A Policy Gradient Algorithm for Learning to Learn in Multiagent Reinforcement Learning

Dong-Ki Kim [1 2] **Miao Liu** [2 3] **Matthew Riemer** [2 3] **Chuangchuang Sun** [1 2] **Marwa Abdulhai** [1 2] **Golnaz Habibi** [1 2] **Sebastian Lopez-Cot** [1 2] **Gerald Tesauro** [2 3] **Jonathan P. How** [1 2]

Abstract

A fundamental challenge in multiagent reinforcement learning is to learn beneficial behaviors in a shared environment with other simultaneously learning agents. In particular, each agent perceives the environment as effectively non-stationary due to the changing policies of other agents. Moreover, each agent is itself constantly learning, leading to natural non-stationarity in the distribution of experiences encountered. In this paper, we propose a novel meta-multiagent policy gradient theorem that directly accounts for the non-stationary policy dynamics inherent to multiagent learning settings. This is achieved by modeling our gradient updates to consider both an agent's own non-stationary policy dynamics and the non-stationary policy dynamics of other agents in the environment. We show that our theoretically grounded approach provides a general solution to the multiagent learning problem, which inherently comprises all key aspects of previous state of the art approaches on this topic. We test our method on a diverse suite of multiagent benchmarks and demonstrate a more efficient ability to adapt to new agents as they learn than baseline methods across the full spectrum of mixed incentive, competitive, and cooperative domains.

1. Introduction

Learning in multiagent settings is inherently more difficult than single-agent learning because an agent interacts both with the environment and other agents (Buşoniu et al., 2010). Specifically, the fundamental challenge in multiagent reinforcement learning (MARL) is the difficulty of learning optimal policies in the presence of other simultaneously learning agents because their changing behaviors jointly affect the environment's transition and reward function. This dependence on non-stationary policies renders the Markov property invalid from the perspective of each agent, requiring agents to adapt their behaviors with respect to potentially large, unpredictable, and endless changes in the policies of fellow agents (Papoudakis et al., 2019). In such environments, it is also critical that agents adapt to the changing behaviors of others in a sample-efficient manner as it is likely that their policies could update again after a small number of interactions. Therefore, effective agents should consider the learning of other agents and adapt quickly to their non-stationary behaviors. Otherwise, undesirable outcomes may arise when an agent is constantly lagging in its ability to deal with the current policies of other agents.

Our paper proposes a new framework based on meta-learning to address the inherent non-stationarity of MARL. Meta-learning (also referred to as learning to learn) was recently shown to be a promising methodology for fast adaptation in multiagent settings. The framework by Al-Shedivat et al. (2018), for example, introduces a meta-optimization scheme in which a meta-agent can adapt more efficiently to changes in a new opponent's policy after collecting only a handful of interactions. The key idea is to model the meta-agent's own learning process so that its updated policy performs better than an evolving opponent. However, prior work does not directly consider the learning processes of other agents during the meta-optimization process, instead treating the other agents as external factors and assuming the meta-agent cannot influence their future policies. As a result, prior work on multiagent meta-learning fails to consider an important property of MARL: other agents in the environment are also learning and adapting their own policies based on interactions with the meta-agent. Thus, the meta-agent is missing an opportunity to influence the others' future policies through these interactions, which could be used to improve its own performance.

Our contributions. With this insight, we make the following primary contributions in this paper:

1) **New theorem:** We derive a new *meta-multiagent policy gradient theorem (Meta-MAPG)* that, for the first time,

[1]MIT-LIDS [2]MIT-IBM Watson AI Lab [3]IBM-Research. Correspondence to: Dong-Ki Kim <dkkim93@mit.edu>, Miao Liu <miao.liu1@ibm.com>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021.

directly models the learning processes of all agents in the environment within a single objective function. We achieve this by extending past work that developed the meta-policy gradient theorem (Meta-PG) in the context of a single agent multi-task RL setting (Al-Shedivat et al., 2018) to the full generality of a multiagent game where every agent learns with a Markovian update function.

2) **Theoretical analysis and comparisons:** Our analysis of Meta-MAPG reveals an inherently added term, not present in Al-Shedivat et al. (2018), closely related to the process of shaping the other agents' learning dynamics in the framework of Foerster et al. (2018a). As such, our work contributes a theoretically grounded framework that unifies the collective benefits of previous work by Al-Shedivat et al. (2018) and Foerster et al. (2018a). Moreover, we formally demonstrate that Meta-PG can be considered a special case of Meta-MAPG if we assume the learning of the other agents in the environment is constrained to be independent of the meta-agent behavior. However, this limiting assumption does not typically hold in practice, and we demonstrate with a motivating example that it can lead to a total divergence of learning that is directly addressed by the correction we derive for the meta-gradient.

3) **Empirical evaluation:** We evaluate Meta-MAPG on a diverse suite of multiagent domains, including the full spectrum of mixed incentive, competitive, and cooperative environments. Our experiments demonstrate that, in contrast with previous state-of-the-art approaches on this topic, Meta-MAPG consistently results in a superior ability to adapt in the presence of novel agents as they learn.

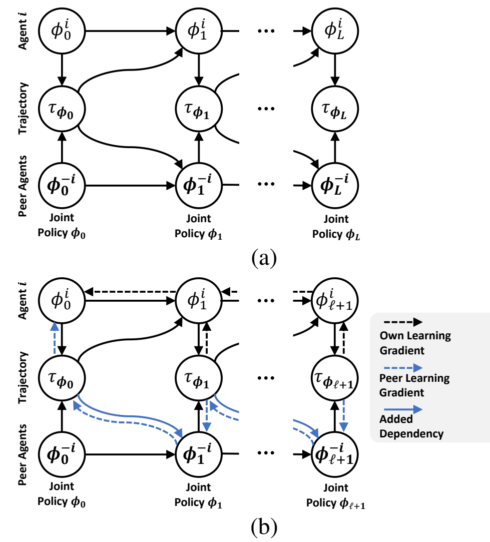


Figure 1. **(a)** A Markov chain of joint policies representing the inherent non-stationarity of MARL. Each agent updates its policy leveraging a Markovian update function, resulting in a change to the joint policy. **(b)** A probabilistic graph for Meta-MAPG. Unlike Meta-PG, our approach actively influences the future policies of other agents as well through the peer learning gradient.

2. Understanding Non-Stationarity in MARL

In MARL, when the policies of all other agents in the environment are stationary, they can be considered to be effectively a part of the environment from the perspective of each agent, resulting in a stationary single agent Markov decision process for each agent to learn from. However, in practice, the other agents in the environment are not fixed, but rather always learning from their recent experiences. As we detail in this section, this results in the inherent non-stationarity of MARL because Markovian update functions for each agent induce a Markov chain of joint policies. Indeed, when the policies of other agents constantly change, the problem is effectively non-stationary from each agent's perspective if the other agents are viewed as part of the environment.

