Multi-Agent Reinforcement Learning: A Selective Overview of Theories and Algorithms

Kaiqing Zhang[♮] Zhuoran Yang[†] Tamer Başar[♮]

Abstract

Recent years have witnessed significant advances in reinforcement learning (RL), which has registered great success in solving various sequential decision-making problems in machine learning. Most of the successful RL applications, e.g., the games of Go and Poker, robotics, and autonomous driving, involve the participation of more than one single agent, which naturally fall into the realm of multi-agent RL (MARL), a domain with a relatively long history, and has recently re-emerged due to advances in single-agent RL techniques. Though empirically successful, theoretical foundations for MARL are relatively lacking in the literature. In this chapter, we provide a selective overview of MARL, with focus on algorithms backed by theoretical analysis. More specifically, we review the theoretical results of MARL algorithms mainly within two representative frameworks, Markov/stochastic games and extensive-form games, in accordance with the types of tasks they address, i.e., fully cooperative, fully competitive, and a mix of the two. We also introduce several significant but challenging applications of these algorithms. Orthogonal to the existing reviews on MARL, we highlight several new angles and taxonomies of MARL theory, including learning in extensive-form games, decentralized MARL with networked agents, MARL in the mean-field regime, (non-)convergence of policy-based methods for learning in games, etc. Some of the new angles extrapolate from our own research endeavors and interests. Our overall goal with this chapter is, beyond providing an assessment of the current state of the field on the mark, to identify fruitful future research directions on theoretical studies of MARL. We expect this chapter to serve as continuing stimulus for researchers interested in working on this exciting while challenging topic.

1 Introduction

Recent years have witnessed sensational advances of reinforcement learning (RL) in many prominent sequential decision-making problems, such as playing the game of Go (Silver

[♮]Department of Electrical and Computer Engineering & Coordinated Science Laboratory, University of Illinois at Urbana-Champaign, 1308 West Main, Urbana, IL, 61801, USA. Email: {kzhang66, basar1}@illinois.edu. Writing of this chapter was supported in part by the US Army Research Laboratory (ARL) Cooperative Agreement W911NF-17-2-0196, and in part by the Air Force Office of Scientific Research (AFOSR) Grant FA9550-19-1-0353.

[†]Department of Operations Research and Financial Engineering, Princeton University, 98 Charlton St, Princeton, NJ, 08540, USA. Email: zy6@princeton.edu.

et al., 2016, 2017), playing real-time strategy games (OpenAI, 2018; Vinyals et al., 2019), robotic control (Kober et al., 2013; Lillicrap et al., 2016), playing card games (Brown and Sandholm, 2017, 2019), and autonomous driving (Shalev-Shwartz et al., 2016), especially accompanied with the development of deep neural networks (DNNs) for function approximation (Mnih et al., 2015). Intriguingly, most of the successful applications involve the participation of more than one single agent/player[1], which should be modeled systematically as multi-agent RL (MARL) problems. Specifically, MARL addresses the sequential decision-making problem of multiple autonomous agents that operate in a common environment, each of which aims to optimize its own long-term return by interacting with the environment and other agents (Busoniu et al., 2008). Besides the aforementioned popular ones, learning in multi-agent systems finds potential applications in other subareas, including cyber-physical systems (Adler and Blue, 2002; Wang et al., 2016), finance (O et al., 2002; Lee et al., 2007), sensor/communication networks (Cortes et al., 2004; Choi et al., 2009), and social science (Castelfranchi, 2001; Leibo et al., 2017).

Largely, MARL algorithms can be placed into three groups, *fully cooperative*, *fully competitive*, and *a mix of the two*, depending on the types of settings they address. In particular, in the cooperative setting, agents collaborate to optimize a common long-term return; while in the competitive setting, the return of agents usually sum up to zero. The mixed setting involves both cooperative and competitive agents, with general-sum returns. Modeling disparate MARL settings requires frameworks spanning from optimization theory, dynamic programming, game theory, and decentralized control, see §2.2 for more detailed discussions. In spite of these existing multiple frameworks, several challenges in MARL are in fact common across the different settings, especially for the theoretical analysis. Specifically, first, the learning goals in MARL are multidimensional, as the objective of all agents are not necessarily aligned, which brings up the challenge of dealing with equilibrium points, as well as some additional performance criteria beyond return-optimization, such as the efficiency of communication/coordination, and robustness against potential adversarial agents. Moreover, as all agents are improving their policies according to their own interests concurrently, the environment faced by each agent becomes *non-stationary*. This breaks or invalidates the basic framework of most theoretical analyses in the single-agent setting. Furthermore, the joint action space that increases exponentially with the number of agents may cause scalability issues, known as the *combinatorial nature* of MARL (Hernandez-Leal et al., 2018). Additionally, the information structure, i.e., *who knows what*, in MARL is more involved, as each agent has limited access to the observations of others, leading to possibly suboptimal decision rules locally. A detailed elaboration on the underlying challenges can be found in §3.

There has in fact been no shortage of efforts attempting to address the above challenges. See Busoniu et al. (2008) for a comprehensive overview of earlier theories and algorithms on MARL. Recently, this domain has gained resurgence of interest due to the advances of single-agent RL techniques. Indeed, a huge volume of work on MARL has appeared lately, focusing on either identifying new learning criteria and/or setups (Foerster et al., 2016; Zazo et al., 2016; Zhang et al., 2018; Subramanian and Mahajan, 2019), or developing new algorithms for existing setups, thanks to the development of deep learn-

[1] Hereafter, we will use *agent* and *player* interchangeably.

ing (Heinrich and Silver, 2016; Lowe et al., 2017; Foerster et al., 2017; Gupta et al., 2017; Omidshafiei et al., 2017; Kawamura et al., 2017; Zhang et al., 2019), operations research (Mazumdar and Ratliff, 2018; Jin et al., 2019; Zhang et al., 2019; Sidford et al., 2019), and multi-agent systems (Oliehoek and Amato, 2016; Arslan and Yüksel, 2017; Yongacoglu et al., 2019; Zhang et al., 2019). Nevertheless, not all the efforts are placed under rigorous theoretical footings, partly due to the limited understanding of even single-agent deep RL theories, and partly due to the inherent challenges in multi-agent settings. As a consequence, it is imperative to review and organize the MARL algorithms with theoretical guarantees, in order to highlight the boundary of existing research endeavors, and stimulate potential future directions on this topic.

In this chapter, we provide a selective overview of theories and algorithms in MARL, together with several significant while challenging applications. More specifically, we focus on two representative frameworks of MARL, namely, Markov/stochastic games and extensive-form games, in discrete-time settings as in standard single-agent RL. In conformity with the aforementioned three groups, we review and pay particular attention to MARL algorithms with convergence and complexity analysis, most of which are fairly recent. With this focus in mind, we note that our overview is by no means comprehensive. In fact, besides the classical reference Busoniu et al. (2008), there are several other reviews on MARL that have appeared recently, due to the resurgence of MARL (Hernandez-Leal et al., 2017, 2018; Nguyen et al., 2018; Oroojlooy Jadid and Hajinezhad, 2019). We would like to emphasize that these reviews provide views and taxonomies that are complementary to ours: Hernandez-Leal et al. (2017) surveys the works that are specifically devised to address *opponent-induced non-stationarity*, one of the challenges we discuss in §3; Hernandez-Leal et al. (2018); Nguyen et al. (2018) are relatively more comprehensive, but with the focal point on *deep* MARL, a subarea with scarce theories thus far; Oroojlooy Jadid and Hajinezhad (2019), on the other hand, focuses on algorithms in the *cooperative* setting only, though the review within this setting is extensive.

Finally, we spotlight several new angles and taxonomies that are comparatively underexplored in the existing MARL reviews, primarily owing to our own research endeavors and interests. First, we discuss the framework of extensive-form games in MARL, in addition to the conventional one of Markov games, or even simplified repeated games (Busoniu et al., 2008; Hernandez-Leal et al., 2017, 2018); second, we summarize the progresses of a recently boosting subarea: decentralized MARL with *networked* agents, as an extrapolation of our early works on this (Zhang et al., 2018,a,b); third, we bring about the *mean-field* regime into MARL, as a remedy for the case with an extremely large population of agents; fourth, we highlight some recent advances in optimization theory, which shed lights on the (non-)convergence of policy-based methods for MARL, especially zero-sum games. We have also reviewed some the literature on MARL in partially observed settings, but without using deep RL as heuristic solutions. We expect these new angles to help identify fruitful future research directions, and more importantly, inspire researchers with interests in establishing rigorous theoretical foundations on MARL.

Roadmap. The remainder of this chapter is organized as follows. In §2, we introduce the background of MARL: standard algorithms for single-agent RL, and the frameworks of MARL. In §3, we summarize several challenges in developing MARL theory, in addition

to the single-agent counterparts. A series of MARL algorithms, mostly with theoretical guarantees, are reviewed and organized in §4, according to the types of tasks they address. In §5, we briefly introduce a few recent successes of MARL driven by the algorithms mentioned, followed by conclusions and several open research directions outlined in §6.

2 Background

In this section, we provide the necessary background on reinforcement learning, in both single- and multi-agent settings.

2.1 Single-Agent RL

A reinforcement learning agent is modeled to perform sequential decision-making by interacting with the environment. The environment is usually formulated as a Markov decision process (MDP), which is formally defined as follows.

Definition 2.1. A *Markov decision process* is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, respectively; $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ for any given action $a \in \mathcal{A}$; $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that determines the immediate reward received by the agent for a transition from (s, a) to s'; $\gamma \in [0, 1]$ is the discount factor that trades off the instantaneous and future rewards.

As a standard model, MDP has been widely adopted to characterize the decision-making of an agent with *full observability* of the system state s.[2] At each time t, the agent chooses to execute an action a_t in face of the system state s_t, which causes the system to transition to $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$. Moreover, the agent receives an instantaneous reward $R(s_t, a_t, s_{t+1})$. The goal of solving the MDP is thus to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, a mapping from the state space $\mathcal{S}$ to the distribution over the action space $\mathcal{A}$, so that $a_t \sim \pi(\cdot|s_t)$ and the discounted accumulated reward

$$\mathbb{E}\left[\sum_{t\geq 0} \gamma^t R(s_t, a_t, s_{t+1}) \,\middle|\, a_t \sim \pi(\cdot|s_t), s_0\right]$$

is maximized. Accordingly, one can define the *state-action function/Q-function*, and *value function* under policy π as

$$Q_\pi(s, a) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t R(s_t, a_t, s_{t+1}) \,\middle|\, a_t \sim \pi(\cdot|s_t), a_0 = a, s_0 = s\right],$$

$$V_\pi(s) = \mathbb{E}\left[\sum_{t\geq 0} \gamma^t R(s_t, a_t, s_{t+1}) \,\middle|\, a_t \sim \pi(\cdot|s_t), s_0 = s\right]$$

[2]The partially observed MDP (POMDP) model is usually advocated when the agent has no access to the exact system state but only an *observation* of the state. See Monahan (1982); Cassandra (1998) for more details on the POMDP model.

for any $s \in \mathcal{S}$ and $a \in \mathcal{A}$, which are the discounted accumulated reward starting from $(s_0, a_0) = (s, a)$ and $s_0 = s$, respectively. The ones corresponding to the optimal policy π^* are usually referred to as the *optimal Q-function* and the *optimal value function*, respectively.

By virtue of the Markovian property, the optimal policy can be obtained by dynamic-programming/backward induction approaches, e.g., value iteration and policy iteration algorithms (Bertsekas, 2005), which require the knowledge of the transition probability and the form of reward function. Reinforcement learning, on the other hand, is to find such an optimal policy without knowing the model. The RL agent learns the policy from experiences collected by interacting with the environment. By and large, RL algorithms can be categorized into two mainstream types, *value-based* and *policy-based* methods.

2.1.1 Value-Based Methods

Value-based RL methods are devised to find a good estimate of the state-action value function, namely, the optimal Q-function Q_{π^*}. The (approximate) optimal policy can then be extracted by taking the greedy action of the Q-function estimate. One of the most popular value-based algorithms is Q-learning (Watkins and Dayan, 1992), where the agent maintains an estimate of the Q-value function $\hat{Q}(s, a)$. When transitioning from state-action pair (s, a) to next state s', the agent receives a payoff r and updates the Q-function according to:

$$\hat{Q}(s, a) \leftarrow (1 - \alpha)\hat{Q}(s, a) + \alpha\Big[r + \gamma \max_{a'} \hat{Q}(s', a')\Big], \tag{2.1}$$

where $\alpha > 0$ is the stepsize/learning rate. Under certain conditions on α, Q-learning can be proved to converge to the optimal Q-value function almost surely (Watkins and Dayan, 1992; Szepesvári and Littman, 1999), with discrete and finite state and action spaces. Moreover, when combined with neural networks for function approximation, deep Q-learning has achieved great empirical breakthroughs in human-level control applications (Mnih et al., 2015). Another popular *on-policy* value-based method is SARSA, whose convergence was established in Singh et al. (2000) for finite-space settings.

An alternative while popular value-based RL algorithm is Monte-Carlo tree search (MCTS) (Chang et al., 2005; Kocsis and Szepesvári, 2006; Coulom, 2006), which estimates the optimal value function by constructing a search tree via Monte-Carlo simulations. Tree polices that judiciously select actions to balance exploration-exploitation are used to build and update the search tree. The most common tree policy is to apply the UCB1 (UCB stands for *upper confidence bound*) algorithm, which was originally devised for stochastic multi-arm bandit problems (Agrawal, 1995; Auer et al., 2002), to each node of the tree. This yields the popular UCT algorithm (Kocsis and Szepesvári, 2006). Convergence guarantee of MCTS had not been fully characterized until very recently (Jiang et al., 2018; Shah et al., 2019).

Besides, another significant task regarding value functions in RL is to estimate the value function associated with a given policy (not only the optimal one). This task, usually referred to as *policy evaluation*, has been tackled by algorithms that follow a similar update as (2.1), named *temporal difference* (TD) learning (Tesauro, 1995; Tsitsiklis and Van Roy, 1997; Sutton and Barto, 2018). Some other common policy evaluation algorithms with

convergence guarantees include gradient TD methods with linear (Sutton et al., 2008, 2009; Liu et al., 2015), and nonlinear function approximations (Bhatnagar et al., 2009). See Dann et al. (2014) for a more detailed review on policy evaluation.

2.1.2 Policy-Based Methods

Another type of RL algorithms directly searches over the policy space, which is usually estimated by parameterized function approximators like neural networks, namely, approximate $\pi(\cdot|s) \approx \pi_\theta(\cdot|s)$. As a consequence, the most straightforward idea, which is to update the parameter along the gradient direction of the long-term reward, has been instantiated by the policy gradient (PG) method. As a key premise for the idea, the closed-form of PG is given as (Sutton et al., 2000)

$$\nabla J(\theta) = \mathbb{E}_{a\sim\pi_\theta(\cdot|s), s\sim\eta_{\pi_\theta}(\cdot)}\Big[Q_{\pi_\theta}(s,a)\nabla \log \pi_\theta(a|s)\Big], \tag{2.2}$$

where $J(\theta)$ and Q_{π_θ} are the expected return and Q-function under policy π_θ, respectively, $\nabla \log \pi_\theta(a|s)$ is the score function of the policy, and η_{π_θ} is the state occupancy measure, either discounted or ergodic, under policy π_θ. Then, various policy gradient methods, including REINFORCE (Williams, 1992), G(PO)MDP (Baxter and Bartlett, 2001), and actor-critic algorithms (Konda and Tsitsiklis, 2000; Bhatnagar et al., 2009), have been proposed by estimating the gradient in different ways. A similar idea also applies to deterministic policies in continuous-action settings, whose PG has been derived by Silver et al. (2014). Besides gradient-based ones, several other policy optimization methods have achieved state-of-the-art performance in many applications, including PPO (Schulman et al., 2017), TRPO (Schulman et al., 2015), and soft actor-critic (Haarnoja et al., 2018).

Compared with value-based methods, policy-based ones enjoy better convergence guarantees (Konda and Tsitsiklis, 2000; Yang et al., 2018; Zhang et al., 2019; Agarwal et al., 2019), especially with neural networks for function approximation (Liu et al., 2019; Wang et al., 2019), which can readily handle massive or even continuous state-action spaces.

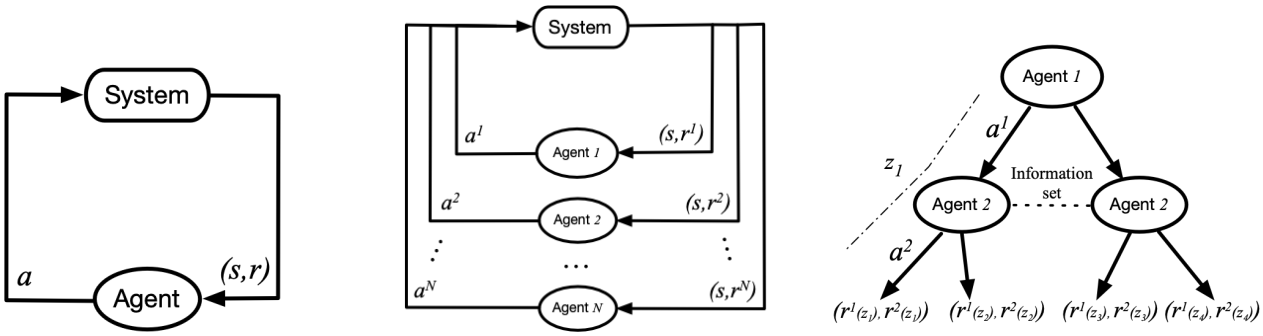
2.2 Multi-Agent RL Framework

In a similar vein, multi-agent RL also addresses sequential decision-making problems, but with more than one agent involved. In particular, both the evolution of the system state and the reward received by each agent are influenced by the joint actions of all agents. More intriguingly, each agent has its own long-term reward to optimize, which now becomes a function of the policies of all other agents. Such a general model finds broad applications in practice, see §5 for a detailed review of several significant ones.

In general, there exist two seemingly different but closely related theoretical frameworks for MARL, Markov/stochastic games and extensive-form games, as to be introduced next. Evolution of the systems under different frameworks are illustrated in Figure 1.

2.2.1 Markov/Stochastic Games

One direct generalization of MDP that captures the intertwinement of multiple agents is Markov games (MGs), also known as stochastic games (Shapley, 1953). Originated from



(a) MDP (b) Markov game (c) Extensive-form game

Figure 1: Schematic diagrams for the system evolution of a Markov decision process, a Markov game, and an extensive-form game, which correspond to the frameworks for single- and multi-agent RL, respectively. Specifically, in an MDP as in (a), the agent observes the state s and receives reward r from the system, after outputting the action a; in an MG as in (b), all agents choose actions a^i simultaneously, after observing the system state s and receiving each individual reward r^i; in a two-player extensive-form game as in (c), the agents make decisions on choosing actions a^i alternately, and receive each individual reward $r^i(z)$ at the end of the game, with z being the terminal history. In the imperfect information case, player 2 is uncertain about where he/she is in the game, which makes the information set non-singleton.

the seminal work Littman (1994), the framework of MGs has long been used in the literature to develop MARL algorithms, see §4 for more details. We introduce the formal definition as below.

Definition 2.2. A *Markov game* is defined by a tuple $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma)$, where $\mathcal{N} = \{1, \cdots, N\}$ denotes the set of $N > 1$ agents, $\mathcal{S}$ denotes the state space observed by all agents, $\mathcal{A}^i$ denotes the action space of agent i. Let $\mathcal{A} := \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$, then $\mathcal{P} : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ denotes the transition probability from any state $s \in \mathcal{S}$ to any state $s' \in \mathcal{S}$ for any joint action $a \in \mathcal{A}$; $R^i : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is the reward function that determines the immediate reward received by agent i for a transition from (s, a) to s'; $\gamma \in [0, 1]$ is the discount factor.

At time t, each agent $i \in \mathcal{N}$ executes an action a_t^i, according to the system state s_t. The system then transitions to state s_{t+1}, and rewards each agent i by $R^i(s_t, a_t, s_{t+1})$. The goal of agent i is to optimize its own long-term reward, by finding the policy $\pi^i : \mathcal{S} \to \Delta(\mathcal{A}^i)$ such that $a_t^i \sim \pi^i(\cdot|s_t)$. As a consequence, the value-function $V^i : \mathcal{S} \to \mathbb{R}$ of agent i becomes a function of the joint policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ defined as $\pi(a|s) := \prod_{i\in\mathcal{N}} \pi^i(a^i|s)$. In particular, for any joint policy π and state $s \in \mathcal{S}$,

$$V^i_{\pi^i, \pi^{-i}}(s) := \mathbb{E}\left[\sum_{t\geq 0} \gamma^t R^i(s_t, a_t, s_{t+1}) \,\middle|\, a_t^i \sim \pi^i(\cdot|s_t), s_0 = s \right], \tag{2.3}$$

where $-i$ represents the indices of all agents in $\mathcal{N}$ except agent i. Hence, the solution concept of MG deviates from that of MDP, since the *optimal* performance of each agent is controlled not only by its own policy, but also the choices of all other players of the game.

The most common solution concept, Nash equilibrium (NE), is defined as follows (Başar and Olsder, 1999).

Definition 2.3. A *Nash equilibrium* of the Markov game $(\mathcal{N}, \mathcal{S}, \{\mathcal{A}^i\}_{i\in\mathcal{N}}, \mathcal{P}, \{R^i\}_{i\in\mathcal{N}}, \gamma)$ is a joint policy $\pi^* = (\pi^{1,*}, \cdots, \pi^{N,*})$, such that for any $s \in \mathcal{S}$ and $i \in \mathcal{N}$

$$V^i_{\pi^{i,*},\pi^{-i,*}}(s) \geq V^i_{\pi^i,\pi^{-i,*}}(s), \qquad \text{for any } \pi^i.$$

Nash equilibrium characterizes an equilibrium point π^*, from which none of the agents has any incentive to deviate. In other words, for any agent $i \in \mathcal{N}$, the policy $\pi^{i,*}$ is the *best-response* of $\pi^{-i,*}$. As a standard learning goal for MARL, NE always exists for discounted MGs (Filar and Vrieze, 2012), but may not be unique in general. Most of the MARL algorithms are contrived to converge to such an equilibrium point.

The framework of Markov games is general enough to umbrella various MARL settings summarized below.

Cooperative Setting:

In a fully cooperative setting, all agents usually share a common reward function, i.e., $R^1 = R^2 = \cdots = R^N = R$. We note that this model is also referred to as *multi-agent MDPs* (MMDPs) in the AI community (Boutilier, 1996; Lauer and Riedmiller, 2000), and *Markov teams/team Markov games* in the control/game theory community (Yoshikawa, 1978; Ho, 1980; Wang and Sandholm, 2003; Mahajan, 2008). Moreover, from the game-theoretic perspective, this cooperative setting can also be viewed as a special case of Markov *potential* games (González-Sánchez and Hernández-Lerma, 2013; Zazo et al., 2016; Valcarcel Macua et al., 2018), with the potential function being the common accumulated reward. With this model in mind, the value function and Q-function are identical to all agents, which thus enables the single-agent RL algorithms, e.g., Q-learning update (2.1), to be applied, if all agents are coordinated as one decision maker. The global optimum for cooperation now constitutes a Nash equilibrium of the game.

Besides the common-reward model, another slightly more general and surging model for cooperative MARL considers *team-average* reward (Kar et al., 2013; Zhang et al., 2018; Doan et al., 2019). Specifically, agents are allowed to have different reward functions, which may be kept private to each agent, while the goal for cooperation is to optimize the long-term reward corresponding to the average reward $\bar{R}(s,a,s') := N^{-1} \cdot \sum_{i\in\mathcal{N}} R^i(s,a,s')$ for any $(s,a,s') \in \mathcal{S}\times\mathcal{A}\times\mathcal{S}$. The average-reward model, which allows more heterogeneity among agents, includes the model above as a special case. It also preserves privacy among agents, and facilitates the development of *decentralized* MARL algorithms (Kar et al., 2013; Zhang et al., 2018; Wai et al., 2018). Such heterogeneity also necessitates the incorporation of *communication* protocols into MARL, and the analysis of communication-efficient MARL algorithms.

Competitive Setting:

Fully competitive setting in MARL is typically modeled as *zero-sum* Markov games, namely, $\sum_{i\in\mathcal{N}} R^i(s,a,s') = 0$ for any (s,a,s'). For ease of algorithm design and analysis, most literature focused on *two* agents that compete against each other (Littman, 1994), where clearly the reward of one agent is exactly the loss of the other. In addition to direct

applications to game-playing (Littman, 1994; Silver et al., 2017; OpenAI, 2017), zero-sum games also serve as a model for *robust* learning, since the *uncertainty* that impedes the learning process of the agent can be accounted for as a fictitious opponent in the game that is always against the agent (Jacobson, 1973; Başar and Bernhard, 1995; Zhang et al., 2019). Therefore, the Nash equilibrium yields a robust policy that optimizes the *worst-case* long-term reward.

Mixed Setting:

Mixed setting is also known as the *general-sum* game setting, where no restriction is imposed on the goal and relationship among agents (Hu and Wellman, 2003; Littman, 2001). Each agent is self-interested, whose reward may be conflicting with others'. Equilibrium solution concepts from game theory, such as Nash equilibrium (Başar and Olsder, 1999), have the most significant influence on algorithms that are developed for this general setting. Furthermore, we include the setting with both fully cooperative and competitive agents, for example, two zero-sum competitive teams with cooperative agents in each team (Lagoudakis and Parr, 2003; Zhang et al., 2018b; OpenAI, 2018), as instances of the mixed setting as well.

2.2.2 Extensive-Form Games

Even though they constitute a classical formalism for MARL, Markov games can only handle the fully observed case, i.e., the agent has *perfect information* on the system state s_t and the executed action a_t at time t. Nonetheless, a plethora of MARL applications involve agents with only partial observability, i.e., *imperfect information* of the game. Extension of Markov games to partially observed case may be applicable, which, however, is challenging to solve, even under the cooperative setting (Oliehoek and Amato, 2016; Bernstein et al., 2002).[3]

In contrast, another framework, named *extensive-form games* (Osborne and Rubinstein, 1994; Shoham and Leyton-Brown, 2008), can handily model imperfect information for multi-agent decision-making. This framework is rooted in computational game theory and has been shown to admit polynomial-time algorithms under mild conditions (Koller and Megiddo, 1992). We formally introduce the framework of extensive-form games as follows.

