Artificial Intelligence Review
https://doi.org/10.1007/s10462-020-09938-y

A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications

Wei Du[1] · **Shifei Ding**[1,2]

Abstract
Deep reinforcement learning has proved to be a fruitful method in various tasks in the field of artificial intelligence during the last several years. Recent works have focused on deep reinforcement learning beyond single-agent scenarios, with more consideration of multi-agent settings. The main goal of this paper is to provide a detailed and systematic overview of multi-agent deep reinforcement learning methods in views of challenges and applications. Specifically, the preliminary knowledge is introduced first for a better understanding of this field. Then, a taxonomy of challenges is proposed and the corresponding structures and representative methods are introduced. Finally, some applications and interesting future opportunities for multi-agent deep reinforcement learning are given.

Keywords Deep reinforcement learning · Multi-agent · Game theory · Centralized training and decentralized execution · Communication learning · Agent modeling

1 Introduction

Reinforcement Learning (RL) is often considered to be a general formalization of decision-making tasks and a subfield of machine learning. In RL, agents learn not from sample data, as in supervised and unsupervised learning, but from experiences that interact with the environment. With the success of deep neural networks (DNN), reinforcement learning algorithms combine with it and form deep reinforcement learning (DRL) methods to solve complex problems in the real world. The pioneering model is Deep Q-Network, which was able to play Atari console games without adjusting network architecture or hyperparameters. Deep reinforcement learning methods have been extensively researched and significantly improved since then.

Most successful DRL methods have been in the single-agent domains so far, and extending DRL to multi-agent settings is indispensable. However, deep reinforcement learning

✉ Shifei Ding
dingsf@cumt.edu.cn

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

[2] Mine Digitization Engineering Research Center of Ministry of Education of the People's Republic of China, Xuzhou 221116, China

Published online: 24 November 2020

for multi-agent settings is fundamentally more difficult than the single-agent scenario due to the presence of multi-agent pathologies such as the curse of dimensionality and multi-agent credit assignment. Despite this complexity, there has been a lot of work in the fields of general control, robot system (Gu et al. 2017; Kurek and Jakowski 2016), man–machine game (Fu et al. 2019; Lanctot et al. 2017; Leibo et al. 2017), autonomous driving (Shalev-Shwartz et al. 2016), Internet advertising (Jin et al. 2018), and resource utilization (Xi et al. 2018; Perolat et al. 2017).

This paper systematically summarizes several research directions in the field of multi-agent deep reinforcement learning (MDRL), including scalability, non-stationarity, partial observability, communication learning, coordinated exploration, agent modeling. The rest of this paper is structured as follows: firstly, the basic theory of multi-agent reinforcement learning is reviewed in Sect. 2. The latest work of deep reinforcement learning for single-agent and multi-agent settings are summarized in Sect. 3. The applications and prospect of multi-agent deep reinforcement learning are discussed in Sect. 4. The conclusion and directions are given in Sect. 5.

2 Background

2.1 Single-agent reinforcement learning

In single-agent reinforcement learning, the agent learns through interaction with the dynamic environment by a trial and error procedure as shown in Fig. 1. The goal of the agent is to learn an optimal policy by maximizing the expected value of the cumulative sum of rewards.

The basic framework of reinforcement learning is the Markov decision process, which is a random process represented by the tuple (S, A, R, P).

(1) S is the state space, $s_t \in S$ represents the state that the agent is in at the timestep t
(2) A is the action space, $a_t \in A$ represents the action taken by the agent at the timestep t
(3) R is the reward,$r_t \sim \rho(s_t, a_t)$ represents immediate reward value received by the agent in state s_t performing action a_t

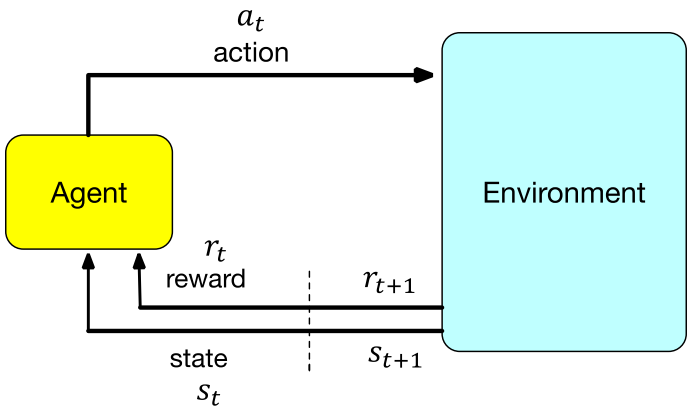

Fig. 1 A single agent interacting with its environment

(4) P is the state transition probability, $p[s_{t+1}|s_t, a_t]$ represents the probability that the agent takes action a_t and transfer from state s_t to next state s_{t+1}.

