indexing procedure to a short-term observation history that contains the current public observation as well as the previous public observation, i.e. $\{\langle o_f^{t-1}, o_r^{t-1}\rangle, \langle o_f^t, o_r^t\rangle\}$. We include 5 models of the other agent in the pre-defined model set M_j. Three models lead to solely picking *self*, *balance*, and *group* action, respectively, no matter which observation sequence is perceived. One model picks *group* action for observation sequences $\{\langle o_e^{t-1}, o_r^{t-1}\rangle, \langle o_e^t, o_r^t\rangle\}$, $\{\langle o_s^{t-1}, o_r^{t-1}\rangle, \langle o_e^t, o_r^t\rangle\}$, *balance* action for observation sequence $\{\langle o_e^{t-1}, o_r^{t-1}\rangle, \langle o_s^t, o_r^t\rangle\}$, $\{\langle o_s^{t-1}, o_r^{t-1}\rangle, \langle o_s^t, o_r^t\rangle\}$, and *self* action for observation sequences $\{\langle o_m^{t-1}, o_r^{t-1}\rangle, \langle o_m^t, o_r^t\rangle\}$, $\{\langle o_m^{t-1}, o_r^{t-1}\rangle, \langle o_s^t, o_r^t\rangle\}$, $\{\langle o_s^{t-1}, o_r^{t-1}\rangle, \langle o_m^t, o_r^t\rangle\}$. The last model picks *self* action for observation sequences $\{\langle o_e^{t-1}, o_r^{t-1}\rangle, \langle o_e^t, o_r^t\rangle\}$, $\{\langle o_s^{t-1}, o_r^{t-1}\rangle, \langle o_e^t, o_r^t\rangle\}$, *balance* action for observation sequences $\{\langle o_e^{t-1}, o_r^{t-1}\rangle, \langle o_s^t, o_r^t\rangle\}$, $\{\langle o_s^{t-1}, o_r^{t-1}\rangle, \langle o_s^t, o_r^t\rangle\}$, and *group* action for observation sequences $\{\langle o_m^{t-1}, o_r^{t-1}\rangle, \langle o_m^t, o_r^t\rangle\}$, $\{\langle o_m^{t-1}, o_r^{t-1}\rangle, \langle o_s^t, o_r^t\rangle\}$, and $\{\langle o_s^{t-1}, o_r^{t-1}\rangle, \langle o_m^t, o_r^t\rangle\}$.

5.1 History-Dependent Rewards

We introduce a baseline method, IA2C$^-$ that *does not* include the extra state and observation feature revealing the history-dependent reward. To compensate, it utilizes the LSTM for both actor and critic networks, in order to model the dependence of its immediate rewards on the history of interactions. We establish the need for utilizing recurrence in this case and thereby the need to correctly model the dependence on history, by comparing its performance with IA2C$^-$ that does not use LSTMs, using convolutional neural networks (CNN) instead in the actor and critic.

Notice from Figure 5 that IA2C$^-$ without LSTM converges, but not to the optimal policy in contrast to IA2C$^-$ with LSTM that converges to the optimal policy. From this observation, we conclude that our approach to accommodating history in Section 3.3 is *not only sufficient but also necessary*, because if an alternative I-POMDP model existed that exhibited Markovian dynamics without requiring the extra feature to enable optimal decisions, then IA2C$^-$ without LSTM would have reached the optimal policy as well. We will use IA2C$^-$ with LSTM as a baseline to compare with IA2C$^+$.

5.2 Cooperation in the Organization Domain

The optimal policy for the Organization domain involves performing the *group* joint action in response to observation history that pertain to states s_{vl}, s_l, and s_m; the *balance* joint action in response

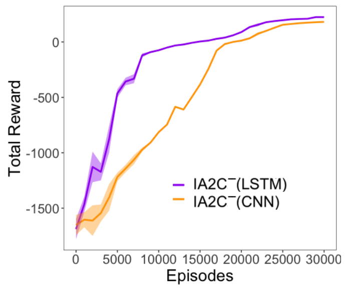


Figure 5: IA2C$^-$ without LSTM (uses a CNN instead) exhibits poor rewards for low numbers of episodes as we may expect, and eventually converges to a policy that is not optimal.

to observations that pertain to state s_h; and the *self* action in response to observations that pertain to state s_{vh}. Notice that this involves cooperation among the agents at various states. To quantify this, we compare the value of the optimal policy with that of performing the *self* joint action for all observations and the *balance* joint action for all observations. Table 1 shows a significant difference between the three values indicating that both cooperation as well as self-interestedness play a role in this domain.