Definition 2.4. An *extensive-form game* is defined by $(\mathcal{N} \cup \{c\}, \mathcal{H}, \mathcal{Z}, \mathcal{A}, \{R^i\}_{i\in\mathcal{N}}, \tau, \pi^c, \mathcal{S})$, where $\mathcal{N} = \{1, \ldots, N\}$ denotes the set of $N > 1$ agents, and c is a special agent called *chance* or *nature*, which has a fixed stochastic policy that specifies the randomness of the environment. Besides, $\mathcal{A}$ is the set of all possible actions that agents can take and $\mathcal{H}$ is the set of all possible *histories*, where each history is a sequence of actions taken from the beginning of the game. Let $\mathcal{A}(h) = \{a | ha \in \mathcal{H}\}$ denote the set of actions available after a nonterminal history h. Suppose an agent takes action $a \in \mathcal{A}(h)$ given history $h \in \mathcal{H}$, which then leads to a new history $ha \in \mathcal{H}$. Among all histories, $\mathcal{Z} \subseteq \mathcal{H}$ is a subset of *terminal histories* that represents the completion of a game. A utility is assigned to each agent $i \in \mathcal{N}$ at a terminal

[3] Partially observed Markov games under the cooperative setting are usually formulated as decentralized POMDP (Dec-POMDP) problems. See §4.1.3 for more discussions on this setting.

history, dictated by the function $R^i\colon \mathcal{Z} \to \mathbb{R}$. Moreover, $\tau : \mathcal{H} \to \mathcal{N} \cup \{c\}$ is the *identification* function that specifies which agent takes the action at each history. If $\tau(h) = c$, the chance agent takes an action a according to its policy π^c, i.e., $a \sim \pi^c(\cdot|h)$. Furthermore, $\mathcal{S}$ is the partition of $\mathcal{H}$ such that for any $s \in \mathcal{S}$ and any $h, h' \in s$, we have $\tau(h) = \tau(h')$ and $\mathcal{A}(h) = \mathcal{A}(h')$. In other words, histories h and h' in the same partition are indistinguishable to the agent that is about to take action, namely $\tau(h)$. The elements in $\mathcal{S}$ are referred to as *information states*.

Intuitively, the imperfect information of an extensive-form game is reflected by the fact that agents cannot distinguish between histories in the same information set. Since we have $\tau(h) = \tau(h')$ and $\mathcal{A}(h) = \mathcal{A}(h')$ for all $h, h' \in s$, for ease of presentation, in the sequel, for all $h \in s$, we let $\mathcal{A}(s)$ and $\tau(s)$ denote $\mathcal{A}(h)$ and $\tau(h)$, respectively. We also define a mapping $I\colon \mathcal{H} \to \mathcal{S}$ by letting $I(h) = s$ if $h \in s$. Moreover, we only consider games where both $\mathcal{H}$ and $\mathcal{A}$ are finite sets. To simplify the notation, for any two histories $h, h' \in \mathcal{H}$, we refer to h as a *prefix* of h', denoted by $h \sqsubseteq h'$, if h' can be reached from h by taking a sequence of actions. In this case, we call h' a *suffix* of h. Furthermore, we assume throughout that the game features *perfect recall*, which implies that each agent remembers the sequence of the information states and actions that have led to its current information state. The assumption of perfect recall is commonly made in the literature, which enables the existence of polynomial-time algorithms for solving the game (Koller and Megiddo, 1992). More importantly, by the celebrated Kuhn's theorem (Kuhn, 1953), under such an assumption, to find the set of Nash equilibria, it suffices to restrict the derivation to the set of *behavioral policies* which map each information set $s \in \mathcal{S}$ to a probability distribution over $\mathcal{A}(s)$. For any $i \in \mathcal{N}$, let $\mathcal{S}^i = \{s \in \mathcal{S}\colon \tau(s) = i\}$ be the set of information states of agent i. A joint policy of the agents is denoted by $\pi = (\pi^1, \ldots, \pi^N)$, where $\pi^i : \mathcal{S}^i \to \Delta(\mathcal{A}(s))$ is the policy of agent i. For any history h and any joint policy π, we define the *reach probability* of h under π as

$$\eta_\pi(h) = \prod_{h'\colon h'a \sqsubseteq h} \pi^{\tau(h')}(a|I(h')) = \prod_{i \in \mathcal{N} \cup \{c\}} \prod_{h'\colon h'a \sqsubseteq h, \tau(h') = i} \pi^i(a|I(h')), \tag{2.4}$$

which specifies the probability that h is created when all agents follow π. We similarly define the reach probability of an information state s under π as $\eta_\pi(s) = \sum_{h \in s} \eta_\pi(h)$. The expected utility of agent $i \in \mathcal{N}$ is thus given by $\sum_{z \in \mathcal{Z}} \eta_\pi(z) \cdot R^i(z)$, which is denoted by $R^i(\pi)$ for simplicity. Now we are ready to introduce the solution concept for extensive-form games, i.e., Nash equilibrium and its ϵ-approximation, as follows.

Definition 2.5. An *ϵ-Nash equilibrium* of an extensive-form game represented by $(\mathcal{N} \cup \{c\}, \mathcal{H}, \mathcal{Z}, \mathcal{A}, \{R^i\}_{i \in \mathcal{N}}, \tau, \pi^c, \mathcal{S})$ is a joint policy $\pi^* = (\pi^{1,*}, \cdots, \pi^{N,*})$, such that for any $i \in \mathcal{N}$,

$$R^i(\pi^{i,*}, \pi^{-i,*}) \geq R^i(\pi^i, \pi^{-i,*}) - \epsilon, \qquad \text{for any policy } \pi^i \text{ of agent } i.$$

Here π^{-i} denotes the joint policy of agents in $\mathcal{N} \setminus \{i\}$ where agent j adopts policy π^j for all $j \in \mathcal{N} \setminus \{i\}$. Additionally, if $\epsilon = 0$, π^* constitutes a *Nash equilibrium*.

Various Settings:

Extensive-form games are in general used to model non-cooperative settings. Specifically, zero-sum/constant-sum utility with $\sum_{i \in \mathcal{N}} R^i = k$ for some constant k corresponds

to the fully competitive setting; general-sum utility function results in the mixed setting. More importantly, settings of different information structures can also be characterized by extensive-form games. In particular, a *perfect information* game is one where each information set is a singleton, i.e., for any $s \in \mathcal{S}$, $|s| = 1$; an *imperfect information* game is one where there exists $s \in \mathcal{S}$, $|s| > 1$. In other words, with imperfect information, the information state s used for decision-making represents more than one possible history, and the agent cannot distinguish between them.

Among various settings, the zero-sum imperfect information setting has been the main focus of theoretical studies that bridge MARL and extensive-form games (Zinkevich et al., 2008; Heinrich et al., 2015; Srinivasan et al., 2018; Omidshafiei et al., 2019). It has also motivated MARL algorithms that revolutionized competitive setting applications like Poker AI (Rubin and Watson, 2011; Brown and Sandholm, 2019).

Connection to Markov Games:

Note that the two formalisms in Definitions 2.2 and 2.4 are connected. In particular, for simultaneous-move Markov games, the choices of actions by other agents are unknown to an agent, which thus leads to different histories that can be aggregated as one information state s. Histories in these games are then sequences of *joint* actions, and the discounted accumulated reward instantiates the utility at the end of the game. Conversely, by simply setting $\mathcal{A}^j = \emptyset$ at the state s for agents $j \neq \tau(s)$, the extensive-form game reduces to a Markov game with *state-dependent* action spaces. See Lanctot et al. (2019) for a more detailed discussion on the connection.

Remark 2.6 (Other MARL Frameworks)**.** Several other theoretical frameworks for MARL also exist in the literature, e.g., normal-form and/or repeated games (Claus and Boutilier, 1998; Bowling and Veloso, 2001; Kapetanakis and Kudenko, 2002; Conitzer and Sandholm, 2007), and partially observed Markov games (Hansen et al., 2004; Amato et al., 2013; Amato and Oliehoek, 2015). However, the former framework can be viewed as a special case of MGs, with a singleton state; most early theories of MARL in this framework have been restricted to small scale problems (Bowling and Veloso, 2001; Conitzer and Sandholm, 2007; Kapetanakis and Kudenko, 2002) only. MARL in the latter framework, on the other hand, is inherently challenging to address in general (Bernstein et al., 2002; Hansen et al., 2004), leading to relatively scarce theories in the literature. Due to space limitation, we do not introduce these models here in any detail. We will briefly review MARL algorithms under some of these models, especially the partially observed setting, in §4, though. Interested readers are referred to the early review Busoniu et al. (2008) for more discussions on MARL in normal-form/repeated games.

3 Challenges in MARL Theory

Despite a general model with broad applications, MARL suffers from several challenges in theoretical analysis, in addition to those that arise in single-agent RL. We summarize below the challenges that we regard as fundamental in developing theories for MARL.

3.1 Non-Unique Learning Goals

Unlike single-agent RL, where the goal of the agent is to maximize the long-term return efficiently, the learning goals of MARL can be vague at times. In fact, as argued in Shoham et al. (2003), the *unclarity* of the problems being addressed is the fundamental flaw in many early MARL works. Indeed, the goals that need to be considered in the analysis of MARL algorithms can be multi-dimensional. The most common goal, which has, however, been challenged in Shoham et al. (2003), is the convergence to Nash equilibrium as defined in §2.2. By definition, NE characterizes the point that no agent will deviate from, if any algorithm *finally* converges. This is undoubtedly a reasonable solution concept in game theory, under the assumption that the agents are all *rational*, and are capable of perfectly reasoning and infinite mutual modeling of agents. However, with *bounded rationality*, the agents may only be able to perform *finite* mutual modeling (Shoham and Leyton-Brown, 2008). As a result, the learning dynamics that are devised to converge to NE may not be justifiable for practical MARL agents. Instead, the goal may be focused on designing the best *learning strategy* for a given agent and *a fixed class of the other agents in the game*. In fact, these two goals are styled as *equilibrium agenda* and *AI agenda* in Shoham et al. (2003).

Besides, it has also been controversial that *convergence* (to the equilibrium point) is the dominant performance criterion for MARL algorithm analysis. In fact, it has been recognized in Zinkevich et al. (2006) that value-based MARL algorithms fail to converge to the *stationary* NE of general-sum Markov games, which motivated the new solution concept of *cyclic equilibrium* therein, at which the agents cycle rigidly through a set of stationary policies, i.e., not converging to any NE policy. Alternatively, Bowling and Veloso (2001, 2002) separate the learning goal into being both *stable* and *rational*, where the former ensures the algorithm to be convergent, given a predefined, targeted class of opponents' algorithms, while the latter requires the convergence to a best-response when the other agents remain stationary. If all agents are both stable and rational, convergence to NE naturally arises in this context. Moreover, the notion of *regret* introduces another angle to capture agents' rationality, which measures the performance of the algorithm compared to the best hindsight static strategy (Bowling and Veloso, 2001; Bowling, 2005). No-regret algorithms with asymptotically zero average regret guarantee the convergence to the equilibria of certain games (Hart and Mas-Colell, 2001; Bowling, 2005; Zinkevich et al., 2008), which essentially guarantee that the agent is not *exploited* by others.

In addition to the goals concerning optimizing the return, several other goals that are special to multi-agent systems have also drawn increasing attention. For example, Kasai et al. (2008); Foerster et al. (2016); Kim et al. (2019) investigate *learning to communicate*, in order for the agents to better coordinate. Such a concern on communication protocols has naturally inspired the recent studies on *communication-efficient* MARL (Chen et al., 2018; Lin et al., 2019; Ren and Haupt, 2019; Kim et al., 2019). Other important goals include how to learn without over-fitting certain agents (He et al., 2016; Lowe et al., 2017; Grover et al., 2018), and how to learn robustly with either malicious/adversarial or failed/dysfunctional learning agents (Gao et al., 2018; Li et al., 2019; Zhang et al., 2019). Still in their infancy, some works concerning aforementioned goals provide only empirical results, leaving plenty of room for theoretical studies.

3.2 Non-Stationarity

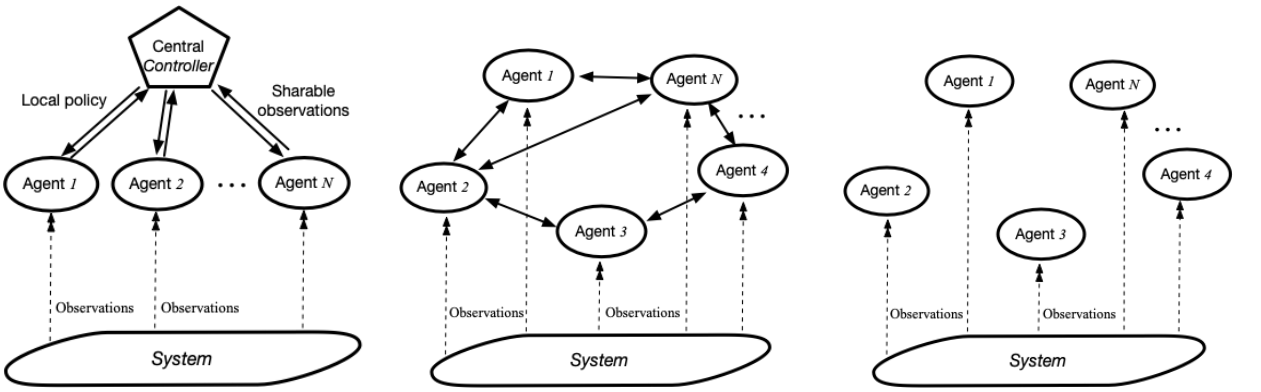
Another key challenge of MARL lies in the fact that multiple agents usually learn concurrently, causing the environment faced by each individual agent to be *non-stationary*. In particular, the action taken by one agent affects the reward of other opponent agents, and the evolution of the state. As a result, the learning agent is required to account for how the other agents behave and adapt to the *joint behavior* accordingly. This invalidates the stationarity assumption for establishing the convergence of single-agent RL algorithms, namely, the Markov property of the environment such that the individual reward and current state depend only on the previous state and action taken. This precludes the direct use of mathematical tools for single-agent RL analysis in MARL.

Indeed, theoretically, if the agent ignores this issue and optimizes its own policy assuming a stationary environment, which is usually referred to as an *independent learner*, the algorithms may fail to converge (Tan, 1993; Claus and Boutilier, 1998), except for several special settings (Arslan and Yüksel, 2017; Yongacoglu et al., 2019). Empirically, however, independent learning may achieve satisfiable performance in practice (Matignon et al., 2012; Foerster et al., 2017). As one of the most well-known issues in MARL, non-stationarity has long been recognized in the literature (Busoniu et al., 2008; Tuyls and Weiss, 2012). A recent comprehensive survey Hernandez-Leal et al. (2017) peculiarly provides an overview on how it is modeled and addressed by state-of-the-art multi-agent learning algorithms. We thus do not include more discussion on this challenge, and refer interested readers to Hernandez-Leal et al. (2017).

3.3 Scalability Issue

To handle non-stationarity, each individual agent may need to account for the *joint action space*, whose dimension increases exponentially with the number of agents. This is also referred to as the *combinatorial nature* of MARL (Hernandez-Leal et al., 2018). Having a large number of agents complicates the theoretical analysis, especially convergence analysis, of MARL. This argument is substantiated by the fact that theories on MARL for the two-player zero-sum setting are much more extensive and advanced than those for general-sum settings with more than two agents, see §4 for a detailed comparison. One possible remedy for the scalability issue is to assume additionally the *factorized* structures of either the value or reward functions with regard to the action dependence; see Guestrin et al. (2002a,b); Kok and Vlassis (2004) for the original heuristic ideas, and Sunehag et al. (2018); Rashid et al. (2018) for recent empirical progress. Relevant theoretical analysis had not been established until recently (Qu and Li, 2019), which considers a special dependence structure, and develops a provably convergent model-based (not RL) algorithm.

Another theoretical challenge of MARL that is brought about independently of, but worsened by, the scalability issue, is to build up theories for deep multi-agent RL. Particularly, scalability issues necessitate the use of function approximation, especially deep neural networks, in MARL. Though empirically successful, the theoretical analysis of deep MARL is an almost uncharted territory, with the currently limited understanding of deep learning theory, not alone the deep RL theory. This is included as one of the future research directions in §6.



(a) Centralized setting (b) Decentralized setting with networked agents (c) Fully decentralized setting

Figure 2: Three representative information structures in MARL. Specifically, in (a), there exists a central controller that can aggregate information from the agents, e.g., joint actions, joint rewards, and joint observations, and even design policies for all agents. The information exchanged between the central controller and the agents can thus include both some private observations from the agents, and the local policies designed for each agent from the controller. In both (b) and (c), there is no such a central controller, and are thus both referred to as decentralized structures. In (b), agents are connected via a possibly time-varying communication network, so that the local information can spread across the network, by information exchange with only each agent's neighbors. (b) is more common in cooperative MARL settings. In (c), the agents are full decentralized, with no explicit information exchange with each other. Instead, each agent makes decisions based on its local observations, without any coordination and/or aggregation of data. The local observations that vary across agents, however, may contain some global information, e.g., the joint actions of other agents, namely the control sharing information structure (Mahajan, 2013). Such a fully decentralized structure can also be found in many game-theoretic learning algorithms.

3.4 Various Information Structures

Compared to the single-agent case, the information structure of MARL, namely, *who knows what* at the training and execution, is more involved. For example, in the framework of Markov games, it suffices to observe the instantaneous state s_t, in order for each agent to make decisions, since the local policy π^i maps from $\mathcal{S}$ to $\Delta(\mathcal{A}^i)$. On the other hand, for extensive-form games, each agent may need to recall the history of past decisions, under the common perfect recall assumption. Furthermore, as self-interested agents, each agent can scarcely access either the *policy* or the rewards of the opponents, but at most the action samples taken by them. This partial information aggravates the issues caused by non-stationarity, as the samples can hardly recover the exact behavior of the opponents' underlying policies, which increases the non-stationarity viewed by individual agents. The extreme case is the aforementioned *independent learning* scheme, which assumes the observability of only the local action and reward, and suffers from non-convergence in general (Tan, 1993).

Learning schemes resulting from various information structures lead to various levels of difficulty for theoretical analysis. Specifically, to mitigate the partial information issue above, a great deal of work assumes the existence of a *central controller* that can collect information such as joint actions, joint rewards, and joint observations, and even design policies for all agents (Hansen et al., 2004; Oliehoek and Amato, 2014; Lowe et al., 2017; Foerster et al., 2017; Gupta et al., 2017; Foerster et al., 2018; Dibangoye and Buffet, 2018; Chen et al., 2018; Rashid et al., 2018). This structure gives birth to the popular learning scheme of *centralized-learning-decentralized-execution*, which stemmed from the works on planning for the partially observed setting, namely, Dec-POMDPs (Hansen et al., 2004; Oliehoek and Amato, 2014; Kraemer and Banerjee, 2016), and has been widely adopted in recent (deep) MARL works (Lowe et al., 2017; Foerster et al., 2017; Gupta et al., 2017; Foerster et al., 2018; Chen et al., 2018; Rashid et al., 2018). For cooperative settings, this learning scheme greatly simplifies the analysis, allowing the use of tools for single-agent RL analysis. Though, for non-cooperative settings with heterogeneous agents, this scheme does not significantly simplify the analysis, as the learning goals of the agents are not aligned, see §3.1.

Nonetheless, generally such a central controller does not exist in many applications, except the ones that can easily access a simulator, such as video games and robotics. As a consequence, a *fully decentralized* learning scheme is preferred, which includes the aforementioned independent learning scheme as a special case. To address the non-convergence issue in independent learning, agents are usually allowed to exchange/share local information with their neighbors over a communication network (Kar et al., 2013; Macua et al., 2015, 2017; Zhang et al., 2018,a,b; Lee et al., 2018; Wai et al., 2018; Doan et al., 2019; Suttle et al., 2019; Doan et al., 2019; Lin et al., 2019). We refer to this setting as *a decentralized one with networked agents*. Theoretical analysis for convergence is then made possible in this setting, the difficulty of which sits between that of single-agent RL and general MARL algorithms. Three different information structures are depicted in Figure 2.

4 MARL Algorithms with Theory

This section provides a selective review of MARL algorithms, and categorizes them according to the tasks to address. Exclusively, we review here the works that are focused on the theoretical studies only, which are mostly built upon the two representative MARL frameworks, fully observed Markov games and extensive-form games, introduced in §2.2. A brief summary on MARL for partially observed Markov games in *cooperative* settings, namely, the Dec-POMDP problems, is also provided below in §4.1, due to their relatively more mature theory than that of MARL for general partially observed Markov games.

4.1 Cooperative Setting

Cooperative MARL constitutes a great portion of MARL settings, where all agents collaborate with each other to achieve some shared goal. Most cooperative MARL algorithms backed by theoretical analysis are devised for the following more specific settings.

4.1.1 Homogeneous Agents

A majority of cooperative MARL settings involve *homogeneous* agents with a *common* reward function that aligns all agents' interests. In the extreme case with large populations of agents, such a homogeneity also indicates that the agents play an *interchangeable* role in the system evolution, and can hardly be distinguished from each other. We elaborate more on homogeneity below.

Multi-Agent MDP & Markov Teams

Consider a Markov game as in Definition 2.2 with $R^1 = R^2 = \cdots = R^N = R$, where the reward $R : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \to \mathbb{R}$ is influenced by the joint action in $\mathcal{A} = \mathcal{A}^1 \times \cdots \times \mathcal{A}^N$. As a result, the Q-function is identical for all agents. Hence, a straightforward algorithm proceeds by performing the standard Q-learning update (2.1) at each agent, but taking the max over the joint action space $a' \in \mathcal{A}$. Convergence to the optimal/equilibrium Q-function has been established in Szepesvári and Littman (1999); Littman (2001), when both state and action spaces are finite.

However, convergence of the Q-function does not necessarily imply that of the equilibrium policy for the Markov team, as any combination of equilibrium policies extracted at each agent may not constitute an equilibrium policy, if the equilibrium policies are non-unique, and the agents fail to agree on which one to select. Hence, convergence to the NE policy is only guaranteed if either the equilibrium is assumed to be unique (Littman, 2001), or the agents are coordinated for equilibrium selection. The latter idea has first been validated in the cooperative repeated games setting (Claus and Boutilier, 1998), a special case of Markov teams with a singleton state, where the agents are joint-action learners (JAL), maintaining a Q-value for joint actions, and learning empirical models of all others. Convergence to equilibrium point is claimed in Claus and Boutilier (1998), without a formal proof. For the actual Markov teams, this coordination has been exploited in Wang and Sandholm (2003), which proposes *optimal adaptive learning* (OAL), the first MARL algorithm with provable convergence to the equilibrium policy. Specifically, OAL first learns the game structure, and constructs virtual games at each state that are *weakly acyclic* with respect to (w.r.t.) a biased set. OAL can be shown to converge to the NE, by introducing the biased adaptive play learning algorithm for the constructed weakly acyclic games, motivated from Young (1993).

Apart from equilibrium selection, another subtlety special to Markov teams (compared to single-agent RL) is the necessity to address the scalability issue, see §3.3. As independent Q-learning may fail to converge (Tan, 1993), one early attempt toward developing scalable while convergent algorithms for MMDPs is Lauer and Riedmiller (2000), which advocates a distributed Q-learning algorithm that converges for deterministic finite MMDPs. Each agent maintains only a Q-table of state s and local action a^i, and successively takes maximization over the joint action a'. No other agent's actions and their histories can be acquired by each individual agent. Several other heuristics (with no theoretical backing) regarding either reward or value function factorization have been proposed to mitigate the scalability issue (Guestrin et al., 2002a,b; Kok and Vlassis, 2004; Sunehag et al., 2018; Rashid et al., 2018). Very recently, Son et al. (2019) provides a rigorous characterization of conditions that justify this value factorization idea. Another recent theoretical work along this direction is Qu and Li (2019), which imposes a spe-

cial dependence structure, i.e., a one-directional tree, so that the (near-)optimal policy of the overall MMDP can be provably well-approximated by *local policies*. Yongacoglu et al. (2019) has studied common interest games, which includes Markov teams as an example, and develops a *decentralized* RL algorithm that relies on only states, local actions and rewards. With the same information structure as independent Q-learning (Tan, 1993), the algorithm is guaranteed to converge to *team optimal* equilibrium policies, and not just equilibrium policies. This is important as in general, a suboptimal equilibrium can perform arbitrarily worse than the optimal equilibrium (Yongacoglu et al., 2019).

For policy-based methods, to our knowledge, the only convergence guarantee for this setting exists in Perolat et al. (2018). The authors propose two-timescale actor-critic *fictitious play* algorithms, where at the slower timescale, the actor mixes the current policy and the best-response one w.r.t. the local Q-value estimate, while at the faster timescale the critic performs policy evaluation, as if all agents' policies are stationary. Convergence is established for *simultaneous move multistage games* with a common (also zero-sum, see §4.2.2) reward, a special Markov team with initial and absorbing states, and each state being visited only once.

Markov Potential Games

From a game-theoretic perspective, a more general framework to embrace cooperation is *potential games* (Monderer and Shapley, 1996), where there exits some *potential* function shared by all agents, such that if any agent changes its policy unilaterally, the change in its reward equals (or proportions to) that in the potential function. Though most potential games are stateless, an extension named *Markov potential games* (MPGs) has gained increasing attention for modeling *sequential* decision-making (González-Sánchez and Hernández-Lerma, 2013; Zazo et al., 2016), which includes Markovian states whose evolution is affected by the joint actions. Indeed, MMDPs/Markov teams constitute a particular case of MPGs, with the potential function being the common reward; such dynamic games can also be viewed as being *strategically equivalent* to Markov teams, using the terminology in, e.g., (Başar and Zaccour, 2018, Chapter 1). Under this model, Valcarcel Macua et al. (2018) provides verifiable conditions for a Markov game to be an MPG, and shows the equivalence between finding closed-loop NE in MPG and solving a single-agent optimal control problem. Hence, single-agent RL algorithms are then enabled to solve this MARL problem.