2.1. Stochastic Games

Interactions between multiple agents can be represented by stochastic games (Shapley, 1953). Specifically, an n-agent stochastic game is defined as a tuple $\mathcal{M}_n = \langle \boldsymbol{\mathcal{I}}, \boldsymbol{\mathcal{S}}, \boldsymbol{\mathcal{A}}, \mathcal{P}, \boldsymbol{\mathcal{R}}, \gamma \rangle$; $\boldsymbol{\mathcal{I}} = \{1, \ldots, n\}$ is the set of n agents, $\boldsymbol{\mathcal{S}}$ is the set of states, $\boldsymbol{\mathcal{A}} = \times_{i \in \mathcal{I}} \mathcal{A}^i$ is the set of action spaces, $\mathcal{P}: \boldsymbol{\mathcal{S}} \times \boldsymbol{\mathcal{A}} \mapsto \boldsymbol{\mathcal{S}}$ is the state transition probability function, $\boldsymbol{\mathcal{R}} = \times_{i \in \mathcal{I}} \mathcal{R}^i$ is the set of reward functions, and $\gamma \in [0, 1)$ is the discount factor. We typeset sets in bold for clarity. Each agent i executes an action at each timestep t according to its stochastic policy $a_t^i \sim \pi^i(a_t^i | s_t, \phi^i)$ parameterized by ϕ^i, where $s_t \in \boldsymbol{\mathcal{S}}$. A joint action $\boldsymbol{a_t} = \{a_t^i, \boldsymbol{a_t^{-i}}\}$ yields a transition from the current state s_t to the next state $s_{t+1} \in \boldsymbol{\mathcal{S}}$ with probability $\mathcal{P}(s_{t+1} | s_t, \boldsymbol{a_t})$, where the notation $\boldsymbol{-i}$ indicates all other agents with the exception of agent i. Agent i then obtains a reward according to its reward function $r_t^i = \mathcal{R}^i(s_t, \boldsymbol{a_t})$. At the end of an episode, the agents collect a trajectory $\tau_{\boldsymbol{\phi}}$ under the joint policy with parameters $\boldsymbol{\phi}$, where $\tau_{\boldsymbol{\phi}} := (s_0, \boldsymbol{a_0}, \boldsymbol{r_0}, \ldots, \boldsymbol{r_H})$, $\boldsymbol{\phi} = \{\phi^i, \boldsymbol{\phi^{-i}}\}$ represents the joint parameters of all policies, $\boldsymbol{r_t} = \{r_t^i, \boldsymbol{r_t^{-i}}\}$ is the joint reward, and H is the episode horizon.

2.2. A Markov Chain of Joint Policies

The perceived non-stationarity in multiagent settings results from a distribution of sequential joint policies, which can be represented by a Markov chain. Formally, a Markov chain of policies begins from a stochastic game between agents with an initial set of joint policies parameterized by $\boldsymbol{\phi_0} = \{\phi_0^i, \boldsymbol{\phi_0^{-i}}\}$. We assume that each agent updates its policy leveraging a Markovian update function $\phi_1^i \sim U^i(\tau_{\boldsymbol{\phi_0}}, \phi_0^i)$ that changes the policy after every K trajectories. After this time period, each agent i updates its policy in order to maximize the expected return expressed as its value function:

$$V_{\boldsymbol{\phi_0}}^i(s_0) = \mathbb{E}_{p(\tau_{\boldsymbol{\phi_0}}|\boldsymbol{\phi_0})}\Big[\sum_{t=0}^{H} \gamma^t r_t^i | s_0\Big] = \mathbb{E}_{p(\tau_{\boldsymbol{\phi_0}}|\boldsymbol{\phi_0})}\Big[G^i(\tau_{\boldsymbol{\phi_0}})\Big], \quad (1)$$

where G^i denotes agent i's discounted return from the beginning of an episode with initial state s_0. The joint policy update results in a transition from $\boldsymbol{\phi_0}$ to the updated set of joint parameters $\boldsymbol{\phi_1}$. The Markov chain continues for a maximum chain length of L (see Figure 1a). This Markov chain perspective highlights the following inherent aspects of the experienced non-stationarity:

1) **Sequential dependency:** The future joint policy parameters $\boldsymbol{\phi_{1:L}} = \{\boldsymbol{\phi_1}, \ldots, \boldsymbol{\phi_L}\}$ sequentially depend on $\boldsymbol{\phi_0}$ since a change in $\tau_{\boldsymbol{\phi_0}}$ results in a change in $\boldsymbol{\phi_1}$, which in turn affects $\tau_{\boldsymbol{\phi_1}}$ and all successive joint policy updates up to $\boldsymbol{\phi_L}$.

2) **Controllable levels of non-stationarity:** As in Al-Shedivat et al. (2018) and Foerster et al. (2018a), we assume stationary policies during the collection of K trajectories, and that the joint policy update happens afterward. In such a setting, it is possible to control the non-stationarity by adjusting the K and H hyperparameters: smaller K and H increase the rate that agents change their policies, leading to a higher degree of non-stationarity in the environment.

3. Learning to Learn in MARL

This section explores learning policies that can adapt quickly to non-stationarity in the policies of other agents in the environment. To achieve this, we leverage meta-learning and devise a new *meta-multiagent policy gradient theorem* that exploits the inherent sequential dependencies of MARL discussed in the previous section. Specifically, our meta-agent addresses this non-stationarity by considering its current policy's impact on its own adapted policies while actively influencing the future policies of other agents as well by inducing changes to the distribution of trajectories. In this section, we first outline the meta-optimization objective of MARL and then derive our policy gradient theorem to optimize this objective. Finally, we discuss how to interpret our meta-policy gradient theorem with respect to prior work.