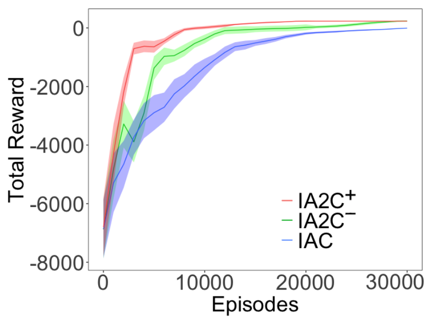
Table 1: Values of the optimal policy and other default behaviors for two agents.

Optimal	Only Group	Only Balance
226.9	132	198

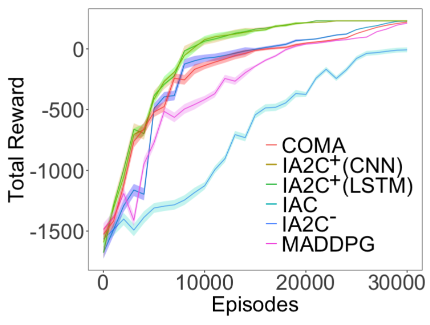
We hypothesize that independently-learning agents may not converge to the cooperation needed in this domain. Consequently, we compare the performance of agents utilizing IA2C$^+$ with CNN as the neural net architecture with those utilizing IA2C$^-$ and independent actor-critic (IAC) without belief filter on Organization with **four** learning agents. Figure 6a shows that agents learning using IA2C$^+$ and IA2C$^-$ learn the optimal policy within 30,000 episodes. We show the mean and standard deviation of 5 runs of each method in this chart. Four agents learn to unanimously pick the *group* individual action to raise the organization's financial health level. When the organization is in a better financial health level, the four agents coordinated to pick two *group* actions, one *balance* action, and one *self* action in order to maximize the total reward. However, IA2C$^+$ converges faster than IA2C$^-$ with the former requiring 20,000 episodes compared to the 30,000 episodes needed by the latter. It takes IA2C$^+$ about 3 hours to converge on a standard Linux 2.3GHz quad-core i7 processor with 8GB memory, while IA2C$^-$ takes about 10 hours to converge on the same machine. We ran IA2C$^+$ on Organization with up to 8 agents – it took 69 hours and 61,000 episodes to converge to optimal. On the other hand, plain IAC based learning failed to learn the optimal policy.

5.3 Comparison with MARL Techniques

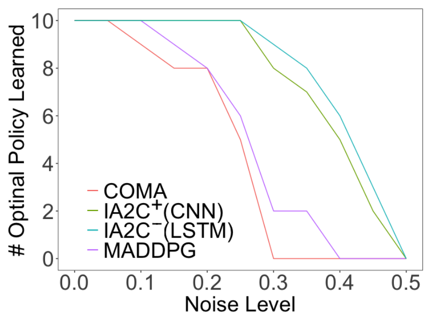
Next, we explore the performance of state-of-the-art MARL methods such as MADDPG [9] and COMA [4] on two-agent Organization. We also include the results of IA2C$^-$ mentioned in Section 5.1 and a variant of IA2C$^+$ with LSTM. We disabled the policy exchange



(a)



(b)



(c)

Figure 6: (a) Four agents, each utilizing IA2C$^+$ or IA2C$^-$, converges (value loss becomes zero) to the optimal policy whereas IAC converges but not to the optimal policy. (b) IA2C$^+$ requires fewer episodes to converge to the optimal policy compared to MADDPG, COMA, and IAC. (c) Number of runs (out of 10) in which the optimal policy was learned as a function of the noise levels in private observations. IA2C$^+$ shows the most robust performance among the MARL techniques allowing for noise up to 0.3 while still learning the optimal policy.

among agents as performed by MADDPG; instead each agent receives a noisy action whose noise probabilities are the same as those of the private observations in the Organization domain. As such, this simulates the effect of private observations, and makes it directly comparable to IA2C$^+$. As COMA employs a centralized critic, we noised the action sent by each actor to the critic to simulate the private observations.