Mean-Field Regime

Another idea toward tackling the scalability issue is to take the setting to the *mean-field* regime, with an extremely large number of homogeneous agents. Each agent's effect on the overall multi-agent system can thus become infinitesimal, resulting in all agents being interchangeable/indistinguishable. The interaction with other agents, however, is captured simply by some mean-field quantity, e.g., the average state, or the empirical distribution of states. Each agent only needs to find the best response to the mean-field, which considerably simplifies the analysis. This mean-field view of multi-agent systems has been approached by the mean-field games (MFGs) model (Huang et al., 2003, 2006; Lasry and Lions, 2007; Bensoussan et al., 2013; Tembine et al., 2013), the team model with mean-field sharing (Arabneydi and Mahajan, 2014; Arabneydi, 2017), and the game

model with mean-field actions (Yang et al., 2018).[4]

MARL in these models have not been explored until recently, mostly in the non-cooperative setting of MFGs, see §4.3 for a more detailed review. Regarding the cooperative setting, recent work Subramanian et al. (2018) studies RL for Markov teams with mean-field sharing (Arabneydi and Mahajan, 2014, 2016; Arabneydi, 2017). Compared to MFG, the model considers agents that share a common reward function depending only on the local state and the mean-field, which encourages cooperation among the agents. Also, the term mean-field refers to the *empirical average* for the states of *finite* population, in contrast to the *expectation* and *probability distribution* of *infinite* population in MFGs. Based on the dynamic programming decomposition for the specified model (Arabneydi and Mahajan, 2014), several popular RL algorithms are easily translated to address this setting (Subramanian et al., 2018). More recently, Carmona et al. (2019a,b) approach the problem from a mean-field control (MFC) model, to model large-population of cooperative decision-makers. Policy gradient methods are proved to converge for linear quadratic MFCs in Carmona et al. (2019a), and mean-field Q-learning is then shown to converge for general MFCs (Carmona et al., 2019b).

4.1.2 Decentralized Paradigm with Networked Agents

Cooperative agents in numerous practical multi-agent systems are not always homogeneous. Agents may have different preferences, i.e., reward functions, while still form a team to maximize the return of the *team-average* reward $\bar{R}$, where $\bar{R}(s,a,s') = N^{-1} \cdot \sum_{i\in\mathcal{N}} R^i(s,a,s')$. More subtly, the reward function is sometimes not sharable with others, as the preference is kept private to each agent. This setting finds broad applications in engineering systems as sensor networks (Rabbat and Nowak, 2004), smart grid (Dall'Anese et al., 2013; Zhang et al., 2018a), intelligent transportation systems (Adler and Blue, 2002; Zhang et al., 2018b), and robotics (Corke et al., 2005).

Covering the homogeneous setting in §4.1.1 as a special case, the specified one definitely requires more coordination, as, for example, the global value function cannot be estimated locally without knowing other agents' reward functions. With a central controller, most MARL algorithms reviewed in §4.1.1 directly apply, since the controller can collect and average the rewards, and distributes the information to all agents. Nonetheless, such a controller may not exist in most aforementioned applications, due to either cost, scalability, or robustness concerns (Rabbat and Nowak, 2004; Dall'Anese et al., 2013; Zhang et al., 2019). Instead, the agents may be able to share/exchange information with their neighbors over a possibly time-varying and sparse communication network, as illustrated in Figure 2 (b). Though MARL under this *decentralized/distributed*[5] paradigm is imperative, it is relatively less-investigated, in comparison to the extensive results on distributed/consensus algorithms that solve *static/one-stage* optimization problems (Nedic

[4]The difference between mean-field teams and mean-field games is mainly the solution concept: optimum versus equilibrium, as the difference between general dynamic team theory (Witsenhausen, 1971; Yoshikawa, 1978; Yüksel and Başar, 2013) and game theory (Shapley, 1953; Filar and Vrieze, 2012). Although the former can be viewed as a special case of the latter, related works are usually reviewed separately in the literature. We follow here this convention.

[5]Note that hereafter we use *decentralized* and *distributed* interchangeably for describing this paradigm.

and Ozdaglar, 2009; Agarwal and Duchi, 2011; Jakovetic et al., 2011; Tu and Sayed, 2012), which, unlike RL, involves no system *dynamics*, and does not maximize the *long-term* objective in a *sequential* fashion.

Learning Optimal Policy

The most significant goal is to learn the optimal joint policy, while each agent only accesses to local and neighboring information over the network. The idea of MARL with networked agents dates back to Varshavskaya et al. (2009). To our knowledge, the first provably convergent MARL algorithm under this setting is due to Kar et al. (2013), which incorporates the idea of *consensus + innovation* to the standard Q-learning algorithm, yielding the *QD-learning* algorithm with the following update

$$Q^i_{t+1}(s,a) \leftarrow Q^i_t(s,a) + \alpha_{t,s,a}\Big[R^i(s,a) + \gamma \min_{a'\in\mathcal{A}} Q^i_t(s',a') - Q^i_t(s,a)\Big] - \beta_{t,s,a}\sum_{j\in\mathcal{N}^i_t}\Big[Q^i_t(s,a) - Q^j_t(s,a)\Big],$$

where $\alpha_{t,s,a},\ \beta_{t,s,a} > 0$ are stepsizes, $\mathcal{N}^i_t$ denotes the set of neighboring agents of agent i, at time t. Compared to the Q-learning update (2.1), *QD*-learning appends an innovation term that captures the difference of Q-value estimates from its neighbors. With certain conditions on the stepsizes, the algorithm is guaranteed to converge to the optimum Q-function for the tabular setting.

Due to the scalability issue, function approximation is vital in MARL, which necessitates the development of policy-based algorithms. Our work Zhang et al. (2018) proposes actor-critic algorithms for this setting. Particularly, each agent parameterizes its own policy $\pi^i_{\theta^i}: \mathcal{S} \to \mathcal{A}^i$ by some parameter $\theta^i \in \mathbb{R}^{m^i}$, the policy gradient of the return is first derived as

$$\nabla_{\theta^i} J(\theta) = \mathbb{E}\Big[\nabla_{\theta^i} \log \pi^i_{\theta^i}(s,a^i)\cdot Q_\theta(s,a)\Big] \tag{4.1}$$

where Q_θ is the global value function corresponding to $\bar{R}$ under the joint policy π_θ that is defined as $\pi_\theta(a|s) := \prod_{i\in\mathcal{N}} \pi^i_{\theta^i}(a^i|s)$. As an analogy to (2.2), the policy gradient in (4.1) involves the expectation of the product between the local score function $\nabla_{\theta^i}\log\pi^i_{\theta^i}(s,a^i)$ and the global Q-function Q_θ. The latter, nonetheless, cannot be estimated individually at each agent. As a result, by parameterizing each local copy of $Q_\theta(\cdot,\cdot)$ as $Q_\theta(\cdot,\cdot;\omega^i)$ for agent i, we propose the following consensus-based TD learning for the critic step, i.e., for estimating $Q_\theta(\cdot,\cdot)$:

$$\widetilde{\omega}^i_t = \omega^i_t + \beta_{\omega,t}\cdot\delta^i_t\cdot\nabla_\omega Q_t(\omega^i_t), \qquad\qquad \omega^i_{t+1} = \sum_{j\in\mathcal{N}} c_t(i,j)\cdot\widetilde{\omega}^j_t, \tag{4.2}$$

where $\beta_{\omega,t} > 0$ is the stepsize, δ^i_t is the local TD-error calculated using $Q_\theta(\cdot,\cdot;\omega^i)$. The first relation in (4.2) performs the standard TD update, followed by a weighted combination of the neighbors' estimates $\widetilde{\omega}^j_t$. The weights here, $c_t(i,j)$, are dictated by the topology of the communication network, with non-zero values only if two agents i and j are connected at time t. They also need to satisfy the *doubly stochastic* property in expectation, so that ω^i_t reaches a *consensual* value for all $i \in \mathcal{N}$ if it converges. Then, each agent i updates its

policy following stochastic policy gradient given by (4.1) in the actor step, using its own Q-function estimate $Q_\theta(\cdot,\cdot;\omega_t^i)$. A variant algorithm is also introduced in Zhang et al. (2018), relying on not the Q-function, but the state-value function approximation, to estimate the global advantage function.

With these in mind, almost sure convergence is established in Zhang et al. (2018) for these decentralized actor-critic algorithms, when linear functions are used for value function approximation. Similar ideas are also extended to the setting with continuous spaces (Zhang et al., 2018a), where deterministic policy gradient (DPG) method is usually used. Off-policy exploration, namely a stochastic behavior policy, is required for DPG, as the deterministic on-policy may not be explorative enough. However, in the multi-agent setting, as the policies of other agents are unknown, the common off-policy approach for DPG (Silver et al., 2014, §4.2) does not apply. Inspired by the expected policy gradient (EPG) method (Ciosek and Whiteson, 2018) which unifies stochastic PG (SPG) and DPG, we develop an algorithm that remains on-policy, but reduces the variance of general SPG (Zhang et al., 2018a). In particular, we derive the multi-agent version of EPG, based on which we develop the actor step that can be implemented in a decentralized fashion, while the critic step still follows (4.2). Convergence of the algorithm is then also guaranteed when linear function approximation is used (Zhang et al., 2018a). In the same vein, Suttle et al. (2019) considers the extension of Zhang et al. (2018) to an off-policy setting, building upon the emphatic temporal differences (ETD) method for the critic (Sutton et al., 2016). By incorporating the analysis of ETD(λ) (Yu, 2015) into Zhang et al. (2018), almost sure convergence guarantee has also been established. Another off-policy algorithm for the same setting is proposed concurrently by Zhang and Zavlanos (2019), where agents do not share their estimates of value function. Instead, the agents aim to reach consensus over the global optimal policy estimation. Provable convergence is then established for the algorithm, with a local critic and a consensus actor.

RL for decentralized networked agents has also been investigated in *multi-task*, in addition to the multi-agent, settings. In some sense, the former can be regarded as a simplified version of the latter, where each agent deals with an *independent MDP* that is not affected by other agents, while the goal is still to learn the optimal joint policy that accounts for the average reward of all agents. Pennesi and Paschalidis (2010) proposes a distributed actor-critic algorithm, assuming that the states, actions, and rewards are all local to each agent. Each agent performs a local TD-based critic step, followed by a consensus-based actor step that follows the gradient calculated using information exchanged from the neighbors. Gradient of the average return is then proved to converge to zero as the iteration goes to infinity. Macua et al. (2017) has developed *Diff-DAC*, another distributed actor-critic algorithm for this setting, from duality theory. The updates resemble those in Zhang et al. (2018), but provide additional insights that actor-critic is actually an instance of the dual ascent method for solving a linear program.

Note that all the aforementioned convergence guarantees are *asymptotic*, i.e., the algorithms converge as the iteration numbers go to infinity, and are restricted to the case with linear function approximations. This fails to quantify the performance when finite iterations and/or samples are used, not to mention when nonlinear functions such as deep neural networks are utilized. As an initial step toward *finite-sample analyses* in this setting with more *general* function approximation, we consider in Zhang et al. (2018b) the *batch*

RL algorithms (Lange et al., 2012), specifically, decentralized variants of the fitted-Q iteration (FQI) (Riedmiller, 2005; Antos et al., 2008). Note that we focus on FQI since it motivates the celebrated deep Q-learning algorithm (Mnih et al., 2015) when deep neural networks are used. We study FQI variants for both the cooperative setting with networked agents, and the competitive setting with two teams of such networked agents (see §4.2.1 for more details). In the former setting, all agents cooperate to iteratively update the global Q-function estimate, by fitting nonlinear least squares with the target values as the responses. In particular, let $\mathcal{F}$ be the function class for Q-function approximation, $\{(s_j, \{a_j^i\}_{i\in\mathcal{N}}, s_j')\}_{j\in[n]}$ be the batch transitions dataset of size n available to all agents, $\{r_j^i\}_{j\in[n]}$ be the local reward samples private to each agent, and $y_j^i = r_j^i + \gamma \cdot \max_{a\in\mathcal{A}} Q_t^i(s_j', a)$ be the local target value, where Q_t^i is agent i's Q-function estimate at iteration t. Then, all agents cooperate to find a common Q-function estimate by solving

$$\min_{f\in\mathcal{F}} \ \frac{1}{N}\sum_{i\in\mathcal{N}} \frac{1}{2n}\sum_{j=1}^{n}\Big[y_j^i - f(s_j, a_j^1, \cdots, a_j^N)\Big]^2. \tag{4.3}$$

Since y_j^i is only known to agent i, the problem in (4.3) aligns with the formulation of *distributed/consensus optimization* in Nedic and Ozdaglar (2009); Agarwal and Duchi (2011); Jakovetic et al. (2011); Tu and Sayed (2012); Hong and Chang (2017); Nedic et al. (2017), whose global optimal solution can be achieved by the algorithms therein, if $\mathcal{F}$ makes $\sum_{j=1}^{n}[y_j^i - f(s_j, a_j^1, \cdots, a_j^N)]^2$ convex for each i. This is indeed the case if $\mathcal{F}$ is a linear function class. Nevertheless, with only a finite iteration of distributed optimization algorithms (common in practice), agents may not reach exact consensus, leading to an error of each agent's Q-function estimate away from the actual optimum of (4.3). Such an error also exists when nonlinear function approximation is used. Considering this error caused by decentralized computation, we follow the *error propagation* analysis stemming from single-agent batch RL (Munos, 2007; Antos et al., 2008; Munos and Szepesvári, 2008; Antos et al., 2008; Farahmand et al., 2010), to establish the finite-sample performance of the proposed algorithms, i.e., how accuracy of the algorithms output depends on the function class $\mathcal{F}$, the number of samples within each iteration n, and the number of iterations for t.

Policy Evaluation

Aside from control, a series of algorithms in this setting focuses on the policy evaluation task only, namely, the critic step of the actor-critic algorithms. With the policy fixed, this task embraces a neater formulation, as the sampling distribution becomes stationary, and the objective becomes convex under linear function approximation. This facilitates the finite-time/sample analyses, in contrast to most control algorithms with only asymptotic guarantees. Specifically, under joint policy π, suppose each agent parameterizes the value function by a linear function class $\{V_\omega(s) := \phi^\top(s)\omega : \omega \in \mathbb{R}^d\}$, where $\phi(s) \in \mathbb{R}^d$ is the feature vector at $s \in \mathcal{S}$, and $\omega \in \mathbb{R}^d$ is the vector of parameters. For notational convenience, let $\Phi := (\cdots; \phi^\top(s); \cdots) \in \mathbb{R}^{|\mathcal{S}|\times d}$, $D = \text{diag}[\{\eta_\pi(s)\}_{s\in\mathcal{S}}] \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ be a diagonal matrix constructed using the state-occupancy measure η_π, $\bar{R}^\pi(s) = N^{-1}\cdot\sum_{i\in\mathcal{N}} R^{i,\pi}(s)$, where $R^{i,\pi}(s) = \mathbb{E}_{a\sim\pi(\cdot|s), s'\sim P(\cdot|s,a)}[R^i(s,a,s')]$, and $P^\pi \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ with the (s,s') element being

$[P^\pi]_{s,s'} = \sum_{a\in\mathcal{A}} \pi(a|s)P(s'|s,a)$. The objective of all agents is to jointly minimize the mean square projected Bellman error (MSPBE) associated with the team-average reward, i.e.,

$$\min_{\omega} \quad \texttt{MSPBE}(\omega) := \left\|\Pi_\Phi\Big(V_\omega - \gamma P^\pi V_\omega - \bar{R}^\pi\Big)\right\|_D^2 = \left\|A\omega - b\right\|_{C^{-1}}^2, \tag{4.4}$$

where $\Pi_\Phi : \mathbb{R}^{|\mathcal{S}|} \to \mathbb{R}^{|\mathcal{S}|}$ defined as $\Pi_\Phi := \Phi(\Phi^\top D\Phi)^{-1}\Phi^\top D$ is the projection operator onto subspace $\{\Phi\omega : \omega \in \mathbb{R}^d\}$, $A := \mathbb{E}\{\phi(s)[\phi(s) - \gamma\phi(s')]^\top\}$, $C := \mathbb{E}[\phi(s)\phi^\top(s)]$, and $b := \mathbb{E}[\bar{R}^\pi(s)\phi(s)]$. By replacing the expectation with samples and using the Fenchel duality, the finite-sum version of (4.4) can be re-formulated as a distributed saddle-point problem

$$\min_{\omega}\max_{\lambda^i, i\in\mathcal{N}} \quad \frac{1}{Nn}\sum_{i\in\mathcal{N}}\sum_{j=1}^{n} 2(\lambda^i)^\top A_j\omega - 2(b_j^i)^\top\lambda^i - (\lambda^i)^\top C_j\lambda^i \tag{4.5}$$

where n is the data size, A_j, C_j and b_j^i are empirical estimates of A, C and $b^i := \mathbb{E}[R^{i,\pi}(s)\phi(s)]$, respectively, using sample j. Note that (4.5) is convex in ω and concave in $\{\lambda^i\}_{i\in\mathcal{N}}$. The use of MSPBE as an objective is standard in multi-agent policy evaluation (Macua et al., 2015; Lee et al., 2018; Wai et al., 2018; Doan et al., 2019), and the idea of saddle-point reformulation has been adopted in Macua et al. (2015); Lee et al. (2018); Wai et al. (2018); Cassano et al. (2018). Note that in Cassano et al. (2018), a variant of MSPBE, named H-truncated λ-weighted MSPBE, is advocated, in order to control the bias of the solution deviated from the actual mean square Bellman error minimizer.

With the formulation (4.4) in mind, Lee et al. (2018) develops a distributed variant of the gradient TD-based method, with asymptotic convergence established using the ordinary differential equation (ODE) method. In Wai et al. (2018), a double averaging scheme that combines the dynamic consensus (Qu and Li, 2017) and the SAG algorithm (Schmidt et al., 2017) has been proposed to solve the saddle-point problem (4.5) with linear rate. Cassano et al. (2018) incorporates the idea of variance-reduction, specifically, AVRG in (Ying et al., 2018), into gradient TD-based policy evaluation. Achieving the same linear rate as Wai et al. (2018), three advantages are claimed in Cassano et al. (2018): i) data-independent memory requirement; ii) use of eligibility traces (Singh and Sutton, 1996); iii) no need for synchronization in sampling. More recently, standard TD learning (Tesauro, 1995), instead of gradient-TD, has been generalized to this MARL setting, with special focuses on finite-sample analyses, see Doan et al. (2019) and Doan et al. (2019). Distributed TD(0) is first studied in Doan et al. (2019), using the proof techniques originated in Bhandari et al. (2018), which requires a projection on the iterates, and the data samples to be independent and identically distributed (i.i.d.). Furthermore, motivated by the recent progress in Srikant and Ying (2019), finite-time performance of the more general distributed TD(λ) algorithm is provided in Doan et al. (2019), with neither projection nor i.i.d. noise assumption needed.

Policy evaluation for networked agents has also been investigated under the setting of *independent* agents interacting with *independent* MDPs. Macua et al. (2015) studies off-policy evaluation based on the importance sampling technique. With no coupling among MDPs, an agent does not need to know the actions of the other agents. Diffusion-based distributed GTD is then proposed, and is shown to convergence in the mean-square sense

with a sublinear rate. In Stanković and Stanković (2016), two variants of the TD-learning, namely, GTD2 and TDC (Sutton et al., 2009), have been designed for this setting, with weak convergence proved by the general stochastic approximation theory in Stanković et al. (2016), when agents are connected by a time-varying communication network. Note that Cassano et al. (2018) also considers the independent MDP setting, with the same results established as the actual MARL setting.

Other Learning Goals

Several other learning goals have also been explored for decentralized MARL with networked agents. Zhang et al. (2016) has considered the *optimal consensus* problem, where each agent over the network tracks the states of its neighbors' as well as a leader's, so that the consensus error is minimized by the joint policy. A policy iteration algorithm is then introduced, followed by a practical actor-critic algorithm using neural networks for function approximation. A similar consensus error objective is also adopted in Zhang et al. (2018), under the name of cooperative multi-agent graphical games. A centralized-critic-decentralized-actor scheme is utilized for developing off-policy RL algorithms.

Communication efficiency, as a key ingredient in the algorithm design for this setting, has drawn increasing attention recently (Chen et al., 2018; Ren and Haupt, 2019; Lin et al., 2019). Specifically, Chen et al. (2018) develops Lazily Aggregated Policy Gradient (LAPG), a distributed PG algorithm that can reduce the communication rounds between the agents and a central controller, by judiciously designing communication trigger rules. Ren and Haupt (2019) addresses the same policy evaluation problem as Wai et al. (2018), and develops a hierarchical distributed algorithm by proposing a mixing matrix different from the doubly stochastic one used in Zhang et al. (2018); Wai et al. (2018); Lee et al. (2018), which allows unidirectional information exchange among agents to save communication. In contrast, the distributed actor-critic algorithm in Lin et al. (2019) reduces the communication by transmitting only one scaled entry of its state vector at each iteration, while preserving provable convergence as in Zhang et al. (2018).

4.1.3 Partially Observed Model

We complete the overview for cooperative settings by briefly introducing a class of significant but challenging models where agents are faced with partial observability. Though common in practice, theoretical analysis of algorithms in this setting is still in its infancy, in contrast to the aforementioned fully observed settings. In general, this setting can be modeled by a decentralized POMDP (Dec-POMDP) (Oliehoek and Amato, 2016), which shares almost all elements such as the reward function and the transition model, as the MMDP/Markov team model in §2.2.1, except that each agent now only has its local observations of the system state s. With no accessibility to other agents' observations, an individual agent cannot maintain a global belief state, the sufficient statistic for decision making in single-agent POMDPs. Hence, Dec-POMDPs have been known to be NEXP-hard (Bernstein et al., 2002), requiring super-exponential time to solve in the worst case.

There is an increasing interest in developing planning/learning algorithms for Dec-POMDPs. Most of the algorithms are based on a *centralized-learning-decentralized-execution* scheme. In particular, the decentralized problem is first reformulated as a centralized

one, which can be solved at a central controller with (a simulator that generates) the observation data of all agents. The policies are then optimized/learned using data, and distributed to all agents for execution. Finite-state controllers (FSCs) are commonly used to represent the local policy at each agent (Bernstein et al., 2009; Amato et al., 2010), which map local observation histories to actions. A Bayesian nonparametric approach is proposed in Liu et al. (2015) to determine the controller size of variable-size FSCs. To efficiently solve the centralized problem, a series of *top-down* algorithms have been proposed. In Oliehoek and Amato (2014), the Dec-POMDP is converted to *non-observable MDP* (NOMDP), a kind of centralized sequential decision-making problem, which is then addressed by some heuristic tree search algorithms. As an extension of the NOMDP conversion, Dibangoye et al. (2016); Dibangoye and Buffet (2018) convert Dec-POMDPs to *occupancy-state* MDPs (oMDPs), where the occupancy-states are distributions over hidden states and joint histories of observations. As the value functions of oMDPs enjoy the piece-wise linearity and convexity, both tractable planning (Dibangoye et al., 2016) and value-based learning (Dibangoye and Buffet, 2018) algorithms have been developed.

To further improve computational efficiency, several sampling-based planning/learning algorithms have also been proposed. In particular, Monte-Carlo sampling with policy iteration and the expectation-maximization algorithm, are proposed in Wu et al. (2010) and Wu et al. (2013), respectively. Furthermore, Monte-Carlo tree search has been applied to special classes of Dec-POMDPs, such as multi-agent POMDPs (Amato and Oliehoek, 2015) and multi-robot active perception (Best et al., 2018). In addition, policy gradient-based algorithms can also be developed for this centralized learning scheme (Dibangoye and Buffet, 2018), with a centralized critic and a decentralized actor. Finite-sample analysis can also be established under this scheme (Amato and Zilberstein, 2009; Banerjee et al., 2012), for tabular settings with finite state-action spaces.

Several attempts have also been made to enable *decentralized learning* in Dec-POMDPs. When the agents share some common information/observations, Nayyar et al. (2013) proposes to reformulate the problem as a centralized POMDP, with the common information being the observations of a *virtual* central controller. This way, the centralized POMDP can be solved individually by each agent. In Arabneydi and Mahajan (2015), the reformulated POMDP has been approximated by finite-state MDPs with exponentially decreasing approximation error, which are then solved by Q-learning. Very recently, Zhang et al. (2019) has developed a tree-search based algorithm for solving this centralized POMDP, which, interestingly, echoes back the heuristics for solving Dec-POMDPs directly as in Amato and Oliehoek (2015); Best et al. (2018), but with a more solid theoretical footing. Note that in both Arabneydi and Mahajan (2015) and Zhang et al. (2019), a common random number generator is used for all agents, in order to avoid communication among agents and enable a decentralized learning scheme.

4.2 Competitive Setting

Competitive settings are usually modeled as *zero-sum* games. Computationally, there exists a great barrier between solving two-player and multi-player zero-sum games. In particular, even the simplest three-player matrix games, are known to be PPAD-complete (Papadimitriou, 1992; Daskalakis et al., 2009). Thus, most existing results on competitive

MARL focus on two-player zero-sum games, with $\mathcal{N} = \{1, 2\}$ and $R^1 + R^2 = 0$ in Definitions 2.2 and 2.4. In the rest of this section, we review methods that provably find a Nash (equivalently, saddle-point) equilibrium in two-player Markov or extensive-form games. The existing algorithms can mainly be categorized into two classes: value-based and policy-based approaches, which are introduced separately in the sequel.

4.2.1 Value-Based Methods

Similar to the MDPs, value-based methods aim to find an optimal value function from which the joint Nash equilibrium policy can be extracted. Moreover, the optimal value function is known to be the unique fixed point of a Bellman operator, which can be obtained via dynamic programming type methods.