3.1. Gradient Based Meta-Optimization in MARL

We formalize the meta-objective of MARL as optimizing meta-agent i's initial policy parameters ϕ_0^i so that it maximizes the expected adaptation performance over a Markov chain of policies drawn from a stationary initial distribution of policies for the other agents $p(\boldsymbol{\phi_0^{-i}})$:

$$\max_{\phi_0^i} \mathbb{E}_{p(\boldsymbol{\phi_0^{-i}})} \Big[\sum_{\ell=0}^{L-1} V^i_{\boldsymbol{\phi_{0:\ell+1}}}(s_0, \phi_0^i) \Big], \tag{2}$$

$$V^i_{\boldsymbol{\phi_{0:\ell+1}}}(s_0,\phi_0^i) = \mathbb{E}_{p(\tau_{\boldsymbol{\phi_{0:\ell}}}|\boldsymbol{\phi_{0:\ell}})}\Big[\mathbb{E}_{p(\tau_{\boldsymbol{\phi_{\ell+1}}}|\boldsymbol{\phi_{\ell+1}})}\big[G^i(\tau_{\boldsymbol{\phi_{\ell+1}}})\big]\Big],$$

where $\tau_{\boldsymbol{\phi_{0:\ell}}} = \{\tau_{\boldsymbol{\phi_0}}, \ldots, \tau_{\boldsymbol{\phi_\ell}}\}$ and $V^i_{\boldsymbol{\phi_{0:\ell+1}}}(s_0, \phi_0^i)$ denotes the meta-value function. This meta-value function generalizes the notion of each agent's primitive value function for the current set of policies $V^i_{\boldsymbol{\phi_0}}(s_0)$ over the length of the Markov chain of policies. In this work, we assume that the Markov chain of policies is governed by a policy gradient update function that corresponds to what is generally referred to as the inner-loop optimization in the literature:

$$\phi^i_{\ell+1} = U^i(\tau_{\boldsymbol{\phi_\ell}}, \phi^i_\ell) := \phi^i_\ell + \alpha^i \nabla_{\phi^i_\ell} \mathbb{E}_{p(\tau_{\boldsymbol{\phi_\ell}}|\boldsymbol{\phi_\ell})} \Big[G^i(\tau_{\boldsymbol{\phi_\ell}})\Big], \tag{3}$$

$$\boldsymbol{\phi^{-i}_{\ell+1}} = U^{-i}(\tau_{\boldsymbol{\phi_\ell}}, \phi^{-i}_\ell) := \boldsymbol{\phi^{-i}_\ell} + \boldsymbol{\alpha^{-i}} \nabla_{\boldsymbol{\phi^{-i}_\ell}} \mathbb{E}_{p(\tau_{\boldsymbol{\phi_\ell}}|\boldsymbol{\phi_\ell})}\Big[\boldsymbol{G^{-i}}(\tau_{\boldsymbol{\phi_\ell}})\Big],$$

where α^i and $\boldsymbol{\alpha^{-i}}$ denote the learning rates.

3.2. The Meta-Multiagent Policy Gradient Theorem

A meta-agent needs to account for both its own learning process and the learning processes of other peer agents in the environment to fully address the inherent non-stationarity of MARL. We will now demonstrate that our generalized meta-policy gradient includes terms that explicitly account for the effect a meta-agent's current policy will have on its own adapted future policies as well as the future policies of peers interacting with it.

Theorem 1. (Meta-Multiagent Policy Gradient Theorem) *For any stochastic game $\mathcal{M}_n$, the gradient of the meta-value function for agent i at state s_0 with respect to current policy parameters ϕ_0^i evolving in the environment along with other peer agents using initial parameters $\boldsymbol{\phi_0^{-i}}$ is:*

$$\begin{aligned} \nabla_{\phi_0^i} V^i_{\boldsymbol{\phi_{0:\ell+1}}}(s_0,\phi_0^i) = \mathbb{E}_{p(\tau_{\boldsymbol{\phi_{0:\ell}}}|\boldsymbol{\phi_{0:\ell}})}\Big[\mathbb{E}_{p(\tau_{\boldsymbol{\phi_{\ell+1}}}|\boldsymbol{\phi_{\ell+1}})}\big[G^i(\tau_{\boldsymbol{\phi_{\ell+1}}}) \\ \Big(\underbrace{\nabla_{\phi_0^i} \log\pi(\tau_{\boldsymbol{\phi_0}}|\phi_0^i)}_{\text{Current Policy}} + \underbrace{\textstyle\sum_{\ell'=0}^{\ell} \nabla_{\phi_0^i} \log\pi(\tau_{\boldsymbol{\phi_{\ell'+1}}}|\phi^i_{\ell'+1})}_{\text{Own Learning}} + \\ \underbrace{\textstyle\sum_{\ell'=0}^{\ell} \nabla_{\phi_0^i} \log\pi(\tau_{\boldsymbol{\phi_{\ell'+1}}}|\boldsymbol{\phi^{-i}_{\ell'+1}})}_{\text{Peer Learning}}\Big)\big]\Big]. \end{aligned} \tag{4}$$

Proof. See Appendix A for a detailed proof. ☐

In particular, Meta-MAPG has three primary terms. The first term corresponds to the standard policy gradient with respect to the current policy parameters used during the initial trajectory. Meanwhile, the second term $\nabla_{\phi_0^i} \log \pi(\tau_{\boldsymbol{\phi_{\ell'+1}}}|\phi^i_{\ell'+1})$ explicitly differentiates through $\log \pi(\tau_{\boldsymbol{\phi_{\ell'+1}}}|\phi^i_{\ell'+1})$ with respect to ϕ_0^i. This enables a meta-agent to model its own learning dynamics and account for the impact of ϕ_0^i on its eventual adapted parameters $\phi^i_{\ell'+1}$. By contrast, the last term $\nabla_{\phi_0^i} \log \pi(\tau_{\boldsymbol{\phi_{\ell'+1}}}|\boldsymbol{\phi^{-i}_{\ell'+1}})$ aims to additionally compute gradients through the sequential dependency between the meta-agent's initial policy ϕ_0^i and the future policies of other agents in the environment $\boldsymbol{\phi^{-i}_{1:\ell+1}}$. As a result, the peer learning term enables a meta-agent to learn to change $\tau_{\boldsymbol{\phi_0}}$ in a way that influences the future policies of other agents and improves its adaptation performance over the Markov chain of policies.

Interestingly, the peer learning term that naturally arises when taking the gradient in Meta-MAPG is similar to a term previously considered in the literature by Foerster et al.

(2018a). However, for the Learning with Opponent Learning Awareness (LOLA) approach (Foerster et al., 2018a), this term was derived in an alternate way following a first order Taylor approximation with respect to the value function. Moreover, the own learning term previously appeared in the derivation of Meta-PG (Al-Shedivat et al., 2018). Indeed, it is quite surprising to see how taking a principled policy gradient while leveraging a more general set of assumptions leads to a unification of the benefits of key prior work (Al-Shedivat et al., 2018; Foerster et al., 2018a) on adjusting to the learning behavior of other agents in MARL.