We show the results in Figure 6b. We point out that all methods eventually converge as their respective value losses become zero. While IA2C$^+$ with CNN and IA2C$^+$ with LSTM converges to the optimal policy at 20,000 episodes, and IA2C$^-$ converges to the optimal policy at 30,000 episodes, both MADDPG and COMA do not converge within these many episodes, instead requiring almost twice as many episodes. IA2C$^+$ with CNN has similar performance with IA2C$^+$ with LSTM while only using half the time to train. Closer inspection revealed that the belief filtering often predicted the true action of the other agent with a high probability despite the noisy observations. This more accurate prediction of the other agent's actions in about 88.6% of the episodes, *despite none of the models in M_j being individually correct*, results in faster convergence to the optimal policy for each agent. IAC without belief filter shows the worst performance as we may expect, and fails to converge to the optimal policy as seen in the previous subsection.

5.4 Varying Private Observation Noise

Finally, we vary the noisiness of the private observations to test the robustness of the methods to increasing uncertainty. We gradually increase the noise level from 0 to 0.5, where noise level 0 means that the other agent's actions are perfectly observed, and noise level 0.5 means that there is only a 0.5 probability that the received observations indicate the correct actions of the other agent.

We record the number of runs out of 10 for which each method learned the optimal policy for various noise levels. Figure 6c shows the result of this experiment. IA2C$^+$ is able to consistently learn the optimal policy when the private observation noise level is increased up to 0.3. Between noise levels 0.3 and 0.5, we observe an increasing number of runs where it fails to learn the optimal policy, failing completely beyond noise level 0.5. IA2C$^+$ with LSTM has a slightly higher performance than IA2C$^+$ with CNN, however, it requires almost twice as many episodes as IA2C$^+$ with CNN to converge. In contrast, COMA and MADDPG start to fail from noise level of around 0.1, and fail completely beyond noise levels 0.35 and 0.4, respectively. Though the pre-defined models used in IA2C$^+$ belief update may have deviations from other agents' true models, beliefs over this set leads to highly accurate action predictions. We conclude that IA2C$^+$ demonstrates consistent learning and robustness to higher levels of noise compared to the baselines in the Organization domain.

6 RELATED WORK

While multi-agent RL mainly addresses purely cooperative or competitive tasks, there has been some work recently that address a mix of the two settings. We first discuss one prominent integrative work, the cooperative-competitive process (CCP), and then discuss recent work in MARL more generally, relating them to our contributions.

6.1 Organization Domain as a CCP

CCP [15] is a framework for modeling mixed cooperative-competitive sequential decision problems. It blends both by introducing a slack parameter which controls the amount of cooperation versus competition. A group-dominant CCP first maximizes the group reward, then optimizes the individual reward while allowing a deviation of up to the slack from the optimal group reward. Individual-dominant CCP follows a similar pattern but with a reversed preference. Nonlinear programming is used for solving the CCPs. Our work can be seen as a pragmatic generalization of the CCP, situated in a MARL context. Instead of manually defining a parameter to choose between cooperate or compete, Kleiman-Weiner et al. [7] present a hierarchical model that integrates low-level action plans to high-level strategy. An agent can infer other agents' high level strategy before deciding whether to select a cooperative or a competitive strategy, and thus perform corresponding low-level actions specified by selected strategies.

Because the CCP as is cannot model Organization, we generalize it to a history-dependent CCP by adding R_{-1} to the reward vector. In addition, we decompose the observation into public and private. Public observations depend on the state and the joint action, while private observations depend on other agent's individual actions. Also, the individual rewards R_i depend on agent i's individual action only, instead of the joint action in the original CCP definition. The history-dependent CCP for Organization is given below.