Specifically, for simultaneous-move Markov games, the value function defined in (2.3) satisfies $V^1_{\pi^1,\pi^2} = -V^2_{\pi^1,\pi^2}$. and thus any Nash equilibrium $\pi^* = (\pi^{1,*}, \pi^{2,*})$ satisfies

$$V^1_{\pi^1,\pi^{1,*}}(s) \le V^1_{\pi^{1,*},\pi^{2,*}}(s) \le V^1_{\pi^{1,*},\pi^2}(s), \qquad \text{for any } \pi = (\pi^1, \pi^2) \text{ and } s \in \mathcal{S}. \tag{4.6}$$

By the Von Neumann minimax theorem (Von Neumann et al., 2007), we define the optimal value function $V^* \colon \mathcal{S} \to \mathbb{R}$ as

$$V^* = \max_{\pi^1} \min_{\pi^2} V^1_{\pi^1,\pi^2} = \min_{\pi^2} \max_{\pi^1} V^1_{\pi^1,\pi^2}, \tag{4.7}$$

Then (4.6) implies that $V^1_{\pi^{1,*},\pi^{2,*}}$ coincides with V^* and any pair of policies π^1 and π^2 that attains the supremum and infimum in (4.7) constitutes a Nash equilibrium. Moreover, similar to MDPs, Shapley (1953) shows that V^* is the unique solution of a Bellman equation and a Nash equilibrium can be constructed based on V^*. Specifically, for any $V \colon \mathcal{S} \to \mathbb{R}$ and any $s \in \mathcal{S}$, we define

$$Q_V(s, a^1, a^2) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a^1,a^2)}\Big[R^1(s, a^1, a^2, s') + \gamma \cdot V(s')\Big], \tag{4.8}$$

which is regarded as a matrix in $\mathbb{R}^{|\mathcal{A}^1| \times |\mathcal{A}^2|}$. Then we define the Bellman operator $\mathcal{T}^*$ by solving a matrix zero-sum game regarding $Q_V(s, \cdot, \cdot)$ as the payoff matrix, i.e., for any $s \in \mathcal{S}$, we define

$$(\mathcal{T}^* V)(s) = \texttt{Value}\Big[Q_V(s, \cdot, \cdot)\Big] = \max_{u \in \Delta(\mathcal{A}^1)} \min_{v \in \Delta(\mathcal{A}^2)} \sum_{a \in \mathcal{A}^1} \sum_{b \in \mathcal{A}^2} u_a \cdot v_b \cdot Q_V(s, a, b), \tag{4.9}$$

where we use $\texttt{Value}(\cdot)$ to denote the optimal value of a matrix zero-sum game, which can be obtained by solving a linear program (Vanderbei et al., 2015). Thus, the Bellman operator $\mathcal{T}^*$ is γ-contractive in the ℓ_∞-norm and V^* in (4.7) is the unique solution to the Bellman equation $V = \mathcal{T}^* V$. Moreover, letting $p_1(V), p_2(V)$ be any solution to the optimization problem in (4.9), we have that $\pi^* = (p_1(V^*), p_2(V^*))$ is a Nash equilibrium specified by Definition 2.3. Thus, based on the Bellman operator $\mathcal{T}^*$, Shapley (1953) proposes the value iteration algorithm, which creates a sequence of value functions $\{V_t\}_{t \ge 1}$ satisfying $V_{t+1} = \mathcal{T}^* V_t$, which converges to V^* with a linear rate. Specifically, we have

$$\|V_{t+1} - V^*\|_\infty = \|\mathcal{T}^* V_t - \mathcal{T}^* V^*\|_\infty \le \gamma \cdot \|V_t - V^*\|_\infty \le \gamma^{t+1} \cdot \|V_0 - V^*\|_\infty.$$

In addition, a value iteration update can be decomposed into the two steps. In particular, letting π^1 be any policy of player 1 and V be any value function, we define Bellman operator $\mathcal{T}^{\pi^1}$ by

$$(\mathcal{T}^{\pi^1} V)(s) = \min_{v \in \Delta(\mathcal{A}^2)} \sum_{a \in \mathcal{A}^1} \sum_{b \in \mathcal{A}^2} \pi^1(a|s) \cdot v_b \cdot Q_V(s,a,b), \tag{4.10}$$

where Q_V is defined in (4.8). Then we can equivalently write a value iteration update as

$$\mu_{t+1} = p_1(V_t) \qquad \text{and} \qquad V_{t+1} = \mathcal{T}^{\mu_{t+1}} V_t. \tag{4.11}$$

Such a decomposition motivates the policy iteration algorithm for two-player zero-sum games, which has been studied in, e.g., Hoffman and Karp (1966); Van Der Wal (1978); Rao et al. (1973); Patek (1997); Hansen et al. (2013). In particular, from the perspective of player 1, policy iteration creates a sequence $\{\mu_t, V_t\}_{t \geq 0}$ satisfying

$$\mu_{t+1} = p_1(V_t) \qquad \text{and} \qquad V_{t+1} = (\mathcal{T}^{\mu_{t+1}})^{\infty} V_t, \tag{4.12}$$

i.e., V_{t+1} is the fixed point of $\mathcal{T}^{\mu_{t+1}}$. The updates for player 2 can be similarly constructed. By the definition in (4.10), the Bellman operator $\mathcal{T}^{\mu_{t+1}}$ is γ-contractive and its fixed point corresponds to the value function associated with $(\mu_{t+1}, \mathrm{Br}(\mu_{t+1}))$, where $\mathrm{Br}(\mu_{t+1})$ is the best response policy of player 2 when the first player adopts μ_{t+1}. Hence, in each step of policy iteration, the player first finds an improved policy μ_{t+1} based on the current function V_t, and then obtains a conservative value function by assuming that the opponent plays the best counter policy $\mathrm{Br}(\mu_{t+1})$. It has been shown in Hansen et al. (2013) that the value function sequence $\{V_t\}_{t \geq 0}$ monotonically increases to V^* with a linear rate of convergence.

Notice that both the value and policy iteration algorithms are model-based due to the need of computing the Bellman operator $\mathcal{T}^{\mu_{t+1}}$ in (4.11) and (4.12). By estimating the Bellman operator via data-driven approximation, Littman (1994) has proposed minimax-Q learning, which extends the well-known Q-learning algorithm (Watkins and Dayan, 1992) for MDPs to zero-sum Markov games. In particular, minimax-Q learning is an online, off-policy, and tabular method which updates the action-value function $Q\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ based on transition data $\{(s_t, a_t, r_t, s_t')\}_{t \geq 0}$, where s_t' is the next state following (s_t, a_t) and r_t is the reward. In the t-th iteration, it only updates the value of $Q(s_t, a_t)$ and keeps other entries of Q unchanged. Specifically, we have

$$Q(s_t, a_t^1, a_t^2) \leftarrow (1 - \alpha_t) \cdot Q(s_t, a_t^1, a_t^2) + \alpha_t \cdot \Big\{ r_t + \gamma \cdot \mathtt{Value}\Big[Q(s_t', \cdot, \cdot)\Big] \Big\}, \tag{4.13}$$

where $\alpha_t \in (0,1)$ is the stepsize. As shown in Szepesvári and Littman (1999), under conditions similar to those for single-agent Q-learning (Watkins and Dayan, 1992), function Q generated by (4.13) converges to the optimal action-value function $Q^* = Q_{V^*}$ defined in (4.8). Moreover, with a slight abuse of notation, if we define the Bellman operator $\mathcal{T}^*$ for action-value functions by

$$(\mathcal{T}^* Q)(s, a_1, a_2) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a^1,a^2)} \Big\{ R^1(s, a^1, a^2, s') + \gamma \cdot \mathrm{Value}\Big[Q(s', \cdot, \cdot)\Big] \Big\}, \tag{4.14}$$

then we have Q^* as the unique fixed point of $\mathcal{T}^*$. Since the target value $r_t+\gamma\cdot\texttt{Value}[Q(s'_t,\cdot,\cdot)]$ in (4.13) is an unbiased estimator of $(\mathcal{T}^*Q)(s_t,a_t^1,a_t^2)$, minimax-Q learning can be viewed as a stochastic approximation algorithm for computing the fixed point of $\mathcal{T}^*$. Following Littman (1994), minimax-Q learning has been further extended to the function approximation setting where Q in (4.13) is approximated by a class of parametrized functions. In particular, Lagoudakis and Parr (2002); Zou et al. (2019) establish the convergence of minimax-Q learning with linear function approximation and temporal-difference updates (Sutton and Barto, 1987). Such a linear value function approximation also applies to a significant class of zero-sum MG instances with continuous state-action spaces, i.e., linear quadratic (LQ) zero-sum games (Başar and Bernhard, 1995; Al-Tamimi et al., 2007a,b), where the reward function is quadratic with respect to the states and actions, while the transition model follows linear dynamics. In this setting, Q-learning based algorithm can be guaranteed to converge to the NE (Al-Tamimi et al., 2007b).

To embrace general function classes, the framework of batch RL (Munos, 2007; Antos et al., 2008; Munos and Szepesvári, 2008; Antos et al., 2008; Farahmand et al., 2016) can be adapted to the multi-agent settings, as in the recent works Yang et al. (2019); Zhang et al. (2018b). As mentioned in §4.1.2 for cooperative batch MARL, each agent iteratively updates the Q-function by fitting least-squares using the target values. Specifically, let $\mathcal{F}$ be the function class of interest and let $\{(s_i,a_i^1,a_i^2,r_i,s'_i)\}_{i\in[n]}$ be the dataset. For any $t\geq 0$, let Q_t be the current iterate in the t-th iteration, and define $y_i=r_i+\gamma\cdot\texttt{Value}[Q_t(s'_i,\cdot,\cdot)]$ for all $i\in[n]$. Then we update Q_t by solving a least-squares regression problem in $\mathcal{F}$, that is,

$$Q_{t+1}\leftarrow\operatorname*{argmin}_{f\in\mathcal{F}}\frac{1}{2n}\sum_{i=1}^{n}\Big[y_i-f(s_i,a_i^1,a_i^2)\Big]^2. \tag{4.15}$$

In such a two-player zero-sum Markov game setting, a finite-sample error bound on the Q-function estimate is established in Yang et al. (2019).

Regarding other finite-sample analyses, very recently, Jia et al. (2019) has studied zero-sum turn-based stochastic games (TBSG), a simplified zero-sum MG when the transition model is embedded in some feature space and a generative model is available. Two Q-learning based algorithms have been proposed and analyzed for this setting. Sidford et al. (2019) has proposed algorithms that achieve near-optimal sample complexity for general zero-sum TBSGs with a generative model, by extending the previous near-optimal Q-learning algorithm for MDPs in Sidford et al. (2018). In the online setting, where the learner controls only one of the players that plays against an arbitrary opponent, Wei et al. (2017) has proposed UCSG, an algorithm for the *average-reward* zero-sum MGs, using the principle of *optimism in the face of uncertainty* (Auer and Ortner, 2007; Jaksch et al., 2010). UCSG is shown to achieve a sublinear regret compared to the game value when competing with an arbitrary opponent, and also achieve $\widetilde{O}(\text{poly}(1/\epsilon))$ sample complexity if the opponent plays an optimistic best response.

Furthermore, when it comes to zero-sum games with imperfect information, Koller et al. (1994); Von Stengel (1996); Koller et al. (1996); Von Stengel (2002) have proposed to transform extensive-form games into normal-form games using the *sequence form* representation, which enables equilibrium finding via linear programming. In addition, by lifting the state space to the space of belief states, Parr and Russell (1995); Rodriguez

et al. (2000); Hauskrecht (2000); Hansen et al. (2004); Buter (2012) have applied dynamic programming methods to zero-sum stochastic games. Both of these approaches guarantee finding of a Nash equilibrium but are only efficient for small-scale problems. Finally, MCTS with UCB-type action selection rule (Chang et al., 2005; Kocsis and Szepesvári, 2006; Coulom, 2006) can also be applied to two-player turn-based games with incomplete information (Kocsis and Szepesvári, 2006; Cowling et al., 2012; Teraoka et al., 2014; Whitehouse, 2014; Kaufmann and Koolen, 2017), which lays the foundation for the recent success of deep RL for the game of Go (Silver et al., 2016). Moreover, these methods are shown to converge to the minimax solution of the game, thus can be viewed as a counterpart of minimax-Q learning with Monte-Carlo sampling.

4.2.2 Policy-Based Methods

Policy-based reinforcement learning methods introduced in §2.1.2 can also be extended to the multi-agent setting. Instead of finding the fixed point of the Bellman operator, a fair amount of methods only focus on a single agent and aim to maximize the expected return of that agent, disregarding the other agents' policies. Specifically, from the perspective of a single agent, the environment is time-varying as the other agents also adapt their policies. Policy based methods aim to achieve the optimal performance when other agents play arbitrarily by minimizing the *(external) regret*, that is, find a sequence of actions that perform nearly as well as the optimal fixed policy in hindsight. An algorithm with negligible average overall regret is called *no-regret* or *Hannan consistent* (Hannan, 1957). Any Hannan consistent algorithm is known to have the following two desired properties. First, when other agents adopt stationary policies, the time-average policy constructed by the algorithm converges to the best response policy (against the ones used by the other agents). Second, more interestingly, in two-player zero-sum games, when both players adopt Hannan consistent algorithms and both their average overall regrets are no more than ϵ, their time-average policies constitute a 2ϵ-approximate Nash equilibrium (Blum and Mansour, 2007). Thus, any Hannan consistent single-agent reinforcement learning algorithm can be applied to find the Nash equilibria of two-player zero-sum games via *self-play*. Most of these methods belong to one of the following two families: fictitious play (Brown, 1951; Robinson, 1951), and counterfactual regret minimization (Zinkevich et al., 2008), which will be summarized below.

Fictitious play is a classic algorithm studied in game theory, where the players play the game repeatedly and each player adopts a policy that best responds to the average policy of the other agents. This method was originally proposed for solving *normal-form games*, which are a simplification of the Markov games defined in Definition 2.2 with $\mathcal{S}$ being a singleton and $\gamma = 0$. In particular, for any joint policy $\pi \in \Delta(\mathcal{A})$ of the N agents, we let π^{-i} be the marginal policy of all players except for player i. For any $t \geq 1$, suppose the agents have played $\{a_\tau : .1 \leq \tau \leq t\}$ in the first t stages. We define x_t as the empirical distribution of $\{a_\tau : .1 \leq \tau \leq t\}$, i.e., $x_t(a) = t^{-1} \sum_{\tau=1}^{t} \mathbb{1}\{a_t = a\}$ for any $a \in \mathcal{A}$. Then, in the t-th stage, each agent i takes action $a_t^i \in \mathcal{A}^i$ according to the best response policy against x_t^{-i}. In other words, each agent plays the best counter policy against the policy of the other agents inferred from history data. Here, for any $\epsilon > 0$ and any $\pi \in \Delta(\mathcal{A})$, we denote by

$\mathrm{Br}_\epsilon(\pi^{-i})$ the ϵ-best response policy of player i, which satisfies

$$R^i\left(\mathrm{Br}_\epsilon(\pi^{-i}), \pi^{-1}\right) \geq \sup_{\mu \in \Delta(\mathcal{A}^i)} R^i(\mu, \pi^{-i}) - \epsilon. \tag{4.16}$$

Moreover, we define $\mathrm{Br}_\epsilon(\pi)$ as the joint policy $(\mathrm{Br}_\epsilon(\pi^{-1}), \ldots, \mathrm{Br}_\epsilon(\pi^{-N})) \in \Delta(\mathcal{A})$ and suppress the subscript ϵ in Br_ϵ if $\epsilon = 0$. By this notation, regarding each $a \in \mathcal{A}$ as a vertex of $\Delta(\mathcal{A})$, we can equivalently write the fictitious process as

$$x_t - x_{t-1} = (1/t) \cdot (a_t - x_{t-1}), \qquad \text{where} \qquad a_t \sim \mathrm{Br}(x_{t-1}). \tag{4.17}$$

As $t \to \infty$, the updates in (4.17) can be approximately characterized by a differential inclusion (Benaïm et al., 2005)

$$\frac{\mathrm{d}x(t)}{\mathrm{d}t} \in \mathrm{Br}(x(t)) - x(t), \tag{4.18}$$

which is known as the continuous-time fictitious play. Although it is well known that the discrete-time fictitious play in (4.17) is not Hannan consistent (Hart and Mas-Colell, 2001; Young, 1993), it is shown in Monderer et al. (1997); Viossat and Zapechelnyuk (2013) that the continuous-time fictitious play in (4.18) is Hannan consistent. Moreover, using tools from stochastic approximation (Kushner and Yin, 2003; Hart and Mas-Colell, 2001), various modifications of discrete-time fictitious play based on techniques such as smoothing or stochastic perturbations have been shown to converge to the continuous-time fictitious play and are thus Hannan consistent (Fudenberg and Levine, 1995; Hofbauer and Sandholm, 2002; Leslie and Collins, 2006; Benaïm and Faure, 2013; Li and Tewari, 2018). As a result, applying these methods with self-play provably finds a Nash equilibrium of a two-player zero-sum normal form game.

Furthermore, fictitious play methods have also been extended to RL settings without the model knowledge. Specifically, using sequence-form representation, Heinrich et al. (2015) has proposed the first fictitious play algorithm for extensive-form games which is realization-equivalent to the *generalized weakened fictitious play* (Leslie and Collins, 2006) for normal-form games. The pivotal insight is that a convex combination of normal-form policies can be written as a weighted convex combination of behavioral policies using realization probabilities. Specifically, recall that the set of information states of agent i was denoted by $\mathcal{S}^i$. When the game has perfect-recall, each $s^i \in \mathcal{S}^i$ uniquely defines a sequence σ_{s^i} of actions played by agent i for reaching state s^i. Then any behavioral policy π^i of agent i induces a *realization probability* $\mathsf{Rp}(\pi^i; \cdot)$ for each sequence σ of agent i, which is defined by $\mathsf{Rp}(\pi^i; \sigma) = \prod_{(\sigma_{s'}, a) \sqsubseteq \sigma} \pi^i(a|s')$, where the product is taken over all $s' \in \mathcal{S}^i$ and $a \in \mathcal{A}^i$ such that $(\sigma_{s'}, a)$ is a subsequence of σ. Using the notation of realization probability, for any two behavioral policies π and $\widetilde{\pi}$ of agent i, the sum

$$\frac{\lambda \cdot \mathsf{Rp}(\pi, \sigma_{s^i}) \cdot \pi(\cdot|s^i)}{\lambda \cdot \mathsf{Rp}(\pi, \sigma_{s^i}) + (1-\lambda) \cdot \mathsf{Rp}(\widetilde{\pi}, \sigma_{s^i})} + \frac{(1-\lambda) \cdot \mathsf{Rp}(\widetilde{\pi}, \sigma_{s^i}) \cdot \widetilde{\pi}(\cdot|s^i)}{\lambda \cdot \mathsf{Rp}(\pi, \sigma_{s^i}) + (1-\lambda) \cdot \mathsf{Rp}(\widetilde{\pi}, \sigma_{s^i})}, \qquad \forall s^i \in \mathcal{S}^i, \tag{4.19}$$

is the mixture policy of π and $\widetilde{\pi}$ with weights $\lambda \in (0,1)$ and $1-\lambda$, respectively. Then, combining (4.16) and (4.19), the fictitious play algorithm in Heinrich et al. (2015) computes a sequence of policies $\{\pi_t\}_{t \geq 1}$. In particular, in the t-th iteration, any agent i first

compute the ϵ_{t+1}-best response policy $\widetilde{\pi}^i_{t+1} \in \mathrm{Br}_\epsilon(\pi^{-i}_t)$ and then constructs π^i_{t+1} as the mixture policy of π^i_t and $\widetilde{\pi}_{t+1}$ with weights $1-\alpha_{t+1}$ and α_{t+1}, respectively. Here, both ϵ_t and α_t are taken to converge to zero as t goes to infinity, and we further have $\sum_{t\geq 1}\alpha_t = \infty$. We note, however, that although such a method provably converges to a Nash equilibrium of a zero-sum game via self-play, it suffers from the curse of dimensionality due to the need to iterate all states of the game in each iteration. For computational efficiency, Heinrich et al. (2015) has also proposed a data-drive fictitious self-play framework where the best-response is computed via fitted Q-iteration (Ernst et al., 2005; Munos, 2007) for the single-agent RL problem, with the policy mixture being learned through supervised learning. This framework was later adopted by Heinrich and Silver (2014, 2016); Kawamura et al. (2017); Zhang et al. (2019) to incorporate other single RL methods such as deep Q-network (Mnih et al., 2015) and Monte-Carlo tree search (Coulom, 2006; Kocsis and Szepesvári, 2006; Browne et al., 2012). Moreover, in a more recent work, Perolat et al. (2018) has proposed a smooth fictitious play algorithm (Fudenberg and Levine, 1995) for zero-sum stochastic games with simultaneous moves. Their algorithm combines the actor-critic framework (Konda and Tsitsiklis, 2000) with fictitious self-play, and infers the opponent's policy implicitly via policy evaluation. Specifically, when the two players adopt a joint policy $\pi = (\pi^1, \pi^2)$, from the perspective of player 1, it infers π^2 implicitly by estimating $\overline{Q}_{\pi^1,\pi^2}$ via temporal-difference learning (Sutton and Barto, 1987), where $\overline{Q}_{\pi^1,\pi^2}\colon \mathcal{S}\times\mathcal{A}^1 \to \mathbb{R}$ is defined as

$$\overline{Q}_{\pi^1,\pi^2}(s,a^1) = \mathbb{E}\left[\sum_{t\geq 0}\gamma^t R^1(s_t,a_t,s_{t+1}) \,\middle|\, s_0 = s, a^1_0 = a^1, a^2_0 \sim \pi^2(\cdot|s), a_t \sim \pi(\cdot|s_t), \forall t \geq 1\right],$$

which is the action-value function of player 1 marginalized by π^2. Besides, the best response policy is obtained by taking the soft-greedy policy with respect to $\overline{Q}_{\pi^1,\pi^2}$, i.e.,

$$\pi^1(a^1|s) \leftarrow \frac{\exp[\eta^{-1}\cdot\overline{Q}_{\pi^1,\pi^2}(s,a^1)]}{\sum_{a^1\in\mathcal{A}^1}\exp[\eta^{-1}\cdot\overline{Q}_{\pi^1,\pi^2}(s,a^1)]}, \tag{4.20}$$

where $\eta > 0$ is the smoothing parameter. Finally, the algorithm is obtained by performing both policy evaluation and policy update in (4.20) simultaneously using two-timescale updates (Borkar, 2008; Kushner and Yin, 2003), which ensure that the policy updates, when using self-play, can be characterized by an ordinary differential equation whose asymptotically stable solution is a smooth Nash equilibrium of the game.

Another family of popular policy-based methods is based on the idea of *counterfactual regret minimization* (CFR), first proposed in Zinkevich et al. (2008), which has been a breakthrough in the effort to solve large-scale extensive-form games. Moreover, from a theoretical perspective, compared with fictitious play algorithms whose convergence is analyzed asymptotically via stochastic approximation, explicit regret bounds are established using tools from online learning (Cesa-Bianchi and Lugosi, 2006), which yield rates of convergence to the Nash equilibrium. Specifically, when N agents play the extensive-form game for T rounds with $\{\pi_t\colon 1 \leq t \leq T\}$, the *regret* of player i is defined as

$$\mathrm{Reg}^i_T = \max_{\pi^i}\sum_{t=1}^{T}\Big[R^i(\pi^i,\pi^{-i}_t) - R^i(\pi^i_t,\pi^{-i}_t)\Big], \tag{4.21}$$

where the maximum is taken over all possible policies of player i. In the following, before we define the notion of counterfactual regret, we first introduce a few notations. Recall that we had defined the reach probability $\eta_\pi(h)$ in (2.4), which can be factorized into the product of each agent's contribution. That is, for each $i \in \mathcal{U} \cup \{c\}$, we can group the probability terms involving π^i into $\eta^i_\pi(h)$ and write $\eta_\pi(h) = \prod_{i \in \mathcal{N} \cup \{c\}} \eta^i_\pi(h) = \eta^i_\pi(h) \cdot \eta^{-i}_\pi(h)$. Moreover, for any two histories $h, h' \in \mathcal{H}$ satisfying $h \sqsubseteq h'$, we define the *conditional reach probability* $\eta_\pi(h'|h)$ as $\eta_\pi(h')/\eta_\pi(h)$ and define $\eta^i_\pi(h'|h)$ similarly. For any $s \in \mathcal{S}^i$ and any $a \in \mathcal{A}(s)$, we define $\mathcal{Z}(s,a) = \{(h,z) \in \mathcal{H} \times \mathcal{Z} \,|\, h \in s, ha \sqsubseteq z\}$, which contains all possible pairs of history in information state s and terminal history after taking action a at s. Then, the *counterfactual value function* is defined as

$$Q^i_{\mathsf{CF}}(\pi, s, a) = \sum_{(h,z) \in \mathcal{Z}(s,a)} \eta^{-i}_\pi(h) \cdot \eta_\pi(z|ha) \cdot R^i(z), \tag{4.22}$$

which is the expected utility obtained by agent i given that it has played to reached state s. We also define $V^i_{\mathsf{CF}}(\pi, s) = \sum_{a \in \mathcal{A}(s)} Q^i_{\mathsf{CF}}(\pi, s, a) \cdot \pi^i(a|s)$. Then the difference between $Q^i_{\mathsf{CF}}(\pi, s, a)$ and $V^i_{\mathsf{CF}}(\pi, s)$ can be viewed as the value of action a at information state $s \in \mathcal{S}^i$, and *counterfactual regret* of agent i at state s is defined as

$$\mathsf{Reg}^i_T(s) = \max_{a \in \mathcal{A}(s)} \sum_{t=1}^{T} \Big[Q^i_{\mathsf{CF}}(\pi_t, s, a) - V^i_{\mathsf{CF}}(\pi_t, s) \Big], \qquad \forall s \in \mathcal{S}^i. \tag{4.23}$$

Moreover, as shown in Theorem 3 of Zinkevich et al. (2008), counterfactual regrets defined in (4.23) provide an upper bound for the total regret in (4.21):

$$\mathsf{Reg}^i_T \le \sum_{s \in \mathcal{S}^i} \mathsf{Reg}^{i,+}_T(s), \tag{4.24}$$

where we let x^+ denote $\max\{x, 0\}$ for any $x \in \mathbb{R}$. This bound lays the foundation of *counterfactual regret minimization* algorithms. Specifically, to minimize the total regret in (4.21), it suffices to minimize the counterfactual regret for each information state, which can be obtained by any online learning algorithm, such as EXP3 (Auer et al., 2002), Hedge (Vovk, 1990; Littlestone and Warmuth, 1994; Freund and Schapire, 1999), and regret matching (Hart and Mas-Colell, 2000). All these methods ensures that the counterfactual regret is $\mathcal{O}(\sqrt{T})$ for all $s \in \mathcal{S}^i$, which leads to an $\mathcal{O}(\sqrt{T})$ upper bound of the total regret. Thus, applying CFR-type methods with self-play to a zero-sum two-play extensive-form game, the average policy is an $\mathcal{O}(\sqrt{1/T})$-approximate Nash equilibrium after T steps. In particular, the vanilla CFR algorithm updates the policies via regret matching, which yields that $\mathsf{Reg}^i_T(s) \le R^i_{\max} \cdot \sqrt{A^i \cdot T}$ for all $s \in \mathcal{S}^i$, where we have introduced

$$R^i_{\max} = \max_{z \in \mathcal{Z}} R^i(z) - \min_{z \in \mathcal{Z}} R^i(z), \qquad A_i = \max_{h:\, \tau(h) = i} |\mathcal{A}(h)|.$$

Thus, by (4.24), the total regret of agent i is bounded by $R^i_{\max} \cdot |\mathcal{S}^i| \cdot \sqrt{A^i \cdot T}$.

One drawback of vanilla CFR is that the entire game tree needs to be traversed in each iteration, which can be computationally prohibitive. A number of CFR variants have

been proposed since the pioneering work Zinkevich et al. (2008) for improving computational efficiency. For example, Lanctot et al. (2009); Burch et al. (2012); Gibson et al. (2012); Johanson et al. (2012); Lisý et al. (2015); Schmid et al. (2019) combine CFR with Monte-Carlo sampling; Waugh et al. (2015); Morrill (2016); Brown et al. (2019) propose to estimate the counterfactual value functions via regression; Brown and Sandholm (2015); Brown et al. (2017); Brown and Sandholm (2017) improve efficiency by pruning suboptimal paths in the game tree; Tammelin (2014); Tammelin et al. (2015); Burch et al. (2019) analyze the performance of a modification named CFR$^+$, and Zhou et al. (2018) proposes lazy updates with a near-optimal regret upper bound.