3.3. Connection to the Meta-Policy Gradient Theorem

The framework of Al-Shedivat et al. (2018) derived the meta-policy gradient theorem for optimizing a setup like this. However, it is important to note that they derived this gradient while making the implicit assumption to ignore the sequential dependence of the future parameters of other agents on ϕ_0^i, resulting in the missing peer learning term:

$$\nabla_{\phi_0^i} V^i_{\boldsymbol{\phi_{0:\ell+1}}}(s_0,\phi_0^i) = \mathbb{E}_{p(\tau_{\boldsymbol{\phi_{0:\ell}}}|\boldsymbol{\phi_{0:\ell}})}\Big[\mathbb{E}_{p(\tau_{\boldsymbol{\phi_{\ell+1}}}|\boldsymbol{\phi_{\ell+1}})}\big[G^i(\tau_{\boldsymbol{\phi_{\ell+1}}}) \big(\underbrace{\nabla_{\phi_0^i}\log\pi(\tau_{\boldsymbol{\phi_0}}|\phi_0^i)}_{\text{Current Policy}} + \underbrace{\textstyle\sum_{\ell'=0}^{\ell}\nabla_{\phi_0^i}\log\pi(\tau_{\boldsymbol{\phi_{\ell'+1}}}|\phi^i_{\ell'+1})}_{\text{Own Learning}}\big)\big]\Big]. \quad (5)$$

Remark 1. *Meta-PG can be considered as a special case of Meta-MAPG when assuming that other agents' learning in the environment is independent of the meta-agent's behavior.*

Proof. See Appendix B for a corollary. □

Specifically, Meta-PG treats other agents as if they are external factors whose learning it cannot affect. For example, a meta-agent in Al-Shedivat et al. (2018) competes against an opponent that has been pre-trained with self-play. The next agent after the current interaction is then loaded as the same opponent with one more step of training with self-play. As a result of this contrived setup, the opponent's policy update is based on data collected without a meta-agent presented in the environment and Meta-PG can get away with assuming that the peer's learning process is not a function of a meta-agent's behavior. However, we note that this assumption does not hold in practice for the general multi-agent learning settings we explore in this work. As highlighted in Section 2, every agent's policy jointly affects the environment's transition and reward functions, resulting in the inherent sequential dependencies of MARL. Therefore, a meta-agent's behavior can affect the future policies of its peers, and these cannot be considered as external factors.

Probabilistic model perspective. Probabilistic models for Meta-PG and Meta-MAPG are depicted in Figure 1b. As shown by the own learning gradient direction, a meta-agent i optimizes ϕ_0^i by accounting for the impact of ϕ_0^i on its updated parameters $\boldsymbol{\phi^i_{1:\ell+1}}$ and adaptation performance $G^i(\tau_{\boldsymbol{\phi_{\ell+1}}})$. However, Meta-PG considers the other

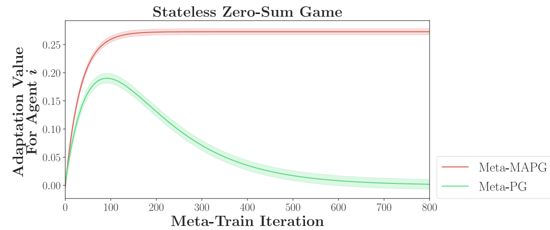


Figure 2. Meta-training result in the stateless zero-sum game when training a meta-agent i with either Meta-PG (Al-Shedivat et al., 2018) or Meta-MAPG. Meta-MAPG achieves better adaptation than Meta-PG thanks to the peer learning gradient. Mean and 95% confidence interval computed for 200 random samples are shown.

agents as external factors that cannot be influenced by the meta-agent, as indicated by the absence of the sequential dependence between $\tau_{\boldsymbol{\phi_{0:\ell}}}$ and $\boldsymbol{\phi^{-i}_{1:\ell+1}}$ in Figure 1b. As a result, the meta-agent loses an opportunity to influence the future policies of other agents and further improve its adaptation performance, which can cause undesirable learning failures as highlighted by the following motivating example.

Example 1. *Failure to consider our influence on the learning of other agents can result in biased and sometimes even counterproductive optimization.*

For example, consider a stateless zero-sum game played between two agents based off the one shown by Letcher et al. (2019). A meta-agent i and an opponent j maximize simple value functions $V^i_{\boldsymbol{\phi_\ell}} = \phi^i_\ell \phi^j_\ell$ and $V^j_{\boldsymbol{\phi_\ell}} = -\phi^i_\ell \phi^j_\ell$ respectively, where $\phi^i_\ell, \phi^j_\ell \in \mathbb{R}$. We assume a maximum Markov chain length L of 1 and focus on meta-agent i's adaptation performance $V^i_{\boldsymbol{\phi_1}}$. Given an opponent's initial policy parameters randomly sampled from $p(\boldsymbol{\phi_0^{-i}})$, we compare training a meta-agent i with either Meta-PG or Meta-MAPG. As Figure 2 shows, Meta-PG performs biased updates that actually make its performance worse over the course of meta-training. In contrast, by considering the opponent's learning process, Meta-MAPG corrects for this bias, displaying smooth convergence to a winning strategy. See Appendix C for more details, including the meta-gradient derivation.

3.4. Algorithm

We provide pseudo-code for Meta-MAPG in Algorithm 1 for meta-training and Algorithm 2 for meta-testing in Appendix D. Note that Meta-MAPG is centralized during meta-training as it requires the policy parameters of other agents to compute the peer learning gradient. For settings where a meta-agent cannot access the policy parameters of other agents during meta-training, we provide a decentralized meta-training algorithm with opponent modeling, motivated by the approach used in Foerster et al. (2018a), in Appendix D.1 that computes the peer learning gradient while leveraging only an approximation of the parameters of peer agents. Once meta-trained in either case, the adaptation to new agents during meta-testing is purely decentralized such that the meta-agent can decide how to shape other agents with its own observations and rewards alone.

4. Related Work

The standard approach for addressing non-stationarity in MARL is to consider information about the other agents and reason about the effects of their joint actions (Hernandez-Leal et al., 2017). The literature on opponent modeling, for instance, infers opponents' behaviors and conditions an agent's policy on the inferred behaviors of others (He et al., 2016; Raileanu et al., 2018; Grover et al., 2018). Studies regarding the centralized training with decentralized execution framework (Lowe et al., 2017; Foerster et al., 2018b; Yang et al., 2018; Wen et al., 2019), which accounts for the behaviors of others through a centralized critic, can also be classified into this category. While this body of work alleviates non-stationarity, it is generally assumed that each agent will have a stationary policy in the future. Because other agents can have different behaviors in the future as a result of learning (Foerster et al., 2018a), this incorrect assumption can cause sample inefficient and improper adaptation. In contrast, Meta-MAPG models each agent's learning process, allowing a meta-learning agent to adapt efficiently.

Our approach is also related to prior work that considers the learning of other agents in the environment. This includes Zhang & Lesser (2010) who attempted to discover the best response adaptation to the anticipated future policy of other agents. Our work is also related, as discussed previously, to LOLA (Foerster et al., 2018a) and more recent improvements (Foerster et al., 2018c). Another relevant idea explored by Letcher et al. (2019) is to interpolate between the frameworks of Zhang & Lesser (2010) and Foerster et al. (2018a) in a way that guarantees convergence while influencing the opponent's future policy. However, all of these approaches only account for the learning processes of other agents and fail to consider an agent's own non-stationary policy dynamics as in the own learning gradient discussed in the previous section. Additionally, these papers do not leverage meta-learning. As a result, these approaches may require many samples to properly adapt to new agents.