- $I = \{1, 2, \ldots, n\}$ is the set of n agents
- $S = S_f \times S_r$, where $S_f = \{s_{vl}, s_l, s_m, s_h, s_{vh}\}$ and S_r is the continuous state feature for memorizing the previous reward
- $\vec{A} = \{self, balance, group\}$ is the set of 3 joint actions, determined by the majority choices of the n agents, as given in Section 3.1.2
- $T : S \times \vec{A} \times S \rightarrow [0, 1]$ is the state transition function mapping state s and joint action $\vec{a}$ to successor state s', such that $T(s, \vec{a}, s') = Pr(s'|s, \vec{a})$
- $O = \{o_e, o_s, o_m\}$ is the set of public observations
- $Z : \vec{A} \times S \times O \rightarrow [0, 1]$ is the public observation function mapping joint action $\vec{a}$ and successor state s' to a public observation such that $Z(\vec{a}, s', o) = Pr(o|\vec{a}, s')$
- $\Omega = \{\omega_s, \omega_b, \omega_g\}$ is the set of 3 private observations of each agent
- $W : A \times \Omega \rightarrow [0, 1]$ is the private observation function mapping individual action a_j to a private observation ω_i such that $W(a_j, \omega_i) = Pr(\omega_i|a_j)$
- $\vec{R} = [R_{-1}, R_0, R_1, R_2, \ldots, R_n]$ is the vector of rewards; for each state s, $R_0(s, \vec{a})$ is a group reward for all agents and $R_i(s, a_i)$ is an individual reward for each agent i. R_{-1} is the discounted history-dependent component of the reward, $R_{-1}(\langle s_f, s_r \rangle) = \phi \cdot s_r$.

When $\phi = 0$ and Ω is empty, the history-dependent CCP reduces to the original CCP. While Wray et al. [15] solve the original CCP in a centralized manner for all agents, our approach is analogous to solving this variant from each individual agent's perspective.

6.2 Multi-agent RL

Multi-agent deep deterministic policy gradient (MADDPG) [9] is a MARL algorithm that can be applied to mixed cooperative-competitive settings. It extends actor-critic by exchanging policies among agents' critic, while the actor only has access to local information. After training is completed, only the actors are deployed in the environment. When direct policy exchange is not possible, each agent maintains an approximation to the true policies of the other agents. The approximation is done by maximizing the log likelihood of the other agents' actions (which the learner is able to observe). However, when the other agents' actions are not perfectly observed due to noise, MADDPG is unable to learn the optimal policy, as our experiments on Organization have shown.

Learning with opponent-learning awareness (LOLA) [3] is another actor-critic method where the agents attempt to directly influence the policy updates of other agents. Instead of learning the best response, LOLA learns to maximize the expected return after the opponent updates its policy with one naive learning step. In this way, a LOLA learner explicitly accounts for the learning of other agents in the environment, within its own learning. LOLA is suitable for both cooperative and competitive problems. Nevertheless, LOLA requires that the agents have access to each others' exact gradients. LOLA with opponent modeling removes this requirement, by estimating the other agent's gradients from the trajectories, using maximum likelihood estimation. However, it is not straightforward to accommodate either the exact gradients or the estimated ones as private observations, in a manner consistent with the private observations of IA2C$^+$, or those of our modified MADDPG and COMA for Organization, thus precluding a fair comparison with these methods in our experiments. For this reason, we have excluded LOLA from our set of baselines.

Independent deep Q-Learning (IQL) [13] extends deep Q-Learning network (DQN) [11] architecture to multi-agent settings by allowing the agents to select actions independently, and to receive separated individual rewards from the environment. IQL can learn policies ranging from fully cooperative to competitive by tuning the reward function. It demonstrates the possibility of decentralized learning in complex multi-agent environments. As we have already included IAC in our set of baselines, and Q-learning is an off-policy technique compared to the on-policy actor-critic based algorithms used in our experiments, we exclude IQL from our set of baselines.

Counterfactual multi-agent policy gradients (COMA) [4] consists of a centralized critic and multiple (decentralized) actors. COMA uses a centralized critic to compute the agent-specific advantage functions that compares the estimated return for the current joint action to a counterfactual baseline that marginalizes out one single agent's action at a time, while keeping all other agents' actions fixed. However, COMA requires access to the true state or the joint action-observation history.