Furthermore, it has been shown recently in Srinivasan et al. (2018) that CFR is closely related to policy gradient methods. To see this, for any joint policy π and any $i \in \mathcal{N}$, we define the action-value function of agent i, denoted by Q^i_π, as

$$Q^i_\pi(s,a) = \frac{1}{\eta_\pi(s)} \cdot \sum_{(h,z)\in\mathcal{Z}(s,a)} \eta_\pi(h) \cdot \eta_\pi(z|ha) \cdot R^i(z), \qquad \forall s \in \mathcal{S}^i, \forall a \in \mathcal{A}(s). \tag{4.25}$$

That is, $Q^i_\pi(s,a)$ the expected utility of agent i when the agents follow policy π and agent i takes action a at information state $s \in \mathcal{S}^i$, conditioning on s being reached. It has been shown in Srinivasan et al. (2018) that the Q^i_{CF} in (4.22) is connected with Q^i_π in (4.25) via $Q^i_{\text{CF}}(\pi,s,a) = Q^i_\pi(s,a) \cdot [\sum_{h\in s} \eta^{-i}_\pi(h)]$. Moreover, in the tabular setting where we regard the joint policy π as a table $\{\pi^i(a|s) \colon s \in \mathcal{S}^i, a \in \mathcal{A}(s)\}$, for any $s \in \mathcal{S}^i$ and any $a \in \mathcal{A}(s)$, the policy gradient $R^i(\pi)$ can be written as

$$\frac{\partial R^i(\pi)}{\partial_{\pi^i(a|s)}} = \eta_\pi(s) \cdot Q^i_\pi(s,a) = \eta^i_\pi(s) \cdot Q^i_{\text{CF}}(\pi,s,a), \qquad \forall s \in \mathcal{S}^i, \forall a \in \mathcal{A}(s).$$

As a result, the advantage actor-critic (A2C) algorithm (Konda and Tsitsiklis, 2000) is equivalent to a particular CFR algorithm, where the policy update rule is specified by the *generalized infinitesimal gradient ascent* algorithm (Zinkevich, 2003). Thus, Srinivasan et al. (2018) proves that the regret of the tabular A2C algorithm is bounded by $|\mathcal{S}^i| \cdot [1 + A^i \cdot (R^i_{\max})^2] \cdot \sqrt{T}$. Following this work, Omidshafiei et al. (2019) shows that A2C where the policy is tabular and is parametrized by a softmax function is equivalent to CFR that uses Hedge to update the policy. Moreover, Lockhart et al. (2019) proposes a policy optimization method known as *exploitability descent*, where the policy is updated using actor-critic, assuming the opponent plays the best counter-policy. This method is equivalent to the CFR-BR algorithm (Johanson et al., 2012) with Hedge. Thus, Srinivasan et al. (2018); Omidshafiei et al. (2019); Lockhart et al. (2019) show that actor-critic and policy gradient methods for MARL can be formulated as CFR methods and thus convergence to a Nash equilibrium of a zero-sum extensive-form game is guaranteed.

In addition, besides fictitious play and CFR methods introduced above, multiple policy optimization methods have been proposed for special classes of two-player zero-sum stochastic games or extensive form games. For example, Monte-Carlo tree search methods have been proposed for perfect-information extensive games with simultaneous moves. It has been shown in Schaeffer et al. (2009) that the MCTS methods with UCB-type action selection rules, introduced in §4.2.1, fail to converge to a Nash equilibrium in simultaneous-move games, as UCB does not take into consideration the possibly adversarial moves of

the opponent. To remedy this issue, Lanctot et al. (2013); Lisý et al. (2013); Tak et al. (2014); Kovařík and Lisý (2018) have proposed to adopt stochastic policies and using Hannan consistent methods such as EXP3 (Auer et al., 2002) and regret matching (Hart and Mas-Colell, 2000) to update the policies. With self-play, Lisý et al. (2013) shows that the average policy obtained by MCTS with any ϵ-Hannan consistent policy update method converges to an $\mathcal{O}(D^2 \cdot \epsilon)$-Nash equilibrium, where D is the maximal depth.

Finally, there are surging interests in investigating policy gradient-based methods in *continuous* games, i.e., the games with continuous state-action spaces. With policy parameterization, finding the NE of zero-sum Markov games becomes a nonconvex-nonconcave saddle-point problem in general (Mazumdar and Ratliff, 2018; Mazumdar et al., 2019; Zhang et al., 2019; Bu et al., 2019). This hardness is inherent, even in the simplest linear quadratic setting with linear function approximation (Zhang et al., 2019; Bu et al., 2019). As a consequence, most of the convergence results are *local* (Mescheder et al., 2017; Mazumdar and Ratliff, 2018; Adolphs et al., 2018; Daskalakis and Panageas, 2018; Mertikopoulos et al., 2019; Fiez et al., 2019; Mazumdar et al., 2019; Jin et al., 2019), in the sense that they address the convergence behavior around local NE points. Still, it has been shown that the vanilla gradient-descent-ascent (GDA) update, which is equivalent to the policy gradient update in MARL, fails to converge to local NEs, for either the non-convergent behaviors such as limit cycling (Mescheder et al., 2017; Daskalakis and Panageas, 2018; Balduzzi et al., 2018; Mertikopoulos et al., 2019), or the existence of non-Nash stable limit points for the GDA dynamics (Adolphs et al., 2018; Mazumdar et al., 2019). Consensus optimization (Mescheder et al., 2017), symplectic gradient adjustment (Balduzzi et al., 2018), and extragradient method (Mertikopoulos et al., 2019) have been advocated to mitigate the oscillatory behaviors around the equilibria; while Adolphs et al. (2018); Mazumdar et al. (2019) exploit the curvature information so that all the stable limit points of the proposed updates are local NEs. Going beyond Nash equilibria, Jin et al. (2019); Fiez et al. (2019) consider gradient-based learning for *Stackelberg equilibria*, which correspond to only the one-sided equilibrium solution in zero-sum games, i.e., either minimax or maximin, as the order of which player acts first is vital in nonconvex-nonconcave problems. Jin et al. (2019) introduces the concept of *local minimax* point as the solution, and shows that GDA converges to local minimax points under mild conditions. Fiez et al. (2019) proposes a two-timescale algorithm where the follower uses a gradient-play update rule, instead of an exact best response strategy, which has been shown to converge to the Stackelberg equilibria. Under a stronger assumption of *gradient dominance*, Sanjabi et al. (2018); Nouiehed et al. (2019) have shown that nested gradient descent methods converge to the stationary points of the outer-loop, i.e., minimax, problem at a sublinear rate.

We note that these convergence results have been developed for *general* continuous games with *agnostic* cost/reward functions, meaning that the functions may have various forms, so long as they are *differentiable*, sometimes even *(Lipschitz) smooth*, w.r.t. each agent's policy parameter. For MARL, this is equivalent to requiring differentiability/smoothness of the long-term *return*, which relies on the properties of the game, as well as of the policy parameterization. Such an assumption is generally very restrictive. For example, the Lipschitz smoothness assumption fails to hold for LQ games (Zhang et al., 2019; Mazumdar et al., 2019; Bu et al., 2019), a special type of MGs. Fortunately, thanks

to the special structure of the LQ setting, Zhang et al. (2019) has proposed several projected nested policy gradient methods that are guaranteed to have *global* convergence to the NE, with convergence rates established. This appears to be the first-of-its-kind result in MARL. Very recently, Bu et al. (2019) improves the results by removing the projection step in the updates, for a more general class of such games.

4.3 Mixed Setting

In stark contrast with the fully collaborative and fully competitive settings, the mixed setting is notoriously challenging and thus rather less well understood. Even in the simplest case of a two-player general sum normal-form game, finding a Nash equilibrium is PPAD-complete (Chen et al., 2009). Moreover, Zinkevich et al. (2006) has proved that value-iteration methods fail to find stationary Nash or correlated equilibria for general-sum Markov games. Recently, it is shown that vanilla policy-gradient methods avoid a non-negligible subset of Nash equilibria in general-sum continuous games (Mazumdar and Ratliff, 2018), including the LQ general-sum games (Mazumdar et al., 2019). Thus, additional structures on either the games or the algorithms need to be exploited, to ascertain provably convergent MARL in the mixed setting.

Value-Based Methods

Under relatively stringent assumptions, several value-based methods that extend Q-learning (Watkins and Dayan, 1992) to the mixed setting are guaranteed to find an equilibrium. In particular, Hu and Wellman (2003) has proposed the Nash-Q learning algorithm for general-sum Markov games, where one maintains N action-value functions $Q_{\mathcal{N}} = (Q^1, \ldots, Q^N) \colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}^N$ for all N agents, which are updated using sample-based estimator of a Bellman operator. Specifically, letting $R_{\mathcal{N}} = (R^1, \ldots, R^N)$ denote the reward functions of the agents, Nash-Q uses the following Bellman operator:

$$(\mathcal{T}^* Q_{\mathcal{N}})(s,a) = \mathbb{E}_{s' \sim \mathcal{P}(\cdot|s,a)} \Big\{ R_{\mathcal{N}}(s,a,s') + \gamma \cdot \texttt{Nash}\Big[Q_{\mathcal{N}}(s',\cdot)\Big] \Big\}, \quad \forall (s,a) \in \mathcal{S} \times \mathcal{A}, \tag{4.26}$$

where $\texttt{Nash}[Q_{\mathcal{N}}(s',\cdot)]$ is the objective value of the Nash equilibrium of the stage game with rewards $\{Q_{\mathcal{N}}(s',a)\}_{a \in \mathcal{A}}$. For zero-sum games, we have $Q^1 = -Q^2$ and thus the Bellman operator defined in (4.26) is equivalent to the one in (4.14) used by minimax-Q learning (Littman, 1994). Moreover, Hu and Wellman (2003) establishes convergence to Nash equilibrium under the restrictive assumption that $\texttt{Nash}[Q_{\mathcal{N}}(s',\cdot)]$ in each iteration of the algorithm has unique Nash equilibrium. In addition, Littman (2001) has proposed the Friend-or-Foe Q-learning algorithm where each agent views the other agent as either a "friend" or a "foe". In this case, $\texttt{Nash}[Q_{\mathcal{N}}(s',\cdot)]$ can be efficiently computed via linear programming. This algorithm can be viewed as a generalization of minimax-Q learning, and Nash equilibrium is guaranteed for two-player zero-sum games and coordination games with unique equilibria. Furthermore, Greenwald et al. (2003) has proposed correlated Q-learning, which replaces $\texttt{Nash}[Q_{\mathcal{N}}(s',\cdot)]$ in (4.26) by computing a correlated equilibrium (Aumann, 1974), a more general equilibrium concept than Nash equilibrium. In a recent work, Perolat et al. (2017) has proposed a batch RL method to find an approximate Nash equilibrium via Bellman residue minimization (Maillard et al., 2010). They have proved

that the global minimizer of the empirical Bellman residue is an approximate Nash equilibrium, followed by the error propagation analysis for the algorithm. Also in the batch RL regime, Zhang et al. (2018b) has considered a simplified mixed setting for decentralized MARL: two teams of cooperative networked agents compete in a zero-sum Markov game. A decentralized variant of FQI, where the agents within one team cooperate to solve (4.3) while the two teams essentially solve (4.15), is proposed. Finite-sample error bounds have then been established for the proposed algorithm.

To address the scalability issue, independent learning is preferred, which, however, fails to converge in general (Tan, 1993). Arslan and Yüksel (2017) has proposed *decentralized Q-learning*, a two timescale modification of Q-learning, that is guaranteed to converge to the equilibrium for *weakly acyclic Markov games* almost surely. Each agent therein only observes local action and reward, and neither observes nor keeps track of others' actions. All agents are instructed to use the same stationary *baseline policy* for many consecutive stages, named *exploration phase*. At the end of the *exploration phase*, all agents are *synchronized* to update their baseline policies, which makes the environment stationary for long enough, and enables the convergence of Q-learning based methods. Note that these algorithms can also be applied to the cooperative setting, as these games include Markov teams as a special case.

Policy-Based Methods

For continuous games, due to the general negative results therein, Mazumdar and Ratliff (2018) introduces a new class of games, *Morse-Smale games*, for which the gradient dynamics correspond to gradient-like flows. Then, definitive statements on almost sure convergence of PG methods to either limit cycles, Nash equilibria, or non-Nash fixed points can be made, using tools from dynamical systems theory. Moreover, Balduzzi et al. (2018); Letcher et al. (2019) have studied the second-order structure of game dynamics, by decomposing it into two components.The first one, named symmetric component, relates to potential games, which yields gradient descent on some implicit function; the second one, named antisymmetric component, relates to *Hamiltonian games* that follows some conservation law, motivated by classical mechanical systems analysis. The fact that gradient descent converges to the Nash equilibrium of both types of games motivates the development of the Symplectic Gradient Adjustment (SGA) algorithm that finds *stable fixed points* of the game, which constitute all local Nash equilibria for zero-sum games, and only a subset of local NE for general-sum games. Chasnov et al. (2019) provides finite-time local convergence guarantees to a neighborhood of a *stable* local Nash equilibrium of continuous games, in both deterministic setting, with exact PG, and stochastic setting, with unbiased PG estimates. Additionally, Chasnov et al. (2019) has also explored the effects of *non-uniform* learning rates on the learning dynamics and convergence rates. Fiez et al. (2019) has also considered *general-sum* Stackelberg games, and shown that the same two-timescale algorithm update as in the zero-sum case now converges almost surely to the stable attractors only. It has also established finite-time performance for local convergence to a neighborhood of a stable Stackelberg equilibrium. In complete analogy to the zero-sum class, these convergence results for continuous games do not apply to MARL in Markov games directly, as they are built upon the differentiability/smoothness of the long-term return, which may not hold for general MGs, for example, LQ games (Mazum-

dar et al., 2019).

Other than continuous games, the policy-based methods summarized in §4.2.2 can also be applied to the mixed setting via self-play. The validity of such an approach is based a fundamental connection between game theory and online learning – If the external regret of each agent is no more than ϵ, then their average policies constitute an ϵ-approximate *coarse correlated equilibrium* (Hart and Mas-Colell, 2000, 2001, 2003). Thus, although in general we are unable to find a Nash equilibrium, policy optimization with self-play guarantees to find a coarse correlated equilibrium.

Mean-Field Regime

The scalability issue in the non-cooperative setting can also be alleviated in the mean-field regime, as the cooperative setting discussed in §4.1.1. For general-sum games, Yang et al. (2018) has proposed a modification of the Nash-Q learning algorithm where the actions of other agents are approximated by their empirical average. That is, the action value function of each agent i is parametrized by $Q^i(s, a^i, \mu_{a^{-i}})$, where $\mu_{a^{-i}}$ is the empirical distribution of $\{a_j : j \neq i\}$. Asymptotic convergence of this mean-field Nash-Q learning algorithm has also been established.

Besides, most mean-field RL algorithms are focused on addressing the *mean-field game* model. In mean-field games, each agent i has a local state $s^i \in \mathcal{S}$ and a local action $a^i \in \mathcal{S}$, and the interaction among other agents is captured by an aggregated effect μ, also known as the mean-field, which is a functional of the empirical distribution of the local states and actions of the agents. Specifically, at the t-th time step, when agent i takes action a_t^i at state s_t^i and the mean-field term is μ_t, it receives an immediate reward $R(s_t^i, a_t^i, \mu_t)$ and its local state evolves into $s_{t+1}^i \sim \mathcal{P}(\cdot|s_t^i, a_t^i, \mu_t) \in \Delta(\mathcal{S})$. Thus, from the perspective of agent i, instead of participating in a multi-agent game, it is faced with a time-varying MDP parameterized by the sequence of mean-field terms $\{\mu_t\}_{t\geq\infty}$, which in turn is determined by the states and actions of all agents. The solution concept in MFGs is the Mean-field equilibrium, which is a sequence of pairs of policy and mean-field terms $\{\pi_t^*, \mu_t^*\}_{t\geq 0}$ that satisfy the following two conditions: (1) $\pi^* = \{\pi_t^*\}_{t\geq 0}$ is the optimal policy for the time-varying MDP specified by $\mu^* = \{\mu_t^*\}_{t\geq 0}$, and (2) μ^* is generated when each agent follows policy π^*. The existence of the mean-field equilibrium for discrete-time MFGs has been studied in Saldi et al. (2018, 2019); Saldi (2019); Saldi et al. (2018) and their constructive proofs exhibit that the mean-field equilibrium can be obtained via a fixed-point iteration. Specifically, one can construct a sequence of policies and mean-field terms $\{\pi^{(i)}\}_{i\geq 1}$ and $\{\mu^{(i)}\}_{i\geq 1}$ such that $\{\pi^{(i)}\}_{t\geq 0}$ solves the time-varying MDP specified by $\mu^{(i)}$, and $\mu^{(i+1)}$ is generated when all players adopt policy $\pi^{(i)}$. Following this agenda, various model-free RL methods are proposed for solving MFGs where $\{\pi^{(i)}\}_{i\geq 1}$ is approximately solved via single-agent RL such as Q-learning (Guo et al., 2019) and policy-based methods (Subramanian and Mahajan, 2019; Fu et al., 2019), with $\{\mu^{(i)}\}_{i\geq 1}$ being estimated via sampling. In addition, Hadikhanloo and Silva (2019); Elie et al. (2019) recently propose fictitious play updates for the mean-field state where we have $\mu^{(i+1)} = (1-\alpha^{(i)})\cdot\mu^{(i)} + \alpha^{(i)}\cdot\hat{\mu}^{(i+1)}$, with $\alpha^{(i)}$ being the learning rate and $\hat{\mu}^{(i+1)}$ being the mean-field term generated by policy $\pi^{(i)}$. Note that the aforementioned works focus on the settings with either *finite* horizon (Hadikhanloo and Silva, 2019; Elie et al., 2019) or *stationary* mean-field equilibria (Guo et al., 2019; Subramanian and Mahajan, 2019; Fu et al., 2019) only. Instead, recent works Anahtarci et al.

Figure 3: Four representative applications of recent successes of MARL: unmanned aerial vehicles, game of Go, Poker games, and team-battle video games.

(2019); Zaman et al. (2020) consider possibly non-stationary mean-field equilibrium in infinite-horizon settings, and develop equilibrium computation algorithms that lay foundations for model-free RL algorithms.

5 Application Highlights

In this section, we briefly review the recent empirical successes of MARL driven by the methods introduced in the previous section. In the following, we focus on the three MARL settings reviewed in §4 and highlight four most representative and practical applications in each setting, as illustrated in Figure 3.

5.1 Cooperative Setting

Unmanned Aerial Vehicles

One prominent application of MARL is the control of practical multi-agent systems, most of which are cooperative and decentralized. Examples of the scenarios include robot team navigation (Corke et al., 2005), smart grid operation (Dall'Anese et al., 2013), and control of mobile sensor networks (Cortes et al., 2004). Here we choose unmanned aerial vehicles (UAVs) (Yang and Liu, 2018; Pham et al., 2018; Tožička et al., 2018; Shamsoshoara et al., 2019; Cui et al., 2019; Qie et al., 2019), a recently surging application scenario of multi-agent autonomous systems, as one representative example. Specifically, a team of UAVs are deployed to accomplish a cooperation task, usually without the coordination of any central controller, i.e., in a decentralized fashion. Each UAV is normally equipped with communication devices, so that they can exchange information with some of their teammates, provided that they are inside its sensing and coverage range. As a consequence, this application naturally fits in the decentralized paradigm with networked agents we advocated in §4.1.2, which is also illustrated in Figure 2 (b). Due to the high-mobility of UAVs, the communication links among agents are indeed *time-varying* and fragile, making (online) cooperation extremely challenging. Various challenges thus arise in the context of cooperative UAVs, some of which have recently been addressed by MARL.

In Yang and Liu (2018), the UAVs' optimal links discovery and selection problem is considered. Each UAV $u \in \mathcal{U}$, where $\mathcal{U}$ is the set of all UAVs, has the capability to perceive the local available channels and select a connected link over a common channel shared by another agent $v \in \mathcal{U}$. Each UAV u has its local set of channels $\mathcal{C}_u$ with $\mathcal{C}_u \bigcap \mathcal{C}_v \neq \emptyset$ for any

u, v, and a connected link between two adjacent UAVs is built if they announce their messages on the same channel simultaneously. Each UAV's local state is whether the previous message has been successfully sent, and its action is to choose a pair (v, ch_u), with $v \in \mathcal{T}_u$ and $ch_u \in \mathcal{C}_u$, where $\mathcal{T}_u$ is the set of teammates that agent u can reach. The availability of local channels $ch_u \in \mathcal{C}_u$ is modeled as probabilistic, and the reward $\mathcal{R}^u$ is calculated by the number of messages that are successfully sent. Essentially, the algorithm in Yang and Liu (2018) is based on independent Q-learning (Tan, 1993), but with two heuristics to improve the tractability and convergence performance: by *fractional slicing*, it treats each dimension (fraction) of the action space independently, and estimate the actual Q-value by the average of that for all fractions; by *mutual sampling*, it shares both state-action pairs and a mutual Q-function parameter. Pham et al. (2018) addresses the problem of *field coverage*, where the UAVs aim to provide a full coverage of an unknown field, while minimizing the overlapping sections among their field of views. Modeled as a Markov team, the overall state s is the concatenation of all local states s_i, which are defined as its 3-D position coordinates in the environment. Each agent chooses to either head different directions, or go up and down, yielding 6 possible actions. Modeled as a Markov team, a multi-agent Q-learning over the *joint* action space is developed, with linear function approximation. In contrast, Shamsoshoara et al. (2019) focuses on spectrum sharing among a network of UAVs. Under a remote sensing task, the UAVs are categorized into two clusters: the relaying ones that provide relay services and gain spectrum access for the remaining ones, which perform the sensing task. Such a problem can be modeled as a *deterministic* MMDP, which can thus be solved by distributed Q-learning proposed in Lauer and Riedmiller (2000), with optimality guaranteed. Moreover, Qie et al. (2019) considers the problem of *simultaneous* target-assignment and path-planning for multiple UAVs. In particular, a team of UAVs $U_i \in \mathbf{U}$, with each U_i's position at time t given by $(x_i^U(t), y_i^U(t))$, aim to cover all the targets $T_j \in \mathbf{T}$ without collision with the threat areas $D_i \in \mathbf{D}$, as well as with other UAVs. For each U_i, a path P_i is planned as $P_i = \{(x_i^U(0), y_i^U(0), \cdots, x_i^U(n), y_i^U(n))\}$, and the length of P_i is denoted by d_i. Thus, the goal is to minimize $\sum_i d_i$ while the collision-free constraints are satisfied. By penalizing the collision in the reward function, such a problem can be characterized as one with a mixed MARL setting that contains both cooperative and competitive agents. Hence, the MADDPG algorithm proposed in Lowe et al. (2017) is adopted, with centralized-learning-decentralized-execution. Two other tasks that can be tackled by MARL include resource allocation in UAV-enabled communication networks, using Q-learning based method (Cui et al., 2019), aerial surveillance and base defense in UAV fleet control, using policy optimization method in a purely centralized fashion (Tožička et al., 2018).

Learning to Communicate

Another application of cooperative MARL aims to foster communication and coordination among a team of agents without explicit human supervision. Such a type of problems is usually formulated as a multi-agent POMDP involving N agents, which is similar to the Markov game introduced in Definition 2.2 except that each agent cannot observe the state $s \in \mathcal{S}$ and that each agent has the same reward function $\mathcal{R}$. More specifically, we assume that each agent $i \in \mathcal{N}$ receives observations from set $\mathcal{Y}^i$ via a noisy observation channel $\mathcal{O}^i : \mathcal{S} \to \mathcal{P}(\mathcal{Y}_i)$ such that agent i observes a random variable $y^i \sim \mathcal{O}^i(\cdot|s)$ when the

environment is at state s. Note that this model can be viewed as a POMDP when there is a central planner that collects the observations of each agent and decides the actions for each agent. Due to the noisy observation channels, in such a model the agents need to communicate with each other so as to better infer the underlying state and make decisions that maximize the expected return shared by all agents. Let $\mathcal{N}_t^i \subseteq \mathcal{N}$ be the neighbors of agent i at the t-th time step, that is, agent i is able to receive a message $m_t^{j\to i}$ from any agent $j \in \mathcal{N}_t^i$ at time t.We let I_t^i denote the information agent i collects up to time t, which is defined as

$$I_t^i = \left\{\left(o_\ell^i, \{a_\ell^j\}_{j\in\mathcal{N}}, \{m_\ell^{j\to i}\}_{j\in\mathcal{N}_\ell^i}\right) : \ell \in \{0,\dots,t-1\}\right\} \bigcup \left\{o_t^i,\right\}, \tag{5.1}$$

which contains its history collected in previous time steps and the observation received at time t. With the information I_t^i, agent i takes an action $a_t^i \in \mathcal{A}^i$ and also broadcasts messages $m_t^{i\to j}$ to all agents j such that $i \in \mathcal{N}_t^j$. That is, the policy π_t^i of agent i is a mapping from $\mathcal{I}_t^i$ to a (random) action $\widetilde{a}_t^i = (a_t^i, \{m_t^{i\to j} : i \in \mathcal{N}_t^j\})$, i.e., $\widetilde{a}_t^i \sim \pi_t^i(\cdot \mid I_t^i)$. Notice that the size of information set I_t^i grows as t grows. To handle the memory issue, it is common to first embed I_t^i in a fixed latent space via recurrent neural network (RNN) or Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and define the value and policy functions on top of the embedded features. Moreover, most existing works in this line of research adopt the paradigm of centralized learning and utilize techniques such as weight-sharing or attention mechanism (Vaswani et al., 2017) to increase computational efficiency. With centralized learning, single-agent RL algorithms such as Q-learning and actor-critic are readily applicable.