Meta-learning (Schmidhuber, 1987; Bengio et al., 1992) has recently become very popular as a method for improving sample efficiency in the presence of changing tasks in the deep RL literature (Wang et al., 2016a; Duan et al., 2016b; Finn et al., 2017; Mishra et al., 2017; Nichol & Schulman, 2018). See Vilalta & Drissi (2002) and Hospedales et al. (2020) for in-depth surveys of meta-learning. In particular, our work builds on the popular model-agnostic meta-learning framework (Finn et al., 2017), where gradient-based learning is used both for conducting so called inner-loop learning and to improve this learning by computing gradients through the computational graph. When we train our agents so that the inner loop can accommodate for a dynamic Markov chain of other agent policies, we are leveraging an approach that has recently become popular for supervised learning called meta-continual learning (Riemer et al., 2019; Javed & White, 2019; Spigler, 2019; Beaulieu et al., 2020; Caccia et al., 2020; Gupta et al., 2020). This means that our agent trains not just to adapt to a single set of policies during meta-training, but rather to adapt to a set of changing policies with Markovian updates. As a result, we avoid an issue of past work (Al-Shedivat et al., 2018) that required the use of importance sampling during meta-testing (see Appendix F.1 for more details).

5. Experiments

We demonstrate Meta-MAPG's efficacy on a diverse suite of domains, including the full spectrum of mixed incentive, competitive, and cooperative settings. The code is available at https://git.io/JZsfg, and video highlights are available at http://bit.ly/2L8Wvch. The mean and 95% confidence interval computed for 10 seeds are shown in each figure.

5.1. Experimental Setup for Meta-Learning in MARL

Training protocol. In our experiments, we evaluate a meta-agent i's adaptability with respect to a peer agent j learning in the environment. Specifically, agent j is sampled from a population of initial peer policy parameters $p(\boldsymbol{\phi}_{\mathbf{0}}^{-\boldsymbol{i}})$. Then, both agents adapt their behavior following the policy gradient with a linear feature baseline (Duan et al., 2016a) (see Figure 3). Importantly, the population captures diverse behaviors of j, such as varying expertise in solving a task, and j's initial policy is hidden to i. Hence, a meta-agent i should: 1) adapt to a differently initialized agent j with varying behaviors and 2) continuously adapt with respect to j's changing policy.

During meta-training, a meta-agent i interacts with peers drawn from the distribution and optimizes its initial policy parameters ϕ_0^i. At the end of meta-training, a new peer j is sampled from the distribution, and we measure i's performance throughout the Markov chain.

Adaptation baselines. We compare Meta-MAPG with the following baseline adaptation strategies for an agent i:

1) Meta-PG (Al-Shedivat et al., 2018): A meta-learning approach that only considers how to improve its own learning. We detail our implementation of Meta-PG and a low-level difference with the original method in Appendix F.1.

2) LOLA-DiCE (Foerster et al., 2018c): An approach that only considers how to shape the learning dynamics of other agents in the environment through the Differentiable Monte-Carlo Estimator (DiCE) operation. Note that LOLA-DiCE is an extension of the original LOLA approach.

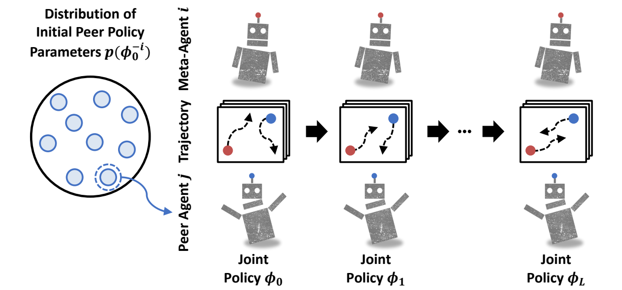


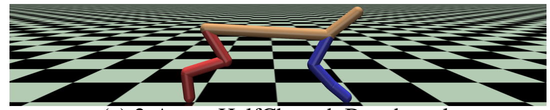
Figure 3. Experimental setup for meta-learning. A peer j's policy is initialized randomly from a population $p(\boldsymbol{\phi_0^{-i}})$ and is updated based on the policy gradient optimization, requiring a meta-agent i to adapt to differently initialized j and its changing policy.

Agent i \ Agent j	C	D
C	(0.5, 0.5)	(-1.5, 1.5)
D	(1.5, -1.5)	(-0.5, -0.5)

(a) IPD payoff table

Agent i \ Agent j	R	P	S
R	(0, 0)	(-1, 1)	(1, -1)
P	(1, -1)	(0, 0)	(-1, 1)
S	(-1, 1)	(1, -1)	(0, 0)

(b) RPS payoff table



(c) 2-Agent HalfCheetah Benchmark

Figure 4. **(a)** IPD payoff table. **(b)** RPS payoff table. **(c)** 2-Agent HalfCheetah benchmark, where two agents are coupled within the robot and control the robot together: the red and blue agent control three joints of the back and front leg, respectively.

3) REINFORCE (Williams, 1992): A simple policy gradient approach that considers neither an agent's own learning nor the learning processes of other agents. This baseline represents multiagent approaches that assume each agent leverages a stationary policy in the future.

Implementation. We implement each adaptation method's policy leveraging an LSTM and use the generalized advantage estimation (Schulman et al., 2016) with a learned value function for the meta-optimization. We also apply DiCE during the inner-loop optimization (Foerster et al., 2018c) to compute the peer learning gradients, and we learn dynamic inner-loop learning rates during meta-training as suggested in Al-Shedivat et al. (2018). We refer to Appendices E, F, G, I for the remaining details including hyperparameters.

Experiment complexity. We study various aspects of Meta-MAPG based on two repeated matrix games (see Figures 4a and 4b). We note that the repeated matrix domains we consider match the environment complexity of previous work on the topic of peer learning awareness in MARL (Foerster et al., 2018a;c; Letcher et al., 2019). We also considered the RoboSumo domain from Al-Shedivat et al. (2018). However, the training performance was very sensitive to the relative weights between the shaped reward functions and we could not reproduce the results from Al-Shedivat et al. (2018) with the default weights. Instead, we experimented with another challenging environment leveraging the multi-agent MuJoCo benchmark (de Witt et al., 2020) to demonstrate the scability of Meta-MAPG. Specifically, the environment shown in Figure 4c has high complexity: 1) the domain has continuous and large observation/action spaces, and 2) the agents are coupled within the same robot, resulting in a more challenging control problem than controlling the robot alone with full autonomy.