7 CONCLUDING REMARKS

We introduced the Organization domain, inspired by typical real-world businesses, where agents must both cooperate and compete to attain optimal behavior. Agents in this domain not only receive noisy observations about the state and others' actions, but also obtain rewards that, in part, depend on the total reward of the previous time step, analogous to bonus pay. Subsequently, the Organization domain offers substantially more realistic challenges than previous MARL domains. The presence of history-dependent rewards challenges the applicability of traditional decision-making frameworks and the need for cooperation precludes independent learning in this domain. We presented a new method that combined decentralized actor-critic based learning with maintaining beliefs over a finite set of candidate models of the other agents. It differs from existing actor-critic methods by not requiring the exact actions performed by other agents. It also handles the non-Markovian reward function of the domain by introducing a new state feature to capture the history-dependent reward. This method is comparatively robust to noisy observations and converges significantly faster to the optimal policy in the Organization domain compared to previous state-of-the-art MARL baselines. An immediate avenue of future work is to further scale the number of agents beyond eight to better simulate real-world business organizations.

Acknowledgment. This work was supported in part by a grant from ONR #N000141310870 (to PD), from NSF #IIS-1910037 (to PD) and #IIS-1526813 (to BB).

REFERENCES

[1] Daniel Dennett. 1971. Intentional systems. *Journal of Philosophy* 68, 4 (1971), 87–106.

[2] Prashant Doshi. 2012. Decision Making in Complex Multiagent Contexts: A Tale of Two Frameworks. *AI Magazine* 33, 4 (Dec. 2012), 82.

[3] Jakob Foerster, Richard Y. Chen, Maruan Al-Shedivat, Shimon Whiteson, Pieter Abbeel, and Igor Mordatch. 2018. Learning with Opponent-Learning Awareness. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 122–130.

[4] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual Multi-Agent Policy Gradients. In *Proceedings of 32nd AAAI Conference on Artificial Intelligence 2018*. AAAI Conference on Artificial Intelligence.

[5] Piotr Gmytrasiewicz and Prashant Doshi. 2005. A Framework for Sequential Planning in Multi-Agent Settings. *Journal of Artificial Intelligence Research* 24 (2005), 49–79.

[6] Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and Acting in Partially Observable Stochastic Domains. *Artificial Intelligence* 101, 1–2 (May 1998), 99–134.

[7] Max Kleiman-Weiner, Mark Ho, Joseph Austerweil, Michael Littman, and Joshua Tenenbaum. 2016. Coordinate to cooperate or compete: Abstract goals and joint intentions in social interaction. In *Proceedings of the 38th Annual Conference of the Cognitive Science Society*. Cognitive Science.

[8] Vijay Konda and John Tsitsiklis. 2000. Actor-Critic Algorithms. In *Advances in Neural Information Processing Systems*, Vol. 12. MIT Press, 1008–1014.

[9] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.

[10] Volodymyr Mnih, Adrià Puigdomènech Badia, Mehdi Mirza, Alex Graves, Tim Harley, Timothy P. Lillicrap, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of the 33rd International Conference on Machine Learning - Volume 48* (New York, NY, USA) *(ICML'16)*. International Conference on Machine Learning, 1928–1937.

[11] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei Rusu, Joel Veness, Marc Bellemare, Alex Graves, Martin Riedmiller, Andreas Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (02 2015), 529–33. https://doi.org/10.1038/nature14236

[12] Richard Sutton and Andrew Barto. 1998. *Reinforcement Learning: An Introduction*. MIT Press.

[13] Ardi Tampuu, Tambet Matiisen, Dorian Kodelja, Ilya Kuzovkin, Kristjan Korjus, Jaan Aru, Juhan Aru, and Raul Vicente. 2017. Multi-Agent Cooperation and Competition with Deep Reinforcement Learning. In *PLOS ONE Journal*, Vol. 12. PLOS ONE.

[14] Oriol Vinyals, Igor Babuschkin, Wojciech Czarnecki, and et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* 575 (2019), 350–354.

[15] Kyle Wray, Akshat Kumar, and Shlomo Zilberstein. 2018. Integrated Cooperation and Competition in Multi-Agent Decision-Making. In *Proceedings of 32nd AAAI Conference on Artificial Intelligence 2018*. AAAI Conference on Artificial Intelligence, 4735–4742.