In particular, Foerster et al. (2016) first proposes to tackle the problem of learning to communicate via deep Q-learning. They propose to use two Q networks that govern taking action $a^i \in \mathcal{A}$ and producing messages separately. Their training algorithm is an extension of the deep recurrent Q-learning (DRQN) (Hausknecht and Stone, 2015), which combines RNN and deep Q-learning (Mnih et al., 2015). Following Foerster et al. (2016), various works (Jorge et al., 2016; Sukhbaatar et al., 2016; Havrylov and Titov, 2017; Das et al., 2017; Peng et al., 2017; Mordatch and Abbeel, 2018; Jiang and Lu, 2018; Jiang et al., 2018; Celikyilmaz et al., 2018; Das et al., 2018; Lazaridou et al., 2018; Cogswell et al., 2019) have proposed a variety of neural network architectures to foster communication among agents. These works combine single-agent RL methods with novel developments in deep learning, and demonstrate their performances via empirical studies. Among these works, Das et al. (2017); Havrylov and Titov (2017); Mordatch and Abbeel (2018); Lazaridou et al. (2018); Cogswell et al. (2019) have reported the emergence of computational communication protocols among the agents when the RL algorithm is trained from scratch with text or image inputs. We remark that the algorithms used in these works are more akin to single-agent RL due to centralized learning. For more details overviews of multi-agent communication, we refer the interested readers to Section 6 of Oroojlooy Jadid and Hajinezhad (2019) and Section 3 of Hernandez-Leal et al. (2018).

5.2 Competitive Setting

In terms of the competitive setting, in the following, we highlight the recent applications of MARL to *the game of Go* and *Texas hold'em poker*, which are archetypal instances of two-player perfect-information and partial-information extensive-form games, respectively.

The Game of Go

The game of Go is a board game played by two competing players, with the goal of surrounding more territory on the board than the opponent. These two players have access to white or black stones respectively, and take turns placing their stones on a 19×19 board, representing their territories. In each move, a player can place a stone to any of the total 361 positions on the board that is not already taken by a stone. Once placed on the board, the stones cannot be moved. But the stones will be removed from the board when completely surrounded by opposing stones. The game terminates when neither of the players is unwilling or unable to make a further move, and the winner is determined by counting the area of the territory and the number of stones captured by the players.

The game of Go can be viewed as a two-player zero-sum Markov game with deterministic state transitions, and the reward only appears at the end of the game. The state of this Markov game is the current configuration of the board and the reward is either one or minus one, representing either a win or a loss, respectively. Specifically, we have $r^1(s) + r^2(s) = 0$ for any state $s \in \mathcal{S}$, and $r^1(s), r^2(s) \in \{1, -1\}$ when s is a terminating state, and $r^1(s) = r^2(s) = 0$ otherwise. Let $V_*^i(s)$ denote the optimal value function of player $i \in \{1, 2\}$. Thus, in this case, $[1 + V^i(s)]/2$ is the probability of player $i \in \{1, 2\}$ winning the game when the current state is s and both players follow the Nash equilibrium policies thereafter. Moreover, as this Markov game is turn-based, it is known that the Nash equilibrium policies of the two players are deterministic (Hansen et al., 2013). Furthermore, since each configuration of the board can be constructed from a sequence of moves of the two players due to deterministic transitions, we can also view the game of Go as an extensive-form game with perfect information. This problem is notoriously challenging due to the gigantic state space. It is estimated in Allis (1994) that the size of state space exceeds 10^{360}, which forbids the usage of any traditional reinforcement learning or searching algorithms.

A significant breakthrough has been made by the *AlphaGo* introduced in Silver et al. (2016), which is the first computer Go program that defeats a human professional player on a full-sized board. AlphaGo integrates a variety of ideas from deep learning and reinforcement learning and tackles the challenge of huge state space by representing the policy and value functions using deep convolutional neural networks (CNN) (Krizhevsky et al., 2012). Specifically, both the policy and value networks are 13-layer CNNs with the same architecture, and a board configuration is represented by 48 features. Thus, both the policy and value networks take inputs of size $19 \times 19 \times 48$. These two networks are trained through a novel combination of supervised learning from human expert data and reinforcement learning from Monte-Carlo tree search (MCTS) and self-play. Specifically, in the first stage, the policy network is trained by supervised learning to predict the actions made by the human players, where the dataset consists of 30 million positions from the KGS Go server. That is, for any state-action pair (s, a) in the dataset, the action a is treated as the response variable and the state s is regarded as the covariate. The weights

of the policy network is trained via stochastic gradient ascent to maximize the likelihood function. After initializing the policy network via supervised learning, in the second stage of the pipeline, both the policy and value networks are trained via reinforcement learning and self-play. In particular, new data are generated by games played between the current policy network and a random previous iteration of the policy network. Moreover, the policy network is updated following policy gradient, and the value network aims to find the value function associated with the policy network and is updated by minimizing the mean-squared prediction error. Finally, when playing the game, the current iterates of the policy and value networks are combined to produce an improved policy by lookahead search via MCTS. The actual action taken by AlphaGo is determined by such an MCTS policy. Moreover, to improve computational efficiency, AlphaGo uses an asynchronous and distributed version of MCTS to speed up simulation.

Since the advent of AlphaGo, an improved version, known as AlphaGo Zero, has been proposed in Silver et al. (2017). Compared with the vanilla AlphaGo, AlphaGo Zero does not use supervised learning to initialize the policy network. Instead, both the policy and value networks are trained from scratch solely via reinforcement learning and self-play. Besides, instead of having separate policy and value functions share the same network architecture, in AlphaGo Zero, these two networks are aggregated into a single neural network structure. Specifically, the policy and value functions are represented by $(p(s), V(s)) = f_\theta(s)$, where $s \in \mathcal{S}$ is the state which represents the current board, f_θ is a deep CNN with parameter θ, $V(s)$ is a scalar that corresponds to the value function, and $p(s)$ is a vector which represents the policy, i.e., for each entry $a \in \mathcal{A}$, $p_a(s)$ is the probability of taking action a at state s. Thus, under such a network structure, the policy and value networks automatically share the same low-level representations of the states. Moreover, the parameter θ of network f_θ is trained via self-play and MCTS. Specifically, at each time-step t, based on the policy p and value V given by f_{θ_t}, an MCTS policy π_t can be obtained and a move is executed following policy $\pi_t(s_t)$. Such a simulation procedure continues until the current game terminates. Then the outcome of the t-th time-step, $z_t \in \{1, -1\}$, is recorded, according to the perspective of the player at time-step t. Then the parameter θ is updated by a stochastic gradient step on a loss function ℓ_t, which is defined as

$$\ell_t(\theta) = [z - V(s_t)]^2 - \pi_t^\top \log p(s_t) + c \cdot \|\theta\|_2^2, \qquad (p(\cdot), V(\cdot)) = f_\theta(\cdot).$$

Thus, ℓ_t is the sum of the mean-squared prediction error of the value function, cross-entropy loss between the policy network and the MCTS policy, and a weight-decay term for regularization. It is reported that AlphaGo Zero has defeated the strongest versions of the previous AlphaGo and that it also has demonstrated non-standard Go strategies that had not been discovered before. Finally, the techniques adopted in AlphaGo Zero has been generalized to other challenging board games. Specifically, Silver et al. (2018) proposes the AlphaZero program that is trained by self-play and reinforcement learning with zero human knowledge and achieves superhuman performances in the games of chess, shogi, and Go.

Texas Hold'em Poker

Another remarkable applicational achievement of MARL in the competitive setting focuses on developing artificial intelligence in the Texas hold'em poker, which is one of

the most popular variations of the poker. Texas hold'em is usually played by a group of two or more players, where each player is first dealt with two *private cards* face down. Then five *community cards* are dealt face up in three rounds. In each round. each player has four possible actions – *check*, *call*, *raise*, and *fold*. After all the cards are dealt, each player who has not folded have seven cards in total, consisting of five community cards and two private cards. Each of these players then finds the best five-card poker hand out of all combinations of the seven cards. The player with the best hand is the winner and wins all the money that the players wager for that hand, which as also known as the *pot*. Note that each hand of Texas hold'em terminates after three rounds, and the payoffs of the player are only known after the hand ends. Also notice that each player is unaware of the private cards of the rest of the players. Thus, Texas hold'em is an instance of multi-player extensive-form game with incomplete information. The game is called *heads-up* when there are only two players. When both the bet sizes and the amount of allowed raises are fixed, the game is called *limit hold'em*. In the no-limit hold'em, however, each player may bet or raise any amount up to all of the money the player has at the table, as long as it exceeds the previous bet or raise.

There has been quest for developing superhuman computer poker programs for over two decades (Billings et al., 2002; Rubin and Watson, 2011). Various methods have been shown successful for simple variations of poker such as Kuhn poker (Kuhn, 1950) and Leduc hold'em (Southey et al., 2005). However, the full-fledged Texas hold'em is much more challenging and several breakthroughs have been achieved only recently. The simplest version of Texas hold'em is *heads-up limit hold'em* (HULHE), which has 3.9×10^{14} information sets in total (Bowling et al., 2015), where a player is required to take an action at each information set. Bowling et al. (2015) has for the first time reported solving HULHE to approximate Nash equilibrium via CFR$^+$ (Tammelin, 2014; Tammelin et al., 2015), a variant of counterfactual regret minimization (Zinkevich et al., 2008). Subsequently, other methods such as Neural Fictitious Self-Play (Heinrich and Silver, 2016) and Monte-Carlo tree search with self-play (Heinrich and Silver, 2015) have also been adopted to successfully solve HULHE.

Despite these breakthroughs, *heads-up no-limit hold'em* (HUNL) has remained open until recently, which has more than 6×10^{161} information sets, an astronomical number. Thus, in HUNL, it is impossible (in today's computational power) to traverse all information sets, making it inviable to apply CFR$^+$ as in Bowling et al. (2015). Ground-breaking achievements have recently been made by *DeepStack* (Moravčík et al., 2017) and *Libratus* (Brown and Sandholm, 2018), two computer poker programs developed independently, which defeat human professional poker players in HUNL for the first time. Both of these programs adopt CFR as the backbone of their algorithmic frameworks, but adopt different strategies for handling the gigantic size of the game. In particular, DeepStack applies deep learning to learn good representations of the game and propose *deep counterfactual value networks* to integrate deep learning and CFR. Moreover, DeepStack adopts limited depth lookahead planning to reduce the gigantic 6×10^{161} information sets to no more than 10^{17} information sets, thus making it possible to enumerate all information sets. In contrast, *Libratus* does not utilize any deep learning techniques. Instead, it reduces the size of the game by computation of an abstraction of the game, which is possible since many of the information sets are very similar. Moreover, it further reduces the complexity using the

sub-game decomposition technique (Burch et al., 2014; Moravcik et al., 2016; Brown and Sandholm, 2017) for imperfect-information games and by constructing fine-grained abstractions of the sub-games. When the abstractions are constructed, an improved version of the Monte-Carlo CFR (Lanctot et al., 2009; Burch et al., 2012; Gibson et al., 2012) is utilized to compute the policy. Furthermore, very recently, based upon *Libratus*, Brown and Sandholm (2019) has proposed *Pluribus*, a computer poker program that has been shown to be stronger than top human professionals in no-limit Texas hold'em poker with six players. The success of Pluribus can be attributed to the following techniques that have appeared in the literature: abstraction and sub-game decomposition for large-scale imperfect-information games, Monte-Carlo CFR, self-play, and depth-limited search.

Other Applications

Furthermore, another popular testbed of MARL is the StarCraft II (Vinyals et al., 2017), which is an immensely popular multi-player real-strategy computer game. This game can be formulated as a multi-agent Markov game with partial observation, where each player has only limited information of the game state. Designing reinforcement learning systems for StarCraft II is extremely challenging due to the needs to make decisions under uncertainty and incomplete information, to consider the optimal strategy in the long-run, and to design good reward functions that elicits learning. Since released, both the full-game and sub-game version of StarCraft II have gained tremendous research interest. A breakthrough in this game was achieved by *AlphaStar*, recently proposed in Vinyals et al. (2019), which has demonstrated superhuman performances in zero-sum two-player full-game StarCraft II. Its reinforcement learning algorithm combines LSTM for the parametrization of policy and value functions, asynchronous actor-critic (Mnih et al., 2016) for policy updates, and neural fictitious self-play (Heinrich and Silver, 2016) for equilibrium finding.

5.3 Mixed Settings

Compared to the cooperative and competitive settings, research on MARL under the mixed setting is rather less explored. One application in this setting is multi-player poker. As we have mentioned in §5.2, Pluribus introduced in Brown and Sandholm (2019) has demonstrated superhuman performances in six-player no-limit Texas hold'em. In addition, as an extension of the problem of learning to communicate, introduced in §5.1, there is a line of research that aims to apply MARL to tackle learning social dilemmas, which is usually formulated as a multi-agent stochastic game with partial information. Thus, most of the algorithms proposed under these settings incorporate RNN or LSTM for learning representations of the histories experience by the agent, and the performances of these algorithms are usually exhibited using experimental results; see, e.g., Leibo et al. (2017); Lerer and Peysakhovich (2017); Hughes et al. (2018), and the references therein.

Moreover, another example of the mixed setting is the case where the agents are divided into two opposing teams that play zero-sum games. The reward of a team is shared by each player within this team. Compared with two-player zero-sum games, this setting is more challenging in that both cooperation among teammates and competition against the opposing team need to be taken into consideration. A prominent testbed of this case

is the *Dota* 2 video game, where each of two teams, each with five players, aims to conquer the base of the other team and defend its own base. Each player independently controls a powerful character known as the *hero*, and only observes the state of the game via the video output on the screen. Thus, Dota 2 is a zero-sum Markov game played by two teams, with each agent having imperfect information of the game. For this challenging problem, in 2018, *OpenAI* has proposed the *OpenAI Five* AI system (OpenAI, 2018), which enjoys superhuman performances and has defeated human world champions in an e-sports game. The algorithmic framework integrates LSTM for learning good representations and proximal policy optimization (Schulman et al., 2017) with self-play for policy learning. Moreover, to balance between effective coordination and communication cost, instead of having explicit communication channels among the teams, OpenAI Five utilizes reward shaping by having a hyperparameter, named "team spirit", to balance the relative importance between each hero's individual reward function and the average of the team's reward function.

6 Conclusions and Future Directions

Multi-agent RL has long been an active and significant research area in reinforcement learning, in view of the ubiquity of sequential decision-making with multiple agents coupled in their actions and information. In stark contrast to its great empirical success, theoretical understanding of MARL algorithms is well recognized to be challenging and relatively lacking in the literature. Indeed, establishing an encompassing theory for MARL requires tools spanning dynamic programming, game theory, optimization theory, and statistics, which are non-trivial to unify and investigate within one context.

In this chapter, we have provided a selective overview of mostly recent MARL algorithms, backed by theoretical analysis, followed by several high-profile but challenging applications that have been addressed lately. Following the classical overview Busoniu et al. (2008), we have categorized the algorithms into three groups: those solving problems that are fully cooperative, fully competitive, and a mix of the two. Orthogonal to the existing reviews on MARL, this chapter has laid emphasis on several new angles and taxonomies of MARL theory, some of which have been drawn from our own research endeavors and interests. We note that our overview should not be viewed as a comprehensive one, but instead as a focused one dictated by our own interests and expertise, which should appeal to researchers of similar interests, and provide a stimulus for future research directions in this general topical area. Accordingly, we have identified the following paramount while open avenues for future research on MARL theory.

Partially observed settings: Partial observability of the system states and the actions of other agents is quintessential and inevitable in many practical MARL applications. In general, these settings can be modeled as a partially observed stochastic game (POSG), which includes the cooperative setting with a common reward function, i.e., the Dec-POMDP model, as a special case. Nevertheless, as pointed out in §4.1.3, even the cooperative task is NEXP-hard (Bernstein et al., 2002) and difficult to solve. In fact, the information state for optimal decision-making in POSGs can be very complicated and involve belief generation over the opponents' policies (Hansen et al., 2004), compared to that in POMDPs, which

requires belief on only states. This difficulty essentially stems from the heterogenous beliefs of agents resulting from their own observations obtained from the model, an inherent challenge of MARL mentioned in §3 due to various information structures. It might be possible to start by generalizing the centralized-learning-decentralized-execution scheme for solving Dec-POMDPs (Amato and Oliehoek, 2015; Dibangoye and Buffet, 2018) to solving POSGs.

Deep MARL theory: As mentioned in §3.3, using deep neural networks for function approximation can address the scalability issue in MARL. In fact, most of the recent empirical successes in MARL result from the use of DNNs (Heinrich and Silver, 2016; Lowe et al., 2017; Foerster et al., 2017; Gupta et al., 2017; Omidshafiei et al., 2017). Nonetheless, because of lack of theoretical backings, we have not included details of these algorithms in this chapter. Very recently, a few attempts have been made to understand the global convergence of several single-agent deep RL algorithms, such as neural TD learning (Cai et al., 2019) and neural policy optimization (Wang et al., 2019; Liu et al., 2019), when overparameterized neural networks (Arora et al., 2018; Li and Liang, 2018) are used. It is thus promising to extend these results to multi-agent settings, as initial steps toward theoretical understanding of deep MARL.

Model-based MARL: It may be slightly surprising that very few MARL algorithms in the literature are *model-based*, in the sense that the MARL model is first estimated, and then used as a nominal one to design algorithms. To the best of our knowledge, the only existing model-based MARL algorithms include the early one in Brafman and Tennenholtz (2000) that solves single-controller-stochastic games, a special zero-sum MG; and the later improved one in Brafman and Tennenholtz (2002), named R-MAX, for zero-sum MGs. These algorithms are also built upon the principle of optimism in the face of uncertainty (Auer and Ortner, 2007; Jaksch et al., 2010), as several aforementioned model-free ones. Considering recent progresses in model-based RL, especially its provable advantages over model-free ones in certain regimes (Tu and Recht, 2018; Sun et al., 2019), it is worth generalizing these results to MARL to improve its sample efficiency.

Convergence of policy gradient methods: As mentioned in §4.3, the convergence result of vanilla policy gradient method in general MARL is mostly negative, i.e., it may avoid even the local NE points in many cases. This is essentially related to the challenge of non-stationarity in MARL, see §3.2. Even though some remedies have been advocated (Balduzzi et al., 2018; Letcher et al., 2019; Chasnov et al., 2019; Fiez et al., 2019) to stabilizing the convergence in *general continuous* games, these assumptions are not easily verified/satisfied in MARL, e.g., even in the simplest LQ setting (Mazumdar et al., 2019), as they depend not only on the model, but also on the policy parameterization. Due to this subtlety, it may be interesting to explore the (global) convergence of policy-based methods for MARL, probably starting with the simple LQ setting, i.e., general-sum LQ games, in analogy to that for the zero-sum counterpart (Zhang et al., 2019; Bu et al., 2019). Such an exploration may also benefit from the recent advances of nonconvex-(non)concave optimization (Lin et al., 2018; Jin et al., 2019; Nouiehed et al., 2019).

MARL with robustness/safety concerns: Concerning the challenge of non-unique learn-

ing goals in MARL (see §3.1), we believe it is of merit to consider robustness and/or safety constraints in MARL. To the best of our knowledge, this is still a relatively uncharted territory. In fact, safe RL has been recognized as one of the most significant challenges in the single-agent setting (García and Fernández, 2015). With more than one agents that may have conflicted objectives, guaranteeing safety becomes more involved, as the safety requirement now concerns the coupling of all agents. One straightforward model is constrained multi-agent MDPs/Markov games, with the constraints characterizing the safety requirement. Learning with provably safety guarantees in this setting is non-trivial, but necessary for some safety-critical MARL applications as autonomous driving (Shalev-Shwartz et al., 2016) and robotics (Kober et al., 2013). In addition, it is also natural to think of robustness against adversarial agents, especially in the decentralized/distributed cooperative MARL settings as in Zhang et al. (2018); Chen et al. (2018); Wai et al. (2018), where the adversary may disturb the learning process in an anonymous way – a common scenario in distributed systems. Recent development of robust distributed supervised-learning against Byzantine adversaries (Chen et al., 2017; Yin et al., 2018) may be useful in this context.

References

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, **529** 484–489.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A. et al. (2017). Mastering the game of Go without human knowledge. *Nature*, **550** 354.

OpenAI (2018). Openai five. https://blog.openai.com/openai-five/.

Vinyals, O., Babuschkin, I., Chung, J., Mathieu, M., Jaderberg, M., Czarnecki, W. M., Dudzik, A., Huang, A., Georgiev, P., Powell, R., Ewalds, T., Horgan, D., Kroiss, M., Danihelka, I., Agapiou, J., Oh, J., Dalibard, V., Choi, D., Sifre, L., Sulsky, Y., Vezhnevets, S., Molloy, J., Cai, T., Budden, D., Paine, T., Gulcehre, C., Wang, Z., Pfaff, T., Pohlen, T., Wu, Y., Yogatama, D., Cohen, J., McKinney, K., Smith, O., Schaul, T., Lillicrap, T., Apps, C., Kavukcuoglu, K., Hassabis, D. and Silver, D. (2019). AlphaStar: Mastering the Real-Time Strategy Game StarCraft II. https://deepmind.com/blog/alphastar-mastering-real-time-strategy-game-starcraft-ii/.

Kober, J., Bagnell, J. A. and Peters, J. (2013). Reinforcement learning in robotics: A survey. *International Journal of Robotics Research*, **32** 1238–1274.

Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D. and Wierstra, D. (2016). Continuous control with deep reinforcement learning. In *International Conference on Learning Representations*.

Brown, N. and Sandholm, T. (2017). Libratus: the superhuman ai for no-limit poker. In *International Joint Conference on Artificial Intelligence*.

Brown, N. and Sandholm, T. (2019). Superhuman AI for multiplayer poker. *Science*, **365** 885–890.

Shalev-Shwartz, S., Shammah, S. and Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. *arXiv preprint arXiv:1610.03295*.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, **518** 529–533.

Busoniu, L., Babuska, R., De Schutter, B. et al. (2008). A comprehensive survey of multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part C*, **38** 156–172.

Adler, J. L. and Blue, V. J. (2002). A cooperative multi-agent transportation management and route guidance system. *Transportation Research Part C: Emerging Technologies*, **10** 433–454.

Wang, S., Wan, J., Zhang, D., Li, D. and Zhang, C. (2016). Towards smart factory for industry 4.0: A self-organized multi-agent system with big data based feedback and coordination. *Computer Networks*, **101** 158–168.

O, J., Lee, J. W. and Zhang, B.-T. (2002). Stock trading system using reinforcement learning with cooperative agents. In *International Conference on Machine Learning*.

Lee, J. W., Park, J., Jangmin, O., Lee, J. and Hong, E. (2007). A multiagent approach to q-learning for daily stock trading. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, **37** 864–877.

Cortes, J., Martinez, S., Karatas, T. and Bullo, F. (2004). Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, **20** 243–255.

Choi, J., Oh, S. and Horowitz, R. (2009). Distributed learning and cooperative control for multi-agent systems. *Automatica*, **45** 2802–2814.

Castelfranchi, C. (2001). The theory of social functions: Challenges for computational social science and multi-agent learning. *Cognitive Systems Research*, **2** 5–38.

Leibo, J. Z., Zambaldi, V., Lanctot, M., Marecki, J. and Graepel, T. (2017). Multi-agent reinforcement learning in sequential social dilemmas. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Hernandez-Leal, P., Kartal, B. and Taylor, M. E. (2018). A survey and critique of multiagent deep reinforcement learning. *arXiv preprint arXiv:1810.05587*.

Foerster, J., Assael, Y. M., de Freitas, N. and Whiteson, S. (2016). Learning to communicate with deep multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems*.

Zazo, S., Macua, S. V., Sánchez-Fernández, M. and Zazo, J. (2016). Dynamic potential games with constraints: Fundamentals and applications in communications. *IEEE Transactions on Signal Processing*, **64** 3806–3821.

Zhang, K., Yang, Z., Liu, H., Zhang, T. and Başar, T. (2018). Fully decentralized multiagent reinforcement learning with networked agents. In *International Conference on Machine Learning*.

Subramanian, J. and Mahajan, A. (2019). Reinforcement learning in stationary mean-field games. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Heinrich, J. and Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.

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

Foerster, J., Farquhar, G., Afouras, T., Nardelli, N. and Whiteson, S. (2017). Counterfactual multi-agent policy gradients. *arXiv preprint arXiv:1705.08926*.

Gupta, J. K., Egorov, M. and Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Omidshafiei, S., Pazis, J., Amato, C., How, J. P. and Vian, J. (2017). Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*.

Kawamura, K., Mizukami, N. and Tsuruoka, Y. (2017). Neural fictitious self-play in imperfect information games with many players. In *Workshop on Computer Games*.

Zhang, L., Wang, W., Li, S. and Pan, G. (2019). Monte Carlo neural fictitious self-play: Approach to approximate Nash equilibrium of imperfect-information games. *arXiv preprint arXiv:1903.09569*.

Mazumdar, E. and Ratliff, L. J. (2018). On the convergence of gradient-based learning in continuous games. *arXiv preprint arXiv:1804.05464*.

Jin, C., Netrapalli, P. and Jordan, M. I. (2019). Minmax optimization: Stable limit points of gradient descent ascent are locally optimal. *arXiv preprint arXiv:1902.00618*.

Zhang, K., Yang, Z. and Başar, T. (2019). Policy optimization provably converges to Nash equilibria in zero-sum linear quadratic games. In *Advances in Neural Information Processing Systems*.

Sidford, A., Wang, M., Yang, L. F. and Ye, Y. (2019). Solving discounted stochastic two-player games with near-optimal time and sample complexity. *arXiv preprint arXiv:1908.11071*.

Oliehoek, F. A. and Amato, C. (2016). *A Concise Introduction to Decentralized POMDPs*, vol. 1. Springer.

Arslan, G. and Yüksel, S. (2017). Decentralized Q-learning for stochastic teams and games. *IEEE Transactions on Automatic Control*, **62** 1545–1558.

Yongacoglu, B., Arslan, G. and Yüksel, S. (2019). Learning team-optimality for decentralized stochastic control and dynamic games. *arXiv preprint arXiv:1903.05812*.

Zhang, K., Miehling, E. and Başar, T. (2019). Online planning for decentralized stochastic control with partial history sharing. In *IEEE American Control Conference*.

Hernandez-Leal, P., Kaisers, M., Baarslag, T. and de Cote, E. M. (2017). A survey of learning in multiagent environments: Dealing with non-stationarity. *arXiv preprint arXiv:1707.09183*.

Nguyen, T. T., Nguyen, N. D. and Nahavandi, S. (2018). Deep reinforcement learning for multi-agent systems: A review of challenges, solutions and applications. *arXiv preprint arXiv:1812.11794*.

Oroojlooy Jadid, A. and Hajinezhad, D. (2019). A review of cooperative multi-agent deep reinforcement learning. *arXiv preprint arXiv:1908.03963*.

Zhang, K., Yang, Z. and Başar, T. (2018a). Networked multi-agent reinforcement learning in continuous spaces. In *IEEE Conference on Decision and Control*.