5.2. IPD Environment: Mixed Incentive

Question 1. *Is it essential to consider both an agent's own learning and the learning of others?*

To address this question, we consider the classic iterated prisoner's dilemma (IPD) domain. In IPD, agents i and j act by either (C)ooperating or (D)efecting and receive rewards according to the mixed incentive payoff defined in Figure 4a. As in Foerster et al. (2018a), we model the state space as $s_0 = \varnothing$ and $s_t = \boldsymbol{a_{t-1}}$ for $t \geq 1$. For meta-learning, we construct a population of initial personas $p(\boldsymbol{\phi_0^{-i}})$ that include cooperating personas (i.e., having a probability of cooperating between 0.5 and 1.0 at any state) and defecting personas (i.e., having a probability of cooperating between 0 and 0.5 at any state). Figure 7a in Appendix shows the population distribution utilized for training and evaluation.

The adaptation performance during meta-testing when an agent i, meta-trained with either Meta-MAPG or the baseline methods, interacts with an initially cooperating or defecting agent j is shown in Figure 5a and Figure 5b, respectively. In both cases, our meta-agent successfully infers the underlying persona of j and adapts throughout the Markov chain obtaining higher rewards than the baselines. We observe that performance generally decreases as the number of joint policy updates increases across all adaptation methods. This decrease in performance is expected as each model is playing with another reasonable agent that is also constantly learning. As a result, j realizes it could potentially achieve more reward by defecting more often. Hence, to achieve good adaptation performance in IPD, an agent i should attempt to shape j's future policies toward staying cooperative as long as possible such that i can take advantage, which is achieved by accounting for both an agent's own learning and the learning of other peer agents in Meta-MAPG.

We explore each adaptation method in more detail by visualizing the action probability dynamics throughout the Markov chain. In general, we observe that the baseline methods have converged to initially defecting strategies, attempting to get larger rewards than a peer agent j in the first trajectory $\tau_{\boldsymbol{\phi_0}}$. While this strategy can result in better initial performance than j, the peer agent will quickly change its policy so that it is defecting with high probability as well (see Figures 9 to 11 in Appendix). By contrast, our meta-agent learns to act cooperatively in $\tau_{\boldsymbol{\phi_0}}$ and then take advantage by deceiving agent j as it attempts to cooperate at future steps (see Figure 12 in Appendix).

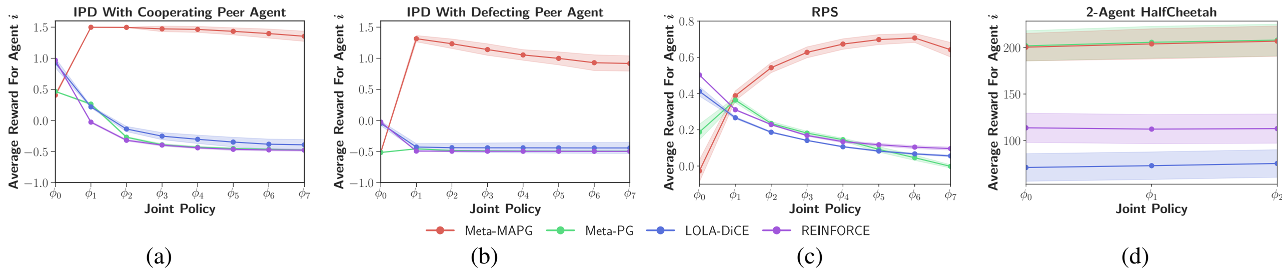


Figure 5. Adaptation performance during meta-testing in mixed incentive ((**a**) and (**b**)), competitive (**c**), and cooperative (**d**) environments. The results show that Meta-MAPG can successfully adapt to a new and learning peer agent throughout the Markov chain.

Question 2. *How is adaptation performance affected by the number of trajectories between changes?*
We control the level of non-stationarity by adjusting the number of trajectories K between updates (refer to Section 2.2). The results in Figure 6a shows that the area under the curve (AUC) (i.e., the reward summation during $\boldsymbol{\phi_{1:L}}$) generally decreases when K decreases in IPD. This result is expected since the inner-loop updates are based on the policy gradient, which can suffer from a high variance. Thus, with a smaller batch size, policy updates have a higher variance and lead to noisier policy updates. As a result, it is harder to anticipate and influence the future policies of peers. Nevertheless, Meta-MAPG achieves the best AUC in all cases.

Question 3. *How does Meta-MAPG perform with decentralized meta-training?*
We compare the performance of Meta-MAPG with and without opponent modeling (OM) in Figure 6b. We note that Meta-MAPG with opponent modeling can infer policy parameters for peer agents and compute the peer learning gradient in a decentralized manner, performing better than the Meta-PG baseline. However, opponent modeling introduces noise in predicting the future policy parameters of peer agents because the parameters must be inferred by observing the actions they take alone without any supervision about the parameters themselves. Thus, as expected, a meta-agent experiences difficulty in correctly considering the learning process of a peer, which leads to lower performance than Meta-MAPG with centralized meta-training.

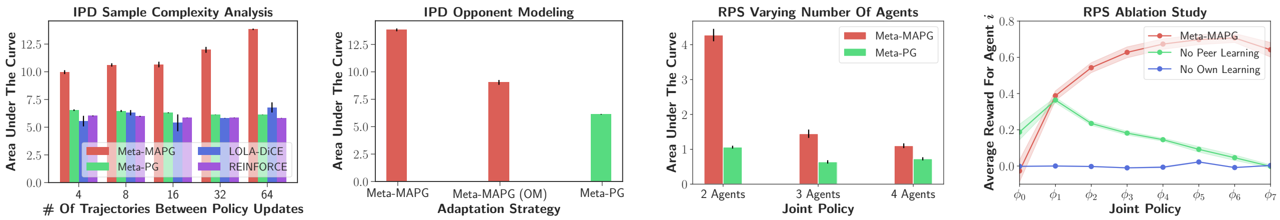
Question 4. *Can Meta-MAPG generalize its learning outside the meta-training distribution?*
We have demonstrated that a meta-agent can generalize well and adapt to a new peer. However, we would like to investigate this further and see whether a meta-agent can still perform well when the meta-testing distribution is drawn from a significantly different distribution in IPD. We thus evaluate Meta-MAPG and Meta-PG using both in distribution (as in the previous questions) and out of distribution personas for j's initial policies (see Figures 7a and 7b in Appendix). Meta-MAPG achieves an AUC of 13.77 ± 0.25 and 11.12 ± 0.33 for the in and out of distribution evaluation, respectively. Meta-PG achieves an AUC of 6.13 ± 0.05 and 7.60 ± 0.07 for the in and out of distribution evaluation, respectively. Variances are based on 5 seeds and we leveraged $K = 64$ for this experiment. We note that Meta-MAPG's performance decreases during the out of distribution evaluation, but still consistently performs better than the baseline.