Zhang, K., Yang, Z., Liu, H., Zhang, T. and Başar, T. (2018b). Finite-sample analyses for fully decentralized multi-agent reinforcement learning. *arXiv preprint arXiv:1812.02783*.

Monahan, G. E. (1982). State of the art–A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, **28** 1–16.

Cassandra, A. R. (1998). *Exact and approximate algorithms for partially observable Markov decision processes*. Brown University.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control*, vol. 1. Athena Scientific Belmont, MA.

Watkins, C. J. and Dayan, P. (1992). Q-learning. *Machine Learning*, **8** 279–292.

Szepesvári, C. and Littman, M. L. (1999). A unified analysis of value-function-based reinforcement-learning algorithms. *Neural Computation*, **11** 2017–2060.

Singh, S., Jaakkola, T., Littman, M. L. and Szepesvári, C. (2000). Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine Learning*, **38** 287–308.

Chang, H. S., Fu, M. C., Hu, J. and Marcus, S. I. (2005). An adaptive sampling algorithm for solving Markov decision processes. *Operations Research*, **53** 126–139.

Kocsis, L. and Szepesvári, C. (2006). Bandit based Monte-Carlo planning. In *European Conference on Machine Learning*. Springer.

Coulom, R. (2006). Efficient selectivity and backup operators in Monte-Carlo tree search. In *International Conference on Computers and Games*.

Agrawal, R. (1995). Sample mean based index policies by $O(logn)$ regret for the multi-armed bandit problem. *Advances in Applied Probability*, **27** 1054–1078.

Auer, P., Cesa-Bianchi, N. and Fischer, P. (2002). Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, **47** 235–256.

Jiang, D., Ekwedike, E. and Liu, H. (2018). Feedback-based tree search for reinforcement learning. In *International Conference on Machine Learning*.

Shah, D., Xie, Q. and Xu, Z. (2019). On reinforcement learning using Monte-Carlo tree search with supervised learning: Non-asymptotic analysis. *arXiv preprint arXiv:1902.05213*.

Tesauro, G. (1995). Temporal difference learning and TD-Gammon. *Communications of the ACM*, **38** 58–68.

Tsitsiklis, J. N. and Van Roy, B. (1997). Analysis of temporal-diffference learning with function approximation. In *Advances in Neural Information Processing Systems*.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press.

Sutton, R. S., Szepesvári, C. and Maei, H. R. (2008). A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. *Advances in Neural Information Processing Systems*, **21** 1609–1616.

Sutton, R. S., Maei, H. R., Precup, D., Bhatnagar, S., Silver, D., Szepesvári, C. and Wiewiora, E. (2009). Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *International Conference on Machine Learning*.

Liu, B., Liu, J., Ghavamzadeh, M., Mahadevan, S. and Petrik, M. (2015). Finite-sample analysis of proximal gradient TD algorithms. In *Conference on Uncertainty in Artificial Intelligence*.

Bhatnagar, S., Precup, D., Silver, D., Sutton, R. S., Maei, H. R. and Szepesvári, C. (2009). Convergent temporal-difference learning with arbitrary smooth function approximation. In *Advances in Neural Information Processing Systems*.

Dann, C., Neumann, G., Peters, J. et al. (2014). Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Research*, **15** 809–883.

Sutton, R. S., McAllester, D. A., Singh, S. P. and Mansour, Y. (2000). Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, **8** 229–256.

Baxter, J. and Bartlett, P. L. (2001). Infinite-horizon policy-gradient estimation. *Journal of Artificial Intelligence Research*, **15** 319–350.

Konda, V. R. and Tsitsiklis, J. N. (2000). Actor-critic algorithms. In *Advances in Neural Information Processing Systems*.

Bhatnagar, S., Sutton, R., Ghavamzadeh, M. and Lee, M. (2009). Natural actor-critic algorithms. *Automatica*, **45** 2471–2482.

Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D. and Riedmiller, M. (2014). Deterministic policy gradient algorithms. In *International Conference on Machine Learning*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A. and Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P. (2015). Trust region policy optimization. In *International Conference on Machine Learning*.

Haarnoja, T., Zhou, A., Abbeel, P. and Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*.

Yang, Z., Zhang, K., Hong, M. and Başar, T. (2018). A finite sample analysis of the actor-critic algorithm. In *IEEE Conference on Decision and Control*.

Zhang, K., Koppel, A., Zhu, H. and Başar, T. (2019). Global convergence of policy gradient methods to (almost) locally optimal policies. *arXiv preprint arXiv:1906.08383*.

Agarwal, A., Kakade, S. M., Lee, J. D. and Mahajan, G. (2019). Optimality and approximation with policy gradient methods in Markov decision processes. *arXiv preprint arXiv:1908.00261*.

Liu, B., Cai, Q., Yang, Z. and Wang, Z. (2019). Neural proximal/trust region policy optimization attains globally optimal policy. *arXiv preprint arXiv:1906.10306*.

Wang, L., Cai, Q., Yang, Z. and Wang, Z. (2019). Neural policy gradient methods: Global optimality and rates of convergence. *arXiv preprint arXiv:1909.01150*.

Shapley, L. S. (1953). Stochastic games. *Proceedings of the National Academy of Sciences*, **39** 1095–1100.

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*.

Başar, T. and Olsder, G. J. (1999). *Dynamic Noncooperative Game Theory*, vol. 23. SIAM.

Filar, J. and Vrieze, K. (2012). *Competitive Markov Decision Processes*. Springer Science & Business Media.

Boutilier, C. (1996). Planning, learning and coordination in multi-agent decision processes. In *Conference on Theoretical Aspects of Rationality and Knowledge*.

Lauer, M. and Riedmiller, M. (2000). An algorithm for distributed reinforcement learning in cooperative multi-agent systems. In *International Conference on Machine Learning*.

Yoshikawa, T. (1978). Decomposition of dynamic team decision problems. *IEEE Transactions on Automatic Control*, **23** 627–632.

Ho, Y.-C. (1980). Team decision theory and information structures. *Proceedings of the IEEE*, **68** 644–654.

Wang, X. and Sandholm, T. (2003). Reinforcement learning to play an optimal Nash equilibrium in team Markov games. In *Advances in Neural Information Processing Systems*.

Mahajan, A. (2008). *Sequential Decomposition of Sequential Dynamic Teams: Applications to Real-Time Communication and Networked Control Systems*. Ph.D. thesis, University of Michigan.

González-Sánchez, D. and Hernández-Lerma, O. (2013). *Discrete-Time Stochastic Control and Dynamic Potential Games: The Euler-Equation Approach*. Springer Science & Business Media.

Valcarcel Macua, S., Zazo, J. and Zazo, S. (2018). Learning parametric closed-loop policies for Markov potential games. In *International Conference on Learning Representations*.

Kar, S., Moura, J. M. and Poor, H. V. (2013). QD-learning: A collaborative distributed strategy for multi-agent reinforcement learning through consensus + innovations. *IEEE Transactions on Signal Processing*, **61** 1848–1862.

Doan, T., Maguluri, S. and Romberg, J. (2019). Finite-time analysis of distributed TD (0) with linear function approximation on multi-agent reinforcement learning. In *International Conference on Machine Learning*.

Wai, H.-T., Yang, Z., Wang, Z. and Hong, M. (2018). Multi-agent reinforcement learning via double averaging primal-dual optimization. In *Advances in Neural Information Processing Systems*.

OpenAI (2017). Openai dota 2 1v1 bot. https://openai.com/the-international/.

Jacobson, D. (1973). Optimal stochastic linear systems with exponential performance criteria and their relation to deterministic differential games. *IEEE Transactions on Automatic Control*, **18** 124–131.

Başar, T. and Bernhard, P. (1995). *H_∞ Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhäuser, Boston.

Zhang, K., Hu, B. and Başar, T. (2019). Policy optimization for $\mathcal{H}_2$ linear control with $\mathcal{H}_\infty$ robustness guarantee: Implicit regularization and global convergence. *arXiv preprint arXiv:1910.09496*.

Hu, J. and Wellman, M. P. (2003). Nash Q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, **4** 1039–1069.

Littman, M. L. (2001). Friend-or-Foe Q-learning in general-sum games. In *International Conference on Machine Learning*.

Lagoudakis, M. G. and Parr, R. (2003). Learning in zero-sum team Markov games using factored value functions. In *Advances in Neural Information Processing Systems*.

Bernstein, D. S., Givan, R., Immerman, N. and Zilberstein, S. (2002). The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, **27** 819–840.

Osborne, M. J. and Rubinstein, A. (1994). *A Course in Game Theory*. MIT Press.

Shoham, Y. and Leyton-Brown, K. (2008). *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge University Press.

Koller, D. and Megiddo, N. (1992). The complexity of two-person zero-sum games in extensive form. *Games and Economic Behavior*, **4** 528–552.

Kuhn, H. (1953). Extensive games and the problem op information. *Contributions to the Theory of Games*, **2** 193–216.

Zinkevich, M., Johanson, M., Bowling, M. and Piccione, C. (2008). Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*.

Heinrich, J., Lanctot, M. and Silver, D. (2015). Fictitious self-play in extensive-form games. In *International Conference on Machine Learning*.

Srinivasan, S., Lanctot, M., Zambaldi, V., Pérolat, J., Tuyls, K., Munos, R. and Bowling, M. (2018). Actor-critic policy optimization in partially observable multiagent environments. In *Advances in Neural Information Processing Systems*.

Omidshafiei, S., Hennes, D., Morrill, D., Munos, R., Perolat, J., Lanctot, M., Gruslys, A., Lespiau, J.-B. and Tuyls, K. (2019). Neural replicator dynamics. *arXiv preprint arXiv:1906.00190*.

Rubin, J. and Watson, I. (2011). Computer poker: A review. *Artificial Intelligence*, **175** 958–987.

Lanctot, M., Lockhart, E., Lespiau, J.-B., Zambaldi, V., Upadhyay, S., Pérolat, J., Srinivasan, S., Timbers, F., Tuyls, K., Omidshafiei, S. et al. (2019). Openspiel: A framework for reinforcement learning in games. *arXiv preprint arXiv:1908.09453*.

Claus, C. and Boutilier, C. (1998). The dynamics of reinforcement learning in cooperative multiagent systems. *AAAI Conference on Artificial Intelligence*, **1998** 2.

Bowling, M. and Veloso, M. (2001). Rational and convergent learning in stochastic games. In *International Joint Conference on Artificial Intelligence*, vol. 17.

Kapetanakis, S. and Kudenko, D. (2002). Reinforcement learning of coordination in cooperative multi-agent systems. *AAAI Conference on Artificial Intelligence*, **2002** 326–331.

Conitzer, V. and Sandholm, T. (2007). Awesome: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, **67** 23–43.

Hansen, E. A., Bernstein, D. S. and Zilberstein, S. (2004). Dynamic programming for partially observable stochastic games. In *AAAI Conference on Artificial Intelligence*.

Amato, C., Chowdhary, G., Geramifard, A., Üre, N. K. and Kochenderfer, M. J. (2013). Decentralized control of partially observable markov decision processes. In *IEEE Conference on Decision and Control*.

Amato, C. and Oliehoek, F. A. (2015). Scalable planning and learning for multiagent POMDPs. In *AAAI Conference on Artificial Intelligence*.

Shoham, Y., Powers, R. and Grenager, T. (2003). Multi-agent reinforcement learning: A critical survey. *Technical Report*.

Zinkevich, M., Greenwald, A. and Littman, M. L. (2006). Cyclic equilibria in Markov games. In *Advances in Neural Information Processing Systems*.

Bowling, M. and Veloso, M. (2002). Multiagent learning using a variable learning rate. *Artificial Intelligence*, **136** 215–250.

Bowling, M. (2005). Convergence and no-regret in multiagent learning. In *Advances in Neural Information Processing Systems*.

Hart, S. and Mas-Colell, A. (2001). A reinforcement procedure leading to correlated equilibrium. In *Economics Essays*. Springer, 181–200.

Kasai, T., Tenmoto, H. and Kamiya, A. (2008). Learning of communication codes in multi-agent reinforcement learning problem. In *IEEE Conference on Soft Computing in Industrial Applications*.

Kim, D., Moon, S., Hostallero, D., Kang, W. J., Lee, T., Son, K. and Yi, Y. (2019). Learning to schedule communication in multi-agent reinforcement learning. In *International Conference on Learning Representations*.

Chen, T., Zhang, K., Giannakis, G. B. and Başar, T. (2018). Communication-efficient distributed reinforcement learning. *arXiv preprint arXiv:1812.03239*.

Lin, Y., Zhang, K., Yang, Z., Wang, Z., Başar, T., Sandhu, R. and Liu, J. (2019). A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning. In *IEEE Conference on Decision and Control*.

Ren, J. and Haupt, J. (2019). A communication efficient hierarchical distributed optimization algorithm for multi-agent reinforcement learning. In *Real-world Sequential Decision Making Workshop at International Conference on Machine Learning*.

Kim, W., Cho, M. and Sung, Y. (2019). Message-dropout: An efficient training method for multi-agent deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*.

He, H., Boyd-Graber, J., Kwok, K. and Daumé III, H. (2016). Opponent modeling in deep reinforcement learning. In *International Conference on Machine Learning*.

Grover, A., Al-Shedivat, M., Gupta, J., Burda, Y. and Edwards, H. (2018). Learning policy representations in multiagent systems. In *International Conference on Machine Learning*.

Gao, C., Mueller, M. and Hayward, R. (2018). Adversarial policy gradient for alternating Markov games. In *Workshop at International Conference on Learning Representations*.

Li, S., Wu, Y., Cui, X., Dong, H., Fang, F. and Russell, S. (2019). Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient. In *AAAI Conference on Artificial Intelligence*.

Zhang, X., Zhang, K., Miehling, E. and Basar, T. (2019). Non-cooperative inverse reinforcement learning. In *Advances in Neural Information Processing Systems*.

Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. In *International Conference on Machine Learning*.

Matignon, L., Laurent, G. J. and Le Fort-Piat, N. (2012). Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems. *The Knowledge Engineering Review*, **27** 1–31.

Foerster, J., Nardelli, N., Farquhar, G., Torr, P., Kohli, P., Whiteson, S. et al. (2017). Stabilising experience replay for deep multi-agent reinforcement learning. In *International Conference of Machine Learning*.

Tuyls, K. and Weiss, G. (2012). Multiagent learning: Basics, challenges, and prospects. *AI Magazine*, **33** 41–41.

Guestrin, C., Lagoudakis, M. and Parr, R. (2002a). Coordinated reinforcement learning. In *International Conference on Machine Learning*.

Guestrin, C., Koller, D. and Parr, R. (2002b). Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*.

Kok, J. R. and Vlassis, N. (2004). Sparse cooperative Q-learning. In *International Conference on Machine learning*.

Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W. M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J. Z., Tuyls, K. et al. (2018). Value-decomposition networks for cooperative multi-agent learning based on team reward. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Rashid, T., Samvelyan, M., De Witt, C. S., Farquhar, G., Foerster, J. and Whiteson, S. (2018). QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine learning*.

Qu, G. and Li, N. (2019). Exploiting fast decaying and locality in multi-agent MDP with tree dependence structure. In *IEEE Conference on Decision and Control*.

Mahajan, A. (2013). Optimal decentralized control of coupled subsystems with control sharing. *IEEE Transactions on Automatic Control*, **58** 2377–2382.

Oliehoek, F. A. and Amato, C. (2014). Dec-POMDPs as non-observable MDPs. *IAS Technical Report*.

Foerster, J. N., Farquhar, G., Afouras, T., Nardelli, N. and Whiteson, S. (2018). Counterfactual multi-agent policy gradients. In *AAAI Conference on Artificial Intelligence*.

Dibangoye, J. and Buffet, O. (2018). Learning to act in decentralized partially observable MDPs. In *International Conference on Machine Learning*.

Kraemer, L. and Banerjee, B. (2016). Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing*, **190** 82–94.

Macua, S. V., Chen, J., Zazo, S. and Sayed, A. H. (2015). Distributed policy evaluation under multiple behavior strategies. *IEEE Transactions on Automatic Control*, **60** 1260–1274.

Macua, S. V., Tukiainen, A., Hernández, D. G.-O., Baldazo, D., de Cote, E. M. and Zazo, S. (2017). Diff-dac: Distributed actor-critic for average multitask deep reinforcement learning. *arXiv preprint arXiv:1710.10363*.

Lee, D., Yoon, H. and Hovakimyan, N. (2018). Primal-dual algorithm for distributed reinforcement learning: distributed GTD. In *IEEE Conference on Decision and Control*.

Suttle, W., Yang, Z., Zhang, K., Wang, Z., Başar, T. and Liu, J. (2019). A multi-agent off-policy actor-critic algorithm for distributed reinforcement learning. *arXiv preprint arXiv:1903.06372*.

Doan, T. T., Maguluri, S. T. and Romberg, J. (2019). Finite-time performance of distributed temporal difference learning with linear function approximation. *arXiv preprint arXiv:1907.12530*.

Littman, M. L. (2001). Value-function reinforcement learning in Markov games. *Cognitive Systems Research*, **2** 55–66.

Young, H. P. (1993). The evolution of conventions. *Econometrica: Journal of the Econometric Society* 57–84.

Son, K., Kim, D., Kang, W. J., Hostallero, D. E. and Yi, Y. (2019). QTRAN: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*.

Perolat, J., Piot, B. and Pietquin, O. (2018). Actor-critic fictitious play in simultaneous move multistage games. In *International Conference on Artificial Intelligence and Statistics*.

Monderer, D. and Shapley, L. S. (1996). Potential games. *Games and Economic Behavior*, **14** 124–143.

Başar, T. and Zaccour, G. (2018). *Handbook of Dynamic Game Theory*. Springer.

Huang, M., Caines, P. E. and Malhamé, R. P. (2003). Individual and mass behaviour in large population stochastic wireless power control problems: Centralized and Nash equilibrium solutions. In *IEEE Conference on Decision and Control*.

Huang, M., Malhamé, R. P., Caines, P. E. et al. (2006). Large population stochastic dynamic games: Closed-loop Mckean-Vlasov systems and the Nash certainty equivalence principle. *Communications in Information & Systems*, **6** 221–252.

Lasry, J.-M. and Lions, P.-L. (2007). Mean field games. *Japanese Journal of Mathematics*, **2** 229–260.

Bensoussan, A., Frehse, J., Yam, P. et al. (2013). *Mean Field Games and Mean Field Type Control Theory*, vol. 101. Springer.

Tembine, H., Zhu, Q. and Başar, T. (2013). Risk-sensitive mean-field games. *IEEE Transactions on Automatic Control*, **59** 835–850.

Arabneydi, J. and Mahajan, A. (2014). Team optimal control of coupled subsystems with mean-field sharing. In *IEEE Conference on Decision and Control*.

Arabneydi, J. (2017). *New Concepts in Team Theory: Mean Field Teams and Reinforcement Learning*. Ph.D. thesis, McGill University.

Yang, Y., Luo, R., Li, M., Zhou, M., Zhang, W. and Wang, J. (2018). Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*.

Witsenhausen, H. S. (1971). Separation of estimation and control for discrete time systems. *Proceedings of the IEEE*, **59** 1557–1566.

Yüksel, S. and Başar, T. (2013). *Stochastic Networked Control Systems: Stabilization and Optimization Under Information Constraints*. Springer Science & Business Media.

Subramanian, J., Seraj, R. and Mahajan, A. (2018). Reinforcement learning for mean-field teams. In *Workshop on Adaptive and Learning Agents at International Conference on Autonomous Agents and Multi-Agent Systems*.

Arabneydi, J. and Mahajan, A. (2016). Linear quadratic mean field teams: Optimal and approximately optimal decentralized solutions. *arXiv preprint arXiv:1609.00056*.

Carmona, R., Laurière, M. and Tan, Z. (2019a). Linear-quadratic mean-field reinforcement learning: Convergence of policy gradient methods. *arXiv preprint arXiv:1910.04295*.

Carmona, R., Laurière, M. and Tan, Z. (2019b). Model-free mean-field reinforcement learning: Mean-field mdp and mean-field q-learning. *arXiv preprint arXiv:1910.12802*.

Rabbat, M. and Nowak, R. (2004). Distributed optimization in sensor networks. In *International Symposium on Information Processing in Sensor Networks*.

Dall'Anese, E., Zhu, H. and Giannakis, G. B. (2013). Distributed optimal power flow for smart microgrids. *IEEE Transactions on Smart Grid*, **4** 1464–1475.

Zhang, K., Shi, W., Zhu, H., Dall'Anese, E. and Başar, T. (2018a). Dynamic power distribution system management with a locally connected communication network. *IEEE Journal of Selected Topics in Signal Processing*, **12** 673–687.

Zhang, K., Lu, L., Lei, C., Zhu, H. and Ouyang, Y. (2018b). Dynamic operations and pricing of electric unmanned aerial vehicle systems and power networks. *Transportation Research Part C: Emerging Technologies*, **92** 472–485.

Corke, P., Peterson, R. and Rus, D. (2005). Networked robots: Flying robot navigation using a sensor net. *Robotics Research* 234–243.

Zhang, K., Liu, Y., Liu, J., Liu, M. and Başar, T. (2019). Distributed learning of average belief over networks using sequential observations. *Automatica*.

Nedic, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, **54** 48–61.

Agarwal, A. and Duchi, J. C. (2011). Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems*.

Jakovetic, D., Xavier, J. and Moura, J. M. (2011). Cooperative convex optimization in networked systems: Augmented lagrangian algorithms with directed gossip communication. *IEEE Transactions on Signal Processing*, **59** 3889–3902.

Tu, S.-Y. and Sayed, A. H. (2012). Diffusion strategies outperform consensus strategies for distributed estimation over adaptive networks. *IEEE Transactions on Signal Processing*, **60** 6217–6234.

Varshavskaya, P., Kaelbling, L. P. and Rus, D. (2009). Efficient distributed reinforcement learning through agreement. In *Distributed Autonomous Robotic Systems*. 367–378.

Ciosek, K. and Whiteson, S. (2018). Expected policy gradients for reinforcement learning. *arXiv preprint arXiv:1801.03326*.

Sutton, R. S., Mahmood, A. R. and White, M. (2016). An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Research*, **17** 2603–2631.

Yu, H. (2015). On convergence of emphatic temporal-difference learning. In *Conference on Learning Theory*.

Zhang, Y. and Zavlanos, M. M. (2019). Distributed off-policy actor-critic reinforcement learning with policy consensus. *arXiv preprint arXiv:1903.09255*.

Pennesi, P. and Paschalidis, I. C. (2010). A distributed actor-critic algorithm and applications to mobile sensor network coordination problems. *IEEE Transactions on Automatic Control*, **55** 492–497.

Lange, S., Gabel, T. and Riedmiller, M. (2012). Batch reinforcement learning. In *Reinforcement Learning*. Springer, 45–73.

Riedmiller, M. (2005). Neural fitted Q iteration–first experiences with a data efficient neural reinforcement learning method. In *European Conference on Machine Learning*.

Antos, A., Szepesvári, C. and Munos, R. (2008). Fitted Q-iteration in continuous action-space MDPs. In *Advances in Neural Information Processing Systems*.

Hong, M. and Chang, T.-H. (2017). Stochastic proximal gradient consensus over random networks. *IEEE Transactions on Signal Processing*, **65** 2933–2948.

Nedic, A., Olshevsky, A. and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, **27** 2597–2633.

Munos, R. (2007). Performance bounds in ℓ_p-norm for approximate value iteration. *SIAM Journal on Control and Optimization*, **46** 541–561.

Antos, A., Szepesvári, C. and Munos, R. (2008). Learning near-optimal policies with Bellman-residual minimization based fitted policy iteration and a single sample path. *Machine Learning*, **71** 89–129.

Munos, R. and Szepesvári, C. (2008). Finite-time bounds for fitted value iteration. *Journal of Machine Learning Research*, **9** 815–857.

Farahmand, A.-m., Szepesvári, C. and Munos, R. (2010). Error propagation for approximate policy and value iteration. In *Advances in Neural Information Processing Systems*.

Cassano, L., Yuan, K. and Sayed, A. H. (2018). Multi-agent fully decentralized off-policy learning with linear convergence rates. *arXiv preprint arXiv:1810.07792*.

Qu, G. and Li, N. (2017). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, **5** 1245–1260.

Schmidt, M., Le Roux, N. and Bach, F. (2017). Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, **162** 83–112.

Ying, B., Yuan, K. and Sayed, A. H. (2018). Convergence of variance-reduced learning under random reshuffling. In *IEEE International Conference on Acoustics, Speech and Signal Processing*.

Singh, S. P. and Sutton, R. S. (1996). Reinforcement learning with replacing eligibility traces. *Machine Learning*, **22** 123–158.

Bhandari, J., Russo, D. and Singal, R. (2018). A finite time analysis of temporal difference learning with linear function approximation. In *Conference On Learning Theory*.

Srikant, R. and Ying, L. (2019). Finite-time error bounds for linear stochastic approximation and TD learning. In *Conference on Learning Theory*.

Stanković, M. S. and Stanković, S. S. (2016). Multi-agent temporal-difference learning with linear function approximation: Weak convergence under time-varying network topologies. In *IEEE American Control Conference*.

Stanković, M. S., Ilić, N. and Stanković, S. S. (2016). Distributed stochastic approximation: Weak convergence and network design. *IEEE Transactions on Automatic Control*, **61** 4069–4074.

Zhang, H., Jiang, H., Luo, Y. and Xiao, G. (2016). Data-driven optimal consensus control for discrete-time multi-agent systems with unknown dynamics using reinforcement learning method. *IEEE Transactions on Industrial Electronics*, **64** 4091–4100.

Zhang, Q., Zhao, D. and Lewis, F. L. (2018). Model-free reinforcement learning for fully cooperative multi-agent graphical games. In *International Joint Conference on Neural Networks*.

Bernstein, D. S., Amato, C., Hansen, E. A. and Zilberstein, S. (2009). Policy iteration for decentralized control of Markov decision processes. *Journal of Artificial Intelligence Research*, **34** 89–132.

Amato, C., Bernstein, D. S. and Zilberstein, S. (2010). Optimizing fixed-size stochastic controllers for POMDPs and decentralized POMDPs. *Autonomous Agents and Multi-Agent Systems*, **21** 293–320.

Liu, M., Amato, C., Liao, X., Carin, L. and How, J. P. (2015). Stick-breaking policy learning in Dec-POMDPs. In *International Joint Conference on Artificial Intelligence*.

Dibangoye, J. S., Amato, C., Buffet, O. and Charpillet, F. (2016). Optimally solving Dec-POMDPs as continuous-state MDPs. *Journal of Artificial Intelligence Research*, **55** 443–497.