5.3. RPS Environment: Competitive

Question 5. *How effective is Meta-MAPG in a fully competitive scenario?*
We have demonstrated the benefit of our approach in the mixed incentive scenario of IPD. Here, we consider another classic iterated game, rock-paper-scissors (RPS) with a fully competitive payoff table (see Figure 4b). In RPS, at each time step agents i and j can choose an action of either (R)ock, (P)aper, or (S)cissors. The state space is defined as $s_0 = \varnothing$ and $s_t = \boldsymbol{a_{t-1}}$ for $t \geq 1$. For meta-learning, we consider a population of initial personas $p(\boldsymbol{\phi_0^{-i}})$, including the rock/paper/scissors persona (with a rock/paper/scissors actions probability between $1/3$ and 1).

Figure 5c shows the adaptation performance during meta-testing. Similar to the IPD results, we observe that the baseline methods have effectively converged to win against the opponent j in the first few trajectories. For instance, agent i has a high rock probability when playing against j with a high initial scissors probability (see Figures 13 to 15 in Appendix). This strategy, however, results in the opponent quickly changing its behavior toward the mixed Nash equilibrium strategy of $(1/3, 1/3, 1/3)$ for the rock, paper, and scissors probabilities. In contrast, our meta-agent learned to lose slightly in the first two trajectories $\tau_{\boldsymbol{\phi_{0:1}}}$ to achieve much larger rewards in the later trajectories $\tau_{\boldsymbol{\phi_{2:7}}}$ while relying on its ability to adapt more efficiently than its opponent (see Figure 16 in Appendix). Compared to the IPD results, we observe that it is more difficult for our meta-agent to shape j's future policies in RPS possibly due to the fact that RPS has a fully competitive payoff structure, while IPD has a mixed incentive structure.

Question 6. *How effective is Meta-MAPG in settings with more than one peer?*
Our meta-multiagent policy gradient theorem is general and can be applied to scenarios with more than one peer. To validate this, we experiment with 3-player and 4-player RPS, where we consider sampling peers randomly from the entire persona population. We note that the domain becomes



(a) (b) (c) (d)

Figure 6. (**a**) Adaptation performance with a varying number of trajectories. Meta-MAPG's performance generally improves with a larger K. (**b**) Adaptation performance with opponent modeling. Meta-MAPG with OM computes the peer learning gradient in a decentralized manner. (**c**) Adaptation performance with a varying number of agents, where Meta-MAPG achieves the best AUC in all cases. (**d**) Ablation study for Meta-MAPG. Meta-MAPG achieves significantly better performance than the two ablated baselines.

harder as the number of agents n increases, because the state space dimension grows exponentially as a function of n and a meta-agent needs to consider more peers in shaping their future policies. Figure 6c shows a comparison against the Meta-PG baseline. We generally observe that the peer agents change their policies toward the mixed Nash equilibrium more quickly as the number of agents increases, which results in decreased performance for all methods. Nevertheless, Meta-MAPG achieves the best performance in all cases and can clearly be easily extended to settings with a greater number of agents.

Question 7. *What happens if a meta-agent is trained without the own learning gradient?*

Our meta-multiagent policy gradient theorem inherently includes both the own learning and peer learning gradient, but is it necessary to have both terms? To answer this question, we conduct an ablation study and compare Meta-MAPG to two methods: one trained without the peer learning gradient and another trained without the own learning gradient. Note that not having the peer learning term is equivalent to Meta-PG, and not having the own learning term is similar to LOLA-DiCE but alternatively trained with a meta-optimization procedure. Figure 6d shows that a meta-agent trained without the peer learning term cannot properly exploit the peer agent's learning process. Also, a meta-agent trained without the own learning term cannot change its own policy effectively in response to anticipated learning by peer agents. By contrast, Meta-MAPG achieves superior performance by accounting for both its own learning process and the learning processes of peer agents.

5.4. 2-Agent HalfCheetah Environment: Cooperative

Question 8. *Does considering the peer learning gradient always improve performance?*

To answer this question, we experiment with a fully cooperative setting of the the 2-Agent HalfCheetah domain (de Witt et al., 2020), where the first and second agent control three joints of the back and front leg with continuous action spaces, respectively (see Figure 4c). Both agents receive a joint reward corresponding to making the cheetah robot run to the right as soon as possible. Note that two agents are coupled *within* the cheetah robot, requiring close cooperation and coordination between them.

For meta-learning, we consider a population of teammates with varying degrees of expertise in running to the *left* direction. Specifically, we pre-train a teammate j and build a population based on checkpoints of its parameters during learning. Figure 8 in Appendix shows the distribution of j's expertise. Note that we construct the distribution with a significant gap to ensure that the meta-testing distribution is sufficiently different from the meta-training/validation distribution. This allows us to effectively evaluate the ability of our approach to generalize its learning. During meta-learning, j is randomly initialized from this population of policies. Importantly, j must adapt its behavior in this setting because the agent has achieved the *opposite* skill during pre-training compared to the true objective of moving to the right. Hence, a meta-agent i should succeed by both adapting to differently initialized teammates with varying expertise in moving the opposite direction, and guiding j's learning process to coordinate the eventual right movement.

Our results are displayed in Figure 5d. There are two notable observations. First, influencing peer learning does not help much in cooperative settings and Meta-MAPG performs similarly to Meta-PG. The peer learning gradient attempts to shape the future policies of other agents so that the meta-agent can take advantage. In IPD, for example, the meta-agent influenced j to be cooperative in the future such that the meta-agent can act with a high probability of the defect action and receive higher returns. However, in cooperative settings, due to the joint reward, j is already changing its policies in order to benefit the meta-agent, resulting in a less significant effect with respect to the peer learning gradient. Second, Meta-PG and Meta-MAPG outperform the other approaches of LOLA-DiCE and REINFORCE, achieving higher rewards when interacting with a new teammate.

6. Conclusion

In this paper, we have introduced Meta-MAPG, a meta-learning algorithm that can adapt quickly to non-stationarity in the policies of other agents in a shared environment. The key idea underlying our proposed meta-optimization is to directly model both an agent's own learning process and the non-stationary policy dynamics of other agents. We evaluated our method on several multiagent benchmarks and showed that Meta-MAPG is able to adapt more efficiently than previous state of the art approaches. We hope that our work can help provide the community with a theoretical foundation to build off for addressing the inherent non-stationarity of MARL in a principled manner.

Acknowledgements

Research funded by IBM, Samsung (as part of the MIT-IBM Watson AI Lab initiative) and computational support through Amazon Web Services. Dong-Ki Kim was also supported by a Kwanjeong Educational Foundation Fellowship. The authors would like to thank anonymous reviewers for their helpful comments.