Wu, F., Zilberstein, S. and Chen, X. (2010). Rollout sampling policy iteration for decentralized POMDPs. In *Conference on Uncertainty in Artificial Intelligence*.

Wu, F., Zilberstein, S. and Jennings, N. R. (2013). Monte-Carlo expectation maximization for decentralized POMDPs. In *International Joint Conference on Artificial Intelligence*.

Best, G., Cliff, O. M., Patten, T., Mettu, R. R. and Fitch, R. (2018). Dec-MCTS: Decentralized planning for multi-robot active perception. *International Journal of Robotics Research* 1–22.

Amato, C. and Zilberstein, S. (2009). Achieving goals in decentralized POMDPs. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Banerjee, B., Lyle, J., Kraemer, L. and Yellamraju, R. (2012). Sample bounded distributed reinforcement learning for decentralized POMDPs. In *AAAI Conference on Artificial Intelligence*.

Nayyar, A., Mahajan, A. and Teneketzis, D. (2013). Decentralized stochastic control with partial history sharing: A common information approach. *IEEE Transactions on Automatic Control*, **58** 1644–1658.

Arabneydi, J. and Mahajan, A. (2015). Reinforcement learning in decentralized stochastic control systems with partial history sharing. In *IEEE American Control Conference*.

Papadimitriou, C. H. (1992). On inefficient proofs of existence and complexity classes. In *Annals of Discrete Mathematics*, vol. 51. Elsevier, 245–250.

Daskalakis, C., Goldberg, P. W. and Papadimitriou, C. H. (2009). The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, **39** 195–259.

Von Neumann, J., Morgenstern, O. and Kuhn, H. W. (2007). *Theory of Games and Economic Behavior (commemorative edition)*. Princeton University Press.

Vanderbei, R. J. et al. (2015). *Linear Programming*. Springer.

Hoffman, A. J. and Karp, R. M. (1966). On nonterminating stochastic games. *Management Science*, **12** 359–370.

Van Der Wal, J. (1978). Discounted markov games: Generalized policy iteration method. *Journal of Optimization Theory and Applications*, **25** 125–138.

Rao, S. S., Chandrasekaran, R. and Nair, K. (1973). Algorithms for discounted stochastic games. *Journal of Optimization Theory and Applications*, **11** 627–637.

Patek, S. D. (1997). *Stochastic and Shortest Path Games: Theory and Algorithms*. Ph.D. thesis, Massachusetts Institute of Technology.

Hansen, T. D., Miltersen, P. B. and Zwick, U. (2013). Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM*, **60** 1.

Lagoudakis, M. G. and Parr, R. (2002). Value function approximation in zero-sum Markov games. In *Conference on Uncertainty in Artificial Intelligence*.

Zou, S., Xu, T. and Liang, Y. (2019). Finite-sample analysis for SARSA with linear function approximation. *arXiv preprint arXiv:1902.02234*.

Sutton, R. S. and Barto, A. G. (1987). A temporal-difference model of classical conditioning. In *Proceedings of the Annual Conference of the Cognitive Science Society*.

Al-Tamimi, A., Abu-Khalaf, M. and Lewis, F. L. (2007a). Adaptive critic designs for discrete-time zero-sum games with application to $\mathcal{H}_\infty$ control. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, **37** 240–247.

Al-Tamimi, A., Lewis, F. L. and Abu-Khalaf, M. (2007b). Model-free Q-learning designs for linear discrete-time zero-sum games with application to $\mathcal{H}_\infty$ control. *Automatica*, **43** 473–481.

Farahmand, A.-m., Ghavamzadeh, M., Szepesvári, C. and Mannor, S. (2016). Regularized policy iteration with nonparametric function spaces. *Journal of Machine Learning Research*, **17** 4809–4874.

Yang, Z., Xie, Y. and Wang, Z. (2019). A theoretical analysis of deep Q-learning. *arXiv preprint arXiv:1901.00137*.

Jia, Z., Yang, L. F. and Wang, M. (2019). Feature-based Q-learning for two-player stochastic games. *arXiv preprint arXiv:1906.00423*.

Sidford, A., Wang, M., Wu, X., Yang, L. and Ye, Y. (2018). Near-optimal time and sample complexities for solving Markov decision processes with a generative model. In *Advances in Neural Information Processing Systems*.

Wei, C.-Y., Hong, Y.-T. and Lu, C.-J. (2017). Online reinforcement learning in stochastic games. In *Advances in Neural Information Processing Systems*.

Auer, P. and Ortner, R. (2007). Logarithmic online regret bounds for undiscounted reinforcement learning. In *Advances in Neural Information Processing Systems*.

Jaksch, T., Ortner, R. and Auer, P. (2010). Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, **11** 1563–1600.

Koller, D., Megiddo, N. and von Stengel, B. (1994). Fast algorithms for finding randomized strategies in game trees. *Computing*, **750** 759.

Von Stengel, B. (1996). Efficient computation of behavior strategies. *Games and Economic Behavior*, **14** 220–246.

Koller, D., Megiddo, N. and Von Stengel, B. (1996). Efficient computation of equilibria for extensive two-person games. *Games and economic behavior*, **14** 247–259.

Von Stengel, B. (2002). Computing equilibria for two-person games. *Handbook of Game Theory with Economic Applications*, **3** 1723–1759.

Parr, R. and Russell, S. (1995). Approximating optimal policies for partially observable stochastic domains. In *International Joint Conference on Artificial Intelligence*.

Rodriguez, A. C., Parr, R. and Koller, D. (2000). Reinforcement learning using approximate belief states. In *Advances in Neural Information Processing Systems*.

Hauskrecht, M. (2000). Value-function approximations for partially observable Markov decision processes. *Journal of Artificial Intelligence Research*, **13** 33–94.

Buter, B. J. (2012). *Dynamic Programming for Extensive Form Games with Imperfect Information*. Ph.D. thesis, Universiteit van Amsterdam.

Cowling, P. I., Powley, E. J. and Whitehouse, D. (2012). Information set Monte Carlo tree search. *IEEE Transactions on Computational Intelligence and AI in Games*, **4** 120–143.

Teraoka, K., Hatano, K. and Takimoto, E. (2014). Efficient sampling method for Monte Carlo tree search problem. *IEICE Transactions on Information and Systems*, **97** 392–398.

Whitehouse, D. (2014). *Monte Carlo Tree Search for Games with Hidden Information and Uncertainty*. Ph.D. thesis, University of York.

Kaufmann, E. and Koolen, W. M. (2017). Monte-Carlo tree search by best arm identification. In *Advances in Neural Information Processing Systems*.

Hannan, J. (1957). Approximation to Bayes risk in repeated play. *Contributions to the Theory of Games*, **3** 97–139.

Blum, A. and Mansour, Y. (2007). Learning, regret minimization, and equilibria. *Algorithmic Game Theory* 79–102.

Brown, G. W. (1951). Iterative solution of games by fictitious play. *Activity Analysis of Production and Allocation*, **13** 374–376.

Robinson, J. (1951). An iterative method of solving a game. *Annals of Mathematics* 296–301.

Benaïm, M., Hofbauer, J. and Sorin, S. (2005). Stochastic approximations and differential inclusions. *SIAM Journal on Control and Optimization*, **44** 328–348.

Hart, S. and Mas-Colell, A. (2001). A general class of adaptive strategies. *Journal of Economic Theory*, **98** 26–54.

Monderer, D., Samet, D. and Sela, A. (1997). Belief affirming in learning processes. *Journal of Economic Theory*, **73** 438–452.

Viossat, Y. and Zapechelnyuk, A. (2013). No-regret dynamics and fictitious play. *Journal of Economic Theory*, **148** 825–842.

Kushner, H. J. and Yin, G. G. (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, New York, NY.

Fudenberg, D. and Levine, D. K. (1995). Consistency and cautious fictitious play. *Journal of Economic Dynamics and Control*, **19** 1065–1089.

Hofbauer, J. and Sandholm, W. H. (2002). On the global convergence of stochastic fictitious play. *Econometrica*, **70** 2265–2294.

Leslie, D. S. and Collins, E. J. (2006). Generalised weakened fictitious play. *Games and Economic Behavior*, **56** 285–298.

Benaïm, M. and Faure, M. (2013). Consistency of vanishingly smooth fictitious play. *Mathematics of Operations Research*, **38** 437–450.

Li, Z. and Tewari, A. (2018). Sampled fictitious play is hannan consistent. *Games and Economic Behavior*, **109** 401–412.

Ernst, D., Geurts, P. and Wehenkel, L. (2005). Tree-based batch mode reinforcement learning. *Journal of Machine Learning Research*, **6** 503–556.

Heinrich, J. and Silver, D. (2014). Self-play Monte-Carlo tree search in computer poker. In *Workshops at AAAI Conference on Artificial Intelligence*.

Browne, C. B., Powley, E., Whitehouse, D., Lucas, S. M., Cowling, P. I., Rohlfshagen, P., Tavener, S., Perez, D., Samothrakis, S. and Colton, S. (2012). A survey of Monte Carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, **4** 1–43.

Borkar, V. S. (2008). *Stochastic Approximation: A Dynamical Systems Viewpoint*. Cambridge University Press.

Cesa-Bianchi, N. and Lugosi, G. (2006). *Prediction, Learning, and Games*. Cambridge University Press.

Auer, P., Cesa-Bianchi, N., Freund, Y. and Schapire, R. E. (2002). The nonstochastic multiarmed bandit problem. *SIAM Journal on Computing*, **32** 48–77.

Vovk, V. G. (1990). Aggregating strategies. *Proceedings of Computational Learning Theory*.

Littlestone, N. and Warmuth, M. K. (1994). The weighted majority algorithm. *Information and Computation*, **108** 212–261.

Freund, Y. and Schapire, R. E. (1999). Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, **29** 79–103.

Hart, S. and Mas-Colell, A. (2000). A simple adaptive procedure leading to correlated equilibrium. *Econometrica*, **68** 1127–1150.

Lanctot, M., Waugh, K., Zinkevich, M. and Bowling, M. (2009). Monte Carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems*.

Burch, N., Lanctot, M., Szafron, D. and Gibson, R. G. (2012). Efficient Monte Carlo counterfactual regret minimization in games with many player actions. In *Advances in Neural Information Processing Systems*.

Gibson, R., Lanctot, M., Burch, N., Szafron, D. and Bowling, M. (2012). Generalized sampling and variance in counterfactual regret minimization. In *AAAI Conference on Artificial Intelligence*.

Johanson, M., Bard, N., Lanctot, M., Gibson, R. and Bowling, M. (2012). Efficient Nash equilibrium approximation through Monte Carlo counterfactual regret minimization. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Lisý, V., Lanctot, M. and Bowling, M. (2015). Online Monte Carlo counterfactual regret minimization for search in imperfect information games. In *International Conference on Autonomous Agents and Multi-Agent Systems*.

Schmid, M., Burch, N., Lanctot, M., Moravcik, M., Kadlec, R. and Bowling, M. (2019). Variance reduction in Monte Carlo counterfactual regret minimization (VR-MCCFR) for extensive form games using baselines. In *AAAI Conference on Artificial Intelligence*, vol. 33.

Waugh, K., Morrill, D., Bagnell, J. A. and Bowling, M. (2015). Solving games with functional regret estimation. In *AAAI Conference on Artificial Intelligence*.

Morrill, D. (2016). *Using Regret Estimation to Solve Games Compactly*. Ph.D. thesis, University of Alberta.

Brown, N., Lerer, A., Gross, S. and Sandholm, T. (2019). Deep counterfactual regret minimization. In *International Conference on Machine Learning*.

Brown, N. and Sandholm, T. (2015). Regret-based pruning in extensive-form games. In *Advances in Neural Information Processing Systems*.

Brown, N., Kroer, C. and Sandholm, T. (2017). Dynamic thresholding and pruning for regret minimization. In *AAAI Conference on Artificial Intelligence*.

Brown, N. and Sandholm, T. (2017). Reduced space and faster convergence in imperfect-information games via pruning. In *International Conference on Machine Learning*.

Tammelin, O. (2014). Solving large imperfect information games using CFR+. *arXiv preprint arXiv:1407.5042*.

Tammelin, O., Burch, N., Johanson, M. and Bowling, M. (2015). Solving heads-up limit Texas Hold'em. In *International Joint Conference on Artificial Intelligence*.

Burch, N., Moravcik, M. and Schmid, M. (2019). Revisiting CFR+ and alternating updates. *Journal of Artificial Intelligence Research*, **64** 429–443.

Zhou, Y., Ren, T., Li, J., Yan, D. and Zhu, J. (2018). Lazy-CFR: A fast regret minimization algorithm for extensive games with imperfect information. *arXiv preprint arXiv:1810.04433*.

Zinkevich, M. (2003). Online convex programming and generalized infinitesimal gradient ascent. In *International Conference on Machine Learning*.

Lockhart, E., Lanctot, M., Pérolat, J., Lespiau, J.-B., Morrill, D., Timbers, F. and Tuyls, K. (2019). Computing approximate equilibria in sequential adversarial games by exploitability descent. *arXiv preprint arXiv:1903.05614*.

Johanson, M., Bard, N., Burch, N. and Bowling, M. (2012). Finding optimal abstract strategies in extensive-form games. In *AAAI Conference on Artificial Intelligence*.

Schaeffer, M. S., Sturtevant, N. and Schaeffer, J. (2009). Comparing UCT versus CFR in simultaneous games.

Lanctot, M., Lisý, V. and Winands, M. H. (2013). Monte Carlo tree search in simultaneous move games with applications to Goofspiel. In *Workshop on Computer Games*.

Lisý, V., Kovarik, V., Lanctot, M. and Bošanský, B. (2013). Convergence of Monte Carlo tree search in simultaneous move games. In *Advances in Neural Information Processing Systems*.

Tak, M. J., Lanctot, M. and Winands, M. H. (2014). Monte Carlo tree search variants for simultaneous move games. In *IEEE Conference on Computational Intelligence and Games*.

Kovařík, V. and Lisý, V. (2018). Analysis of hannan consistent selection for Monte Carlo tree search in simultaneous move games. *arXiv preprint arXiv:1804.09045*.

Mazumdar, E. V., Jordan, M. I. and Sastry, S. S. (2019). On finding local Nash equilibria (and only local Nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*.

Bu, J., Ratliff, L. J. and Mesbahi, M. (2019). Global convergence of policy gradient for sequential zero-sum linear quadratic dynamic games. *arXiv preprint arXiv:1911.04672*.

Mescheder, L., Nowozin, S. and Geiger, A. (2017). The numerics of GANs. In *Advances in Neural Information Processing Systems*.

Adolphs, L., Daneshmand, H., Lucchi, A. and Hofmann, T. (2018). Local saddle point optimization: A curvature exploitation approach. *arXiv preprint arXiv:1805.05751*.

Daskalakis, C. and Panageas, I. (2018). The limit points of (optimistic) gradient descent in min-max optimization. In *Advances in Neural Information Processing Systems*.

Mertikopoulos, P., Zenati, H., Lecouat, B., Foo, C.-S., Chandrasekhar, V. and Piliouras, G. (2019). Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *International Conference on Learning Representations*.

Fiez, T., Chasnov, B. and Ratliff, L. J. (2019). Convergence of learning dynamics in Stackelberg games. *arXiv preprint arXiv:1906.01217*.

Balduzzi, D., Racaniere, S., Martens, J., Foerster, J., Tuyls, K. and Graepel, T. (2018). The mechanics of n-player differentiable games. In *International Conference on Machine Learning*.

Sanjabi, M., Razaviyayn, M. and Lee, J. D. (2018). Solving non-convex non-concave min-max games under Polyak-Łojasiewicz condition. *arXiv preprint arXiv:1812.02878*.

Nouiehed, M., Sanjabi, M., Lee, J. D. and Razaviyayn, M. (2019). Solving a class of non-convex min-max games using iterative first order methods. *arXiv preprint arXiv:1902.08297*.

Mazumdar, E., Ratliff, L. J., Jordan, M. I. and Sastry, S. S. (2019). Policy-gradient algorithms have no guarantees of convergence in continuous action and state multi-agent settings. *arXiv preprint arXiv:1907.03712*.

Chen, X., Deng, X. and Teng, S.-H. (2009). Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM*, **56** 14.

Greenwald, A., Hall, K. and Serrano, R. (2003). Correlated Q-learning. In *International Conference on Machine Learning*.

Aumann, R. J. (1974). Subjectivity and correlation in randomized strategies. *Journal of Mathematical Economics*, **1** 67–96.

Perolat, J., Strub, F., Piot, B. and Pietquin, O. (2017). Learning Nash Equilibrium for General-Sum Markov Games from Batch Data. In *International Conference on Artificial Intelligence and Statistics*.

Maillard, O.-A., Munos, R., Lazaric, A. and Ghavamzadeh, M. (2010). Finite-sample analysis of Bellman residual minimization. In *Asian Conference on Machine Learning*.

Letcher, A., Balduzzi, D., Racanière, S., Martens, J., Foerster, J. N., Tuyls, K. and Graepel, T. (2019). Differentiable game mechanics. *Journal of Machine Learning Research*, **20** 1–40.

Chasnov, B., Ratliff, L. J., Mazumdar, E. and Burden, S. A. (2019). Convergence analysis of gradient-based learning with non-uniform learning rates in non-cooperative multi-agent settings. *arXiv preprint arXiv:1906.00731*.

Hart, S. and Mas-Colell, A. (2003). Uncoupled dynamics do not lead to Nash equilibrium. *American Economic Review*, **93** 1830–1836.

Saldi, N., Başar, T. and Raginsky, M. (2018). Markov–Nash equilibria in mean-field games with discounted cost. *SIAM Journal on Control and Optimization*, **56** 4256–4287.

Saldi, N., Başar, T. and Raginsky, M. (2019). Approximate Nash equilibria in partially observed stochastic games with mean-field interactions. *Mathematics of Operations Research*.

Saldi, N. (2019). Discrete-time average-cost mean-field games on Polish spaces. *arXiv preprint arXiv:1908.08793*.

Saldi, N., Başar, T. and Raginsky, M. (2018). Discrete-time risk-sensitive mean-field games. *arXiv preprint arXiv:1808.03929*.

Guo, X., Hu, A., Xu, R. and Zhang, J. (2019). Learning mean-field games. *arXiv preprint arXiv:1901.09585*.

Fu, Z., Yang, Z., Chen, Y. and Wang, Z. (2019). Actor-critic provably finds Nash equilibria of linear-quadratic mean-field games. *arXiv preprint arXiv:1910.07498*.

Hadikhanloo, S. and Silva, F. J. (2019). Finite mean field games: Fictitious play and convergence to a first order continuous mean field game. *Journal de Mathématiques Pures et Appliquées*.

Elie, R., Pérolat, J., Laurière, M., Geist, M. and Pietquin, O. (2019). Approximate fictitious play for mean field games. *arXiv preprint arXiv:1907.02633*.

Anahtarci, B., Kariksiz, C. D. and Saldi, N. (2019). Value iteration algorithm for mean-field games. *arXiv preprint arXiv:1909.01758*.

Zaman, M. A. u., Zhang, K., Miehling, E. and Başar, T. (2020). Approximate equilibrium computation for discrete-time linear-quadratic mean-field games. *Submitted to IEEE American Control Conference*.

Yang, B. and Liu, M. (2018). Keeping in touch with collaborative UAVs: A deep reinforcement learning approach. In *International Joint Conference on Artificial Intelligence*.

Pham, H. X., La, H. M., Feil-Seifer, D. and Nefian, A. (2018). Cooperative and distributed reinforcement learning of drones for field coverage. *arXiv preprint arXiv:1803.07250*.

Tožička, J., Szulyovszky, B., de Chambrier, G., Sarwal, V., Wani, U. and Gribulis, M. (2018). Application of deep reinforcement learning to UAV fleet control. In *SAI Intelligent Systems Conference*.

Shamsoshoara, A., Khaledi, M., Afghah, F., Razi, A. and Ashdown, J. (2019). Distributed cooperative spectrum sharing in UAV networks using multi-agent reinforcement learning. In *IEEE Annual Consumer Communications & Networking Conference*.

Cui, J., Liu, Y. and Nallanathan, A. (2019). The application of multi-agent reinforcement learning in UAV networks. In *IEEE International Conference on Communications Workshops*.

Qie, H., Shi, D., Shen, T., Xu, X., Li, Y. and Wang, L. (2019). Joint optimization of multi-UAV target assignment and path planning based on multi-agent reinforcement learning. *IEEE Access*.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9** 1735–1780.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł. and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*.

Hausknecht, M. and Stone, P. (2015). Deep recurrent q-learning for partially observable mdps. In *2015 AAAI Fall Symposium Series*.

Jorge, E., Kågebäck, M., Johansson, F. D. and Gustavsson, E. (2016). Learning to play guess who? and inventing a grounded language as a consequence. *arXiv preprint arXiv:1611.03218*.

Sukhbaatar, S., Fergus, R. et al. (2016). Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems*.

Havrylov, S. and Titov, I. (2017). Emergence of language with multi-agent games: Learning to communicate with sequences of symbols. In *Advances in neural information processing systems*.

Das, A., Kottur, S., Moura, J. M., Lee, S. and Batra, D. (2017). Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision*.

Peng, P., Wen, Y., Yang, Y., Yuan, Q., Tang, Z., Long, H. and Wang, J. (2017). Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*.

Mordatch, I. and Abbeel, P. (2018). Emergence of grounded compositional language in multi-agent populations. In *AAAI Conference on Artificial Intelligence*.

Jiang, J. and Lu, Z. (2018). Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*.

Jiang, J., Dun, C. and Lu, Z. (2018). Graph convolutional reinforcement learning for multi-agent cooperation. *arXiv preprint arXiv:1810.09202*, **2**.

Celikyilmaz, A., Bosselut, A., He, X. and Choi, Y. (2018). Deep communicating agents for abstractive summarization. *arXiv preprint arXiv:1803.10357*.

Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M. and Pineau, J. (2018). TarMAC: Targeted multi-agent communication. *arXiv preprint arXiv:1810.11187*.

Lazaridou, A., Hermann, K. M., Tuyls, K. and Clark, S. (2018). Emergence of linguistic communication from referential games with symbolic and pixel input. *arXiv preprint arXiv:1804.03984*.

Cogswell, M., Lu, J., Lee, S., Parikh, D. and Batra, D. (2019). Emergence of compositional language with deep generational transmission. *arXiv preprint arXiv:1904.09067*.

Allis, L. (1994). *Searching for Solutions in Games and Artificial Intelligence*. Ph.D. thesis, Maastricht University.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*.

Silver, D., Hubert, T., Schrittwieser, J., Antonoglou, I., Lai, M., Guez, A., Lanctot, M., Sifre, L., Kumaran, D., Graepel, T., Lillicrap, T., Simonyan, K. and Hassabis, D. (2018). A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, **362** 1140–1144.

Billings, D., Davidson, A., Schaeffer, J. and Szafron, D. (2002). The challenge of poker. *Artificial Intelligence*, **134** 201–240.

Kuhn, H. W. (1950). A simplified two-person poker. *Contributions to the Theory of Games*, **1** 97–103.

Southey, F., Bowling, M., Larson, B., Piccione, C., Burch, N., Billings, D. and Rayner, C. (2005). Bayes' bluff: opponent modelling in poker. In *Proceedings of the Twenty-First Conference on Uncertainty in Artificial Intelligence*. AUAI Press.

Bowling, M., Burch, N., Johanson, M. and Tammelin, O. (2015). Heads-up limit hold'em poker is solved. *Science*, **347** 145–149.

Heinrich, J. and Silver, D. (2015). Smooth uct search in computer poker. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*.

Moravčík, M., Schmid, M., Burch, N., Lisý, V., Morrill, D., Bard, N., Davis, T., Waugh, K., Johanson, M. and Bowling, M. (2017). Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, **356** 508–513.

Brown, N. and Sandholm, T. (2018). Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, **359** 418–424.

Burch, N., Johanson, M. and Bowling, M. (2014). Solving imperfect information games using decomposition. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Moravcik, M., Schmid, M., Ha, K., Hladik, M. and Gaukrodger, S. J. (2016). Refining subgames in large imperfect information games. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Brown, N. and Sandholm, T. (2017). Safe and nested subgame solving for imperfect-information games. In *Advances in neural information processing systems*.

Vinyals, O., Ewalds, T., Bartunov, S., Georgiev, P., Vezhnevets, A. S., Yeo, M., Makhzani, A., Küttler, H., Agapiou, J., Schrittwieser, J. et al. (2017). Starcraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*.

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., Choi, D. H., Powell, R., Ewalds, T., Georgiev, P. et al. (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature* 1–5.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D. and Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*.

Lerer, A. and Peysakhovich, A. (2017). Maintaining cooperation in complex social dilemmas using deep reinforcement learning. *arXiv preprint arXiv:1707.01068*.

Hughes, E., Leibo, J. Z., Phillips, M., Tuyls, K., Dueñez-Guzman, E., Castañeda, A. G., Dunning, I., Zhu, T., McKee, K., Koster, R. et al. (2018). Inequity aversion improves cooperation in intertemporal social dilemmas. In *Advances in neural information processing systems*.

Cai, Q., Yang, Z., Lee, J. D. and Wang, Z. (2019). Neural temporal-difference learning converges to global optima. *arXiv preprint arXiv:1905.10027*.

Arora, S., Cohen, N. and Hazan, E. (2018). On the optimization of deep networks: Implicit acceleration by overparameterization. *arXiv preprint arXiv:1802.06509*.

Li, Y. and Liang, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Advances in Neural Information Processing Systems*.

Brafman, R. I. and Tennenholtz, M. (2000). A near-optimal polynomial time algorithm for learning in certain classes of stochastic games. *Artificial Intelligence*, **121** 31–47.

Brafman, R. I. and Tennenholtz, M. (2002). R-max-A general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, **3** 213–231.

Tu, S. and Recht, B. (2018). The gap between model-based and model-free methods on the linear quadratic regulator: An asymptotic viewpoint. *arXiv preprint arXiv:1812.03565*.

Sun, W., Jiang, N., Krishnamurthy, A., Agarwal, A. and Langford, J. (2019). Model-based RL in contextual decision processes: PAC bounds and exponential improvements over model-free approaches. In *Conference on Learning Theory*.

Lin, Q., Liu, M., Rafique, H. and Yang, T. (2018). Solving weakly-convex-weakly-concave saddle-point problems as weakly-monotone variational inequality. *arXiv preprint arXiv:1810.10207*.

García, J. and Fernández, F. (2015). A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, **16** 1437–1480.

Chen, Y., Su, L. and Xu, J. (2017). Distributed statistical machine learning in adversarial settings: Byzantine gradient descent. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, **1** 44.

Yin, D., Chen, Y., Ramchandran, K. and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. *arXiv preprint arXiv:1803.01498*.