References

Al-Shedivat, M., Bansal, T., Burda, Y., Sutskever, I., Mordatch, I., and Abbeel, P. Continuous adaptation via meta-learning in nonstationary and competitive environments. In *International Conference on Learning Representations (ICLR)*, 2018. URL https://openreview.net/forum?id=Sk2u1g-0-.

Beaulieu, S., Frati, L., Miconi, T., Lehman, J., Stanley, K. O., Clune, J., and Cheney, N. Learning to continually learn. *arXiv preprint arXiv:2002.09571*, 2020. URL https://arxiv.org/abs/2002.09571.

Bengio, S., Bengio, Y., Cloutier, J., and Gecsei, J. On the optimization of a synaptic learning rule. In *Preprints Conf. Optimality in Artificial and Biological Neural Networks*, volume 2. Univ. of Texas, 1992.

Buşoniu, L., Babuška, R., and De Schutter, B. *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.

Caccia, M., Rodriguez, P., Ostapenko, O., Normandin, F., Lin, M., Caccia, L., Laradji, I., Rish, I., Lacoste, A., Vazquez, D., et al. Online fast adaptation and knowledge accumulation: a new approach to continual learning. *Neural Information Processing Systems (NeurIPS)*, 2020.

de Witt, C. S., Peng, B., Kamienny, P.-A., Torr, P., Böhmer, W., and Whiteson, S. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. *arXiv preprint arXiv:2003.06709*, 2020.

Duan, Y., Chen, X., Houthooft, R., Schulman, J., and Abbeel, P. Benchmarking deep reinforcement learning for continuous control. volume 48 of *Proceedings of Machine Learning Research*, pp. 1329–1338, New York, New York, USA, 20–22 Jun 2016a. PMLR. URL http://proceedings.mlr.press/v48/duan16.html.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. RL2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016b.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Foerster, J., Chen, R. Y., Al-Shedivat, M., Whiteson, S., Abbeel, P., and Mordatch, I. Learning with opponent-learning awareness. In *International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*, pp. 122–130, 2018a.

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., and Whiteson, S. Counterfactual multi-agent policy gradients. In *Association for the Advancement of Artificial Intelligence (AAAI)*, February 2018b. URL http://www.cs.ox.ac.uk/people/shimon.whiteson/pubs/foersteraaai18.pdf.

Foerster, J., Farquhar, G., Al-Shedivat, M., Rocktäschel, T., Xing, E., and Whiteson, S. DiCE: The infinitely differentiable Monte Carlo estimator. In *International Conference on Machine Learning (ICML)*, volume 80, pp. 1524–1533, 10–15 Jul 2018c. URL http://proceedings.mlr.press/v80/foerster18a.html.

Grover, A., Al-Shedivat, M., Gupta, J., Burda, Y., and Edwards, H. 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.

Gupta, G., Yadav, K., and Paull, L. La-maml: Look-ahead meta learning for continual learning. 2020. URL https://arxiv.org/abs/2007.13904.

He, H., Boyd-Graber, J., Kwok, K., and III, H. D. 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.

Hernandez-Leal, P., Kaisers, M., Baarslag, T., and de Cote, E. M. A survey of learning in multiagent environments: Dealing with non-stationarity. *CoRR*, abs/1707.09183, 2017. URL http://arxiv.org/abs/1707.09183.

Hospedales, T., Antoniou, A., Micaelli, P., and Storkey, A. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

Javed, K. and White, M. Meta-learning representations for continual learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Neural Information Processing Systems (NeurIPS)*, pp. 1818–1828. Curran Associates, Inc., 2019.

Letcher, A., Foerster, J., Balduzzi, D., Rocktäschel, T., and Whiteson, S. Stable opponent shaping in differentiable games. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=SyGjjsC5tQ.

Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, O. P., and Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Neural Information Processing Systems (NeurIPS)*, pp. 6382–6393, 2017.

Mishra, N., Rohaninejad, M., Chen, X., and Abbeel, P. A simple neural attentive meta-learner. *arXiv preprint arXiv:1707.03141*, 2017.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., and Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016. URL http://proceedings.mlr.press/v48/mniha16.html.

Nichol, A. and Schulman, J. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2018.

Papoudakis, G., Christianos, F., Rahman, A., and Albrecht, S. V. Dealing with non-stationarity in multi-agent deep reinforcement learning. *CoRR*, abs/1906.04737, 2019. URL http://arxiv.org/abs/1906.04737.

Raileanu, R., Denton, E., Szlam, A., and Fergus, R. 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.

Riemer, M., Cases, I., Ajemian, R., Liu, M., Rish, I., Tu, Y., and Tesauro, G. Learning to learn without forgetting by maximizing transfer and minimizing interference. *International Conference on Learning Representations (ICLR)*, 2019.

Schmidhuber, J. Evolutionary principles in self-referential learning. on learning now to learn: The meta-meta-meta...-hook. Diploma thesis, Technische Universitat Munchen, Germany, 14 May 1987. URL http://www.idsia.ch/~juergen/diploma.html.

Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P. High-dimensional continuous control using generalized advantage estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Shapley, L. S. Stochastic games. *Proceedings of the National Academy of Sciences*, 39(10):1095–1100, 1953. ISSN 0027-8424. doi: 10.1073/pnas.39.10.1095. URL https://www.pnas.org/content/39/10/1095.

Spigler, G. Meta-learnt priors slow down catastrophic forgetting in neural networks. *arXiv preprint arXiv:1909.04170*, 2019. URL https://arxiv.org/pdf/1909.04170.pdf.

Sutton, R. S. and Barto, A. G. *Introduction to Reinforcement Learning*. MIT Press, Cambridge, MA, USA, 1st edition, 1998. ISBN 0262193981.

Vilalta, R. and Drissi, Y. A perspective view and survey of meta-learning. *Artificial intelligence review*, 18(2):77–95, 2002.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016a.

Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., and de Freitas, N. Sample efficient actor-critic with experience replay. *CoRR*, abs/1611.01224, 2016b. URL http://arxiv.org/abs/1611.01224.

Wei, E., Wicke, D., Freelan, D., and Luke, S. Multiagent soft Q-learning. *CoRR*, abs/1804.09817, 2018. URL http://arxiv.org/abs/1804.09817.

Wen, Y., Yang, Y., Luo, R., Wang, J., and Pan, W. Probabilistic recursive reasoning for multi-agent reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2019. URL https://openreview.net/forum?id=rkl6As0cF7.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, May 1992. ISSN 1573-0565. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W., and Wang, J. 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.

Yu, T., Kumar, S., Gupta, A., Levine, S., Hausman, K., and Finn, C. Gradient surgery for multi-task learning. *arXiv preprint arXiv:2001.06782*, 2020.

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