In RL, the agent in the state s_t selects action a_t depending on policy π and takes the action and transfers to the next state s_{t+1} with the probability $p[s_{t+1}|s_t, a_t]$, meanwhile receives the reward r_t from the environment. Assuming that the reward for each timestep t must be multiplied by a discount factor γ to determine how much importance is to be given to the immediate reward and future rewards, the cumulative rewards from the timestep t to the end of the timestep T is represented as $R_t = \sum_{t'=t}^{T} \gamma^{t'-t} r_t$. The value function Q_π(s, a) refers to taking action a in the current state s and taking the policy π until the episode ends. In this process, the cumulative rewards obtained by agents can be expressed as follows:

$$Q_\pi(s, a) = E_\pi\left[R_t | s_t = s, a_t = a\right] \tag{1}$$

If the expected reward of policy π^* is greater than or equal to the expected reward of all the other policy for all states, then the policy π^* can be called an optimal policy. There may be more than one optimal policy, but they share a common function:

$$Q_*(s, a) = \max_\pi E_\pi\left[R_t | S_t = s, A_t = a\right] \tag{2}$$

The function is called the optimal action-value function, and it follows the Bellman optimality equation:

$$Q_*(s, a) = E_{s' \sim S}\left[r + \gamma \max_{a'} Q\left(s', a'\right) \middle| s, a\right] \tag{3}$$

Linear function approximators are used to approximate the action-value function usually, i.e. $Q(s, a|\theta) \approx Q_*(s, a)$. In addition, deep neural network and other nonlinear function approximators can be used to approximate action-value functions or policy.

2.2 Multi-agent reinforcement learning

The framework of multi-agent reinforcement learning is a stochastic game based on the Markov decision process represented by the tuple $S, A_1 \ldots A_n, R_1 \ldots R_n, P$. Where n refers to the number of agents, $A = A_1 \times \ldots \times A_n$ is the joint action space of all agents, $R_n : S \times A \times S \to R$ is the reward function of each agent, $P : S \times A \times S \to [0, 1]$ is the state transition function, where it assumes that the reward function is bounded (Littman 1994). (Fig. 2).

In the case of multi-agent settings, state transitions are the result of all agents acting together, so the rewards of agents depend on the joint policy, which is represented as $H : S \times A \to [0, 1]$, and the corresponding reward for each agent is

$$R_i^H = E[R_{t+1} | S_t = s, A_{t,i} = a, H] \tag{4}$$

The bellman equation is

$$v_i^H(s) = E_i^H\left[R_{t+1} + \gamma V_i^H\left(S_{t+1}\right) | S_t = s\right] \tag{5}$$

$$Q_i^H(s, a) = E_i^H\left[R_{t+1} + \gamma Q_i^H\left(S_{t+1}, A_{t+1}\right) | S_t = s, A_t = a\right] \tag{6}$$

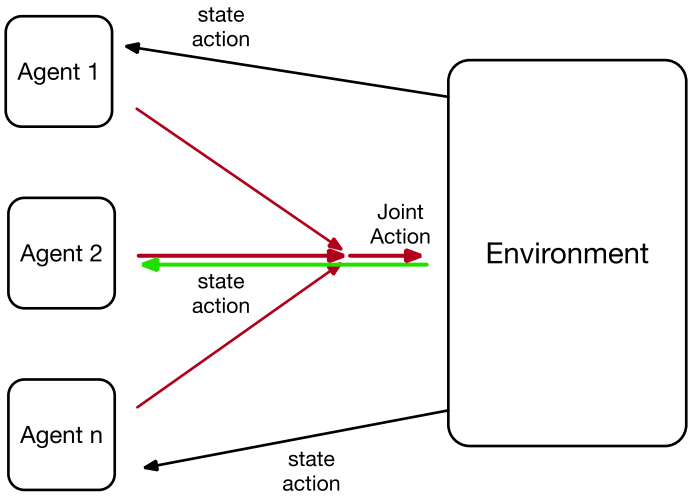

Fig. 2 Multi agents interacting with the environment

Stochastic games can be divided into fully cooperation games, fully competitive games, and mixed games. In a fully cooperative stochastic game, the reward function is the same for all agents, $R_1{=}R_2{=}...{=}R_n$ so the rewards are the same, and the goal of agents is to maximize the common rewards. In a fully competitive stochastic game, for example, if $n = 2, R_1 = -R_2$, then the two agents have opposite goals. In mixed games, agents' rewards are usually different and correlated.

It is a challenge to specify good general goals for agents. Reviewing the previous literature on the definition of learning goals, it mainly can be summarized as two aspects, stability and adaptability. Stability means that agents can converge to a stable policy. Adaptability ensures that the performance of agents does not decrease as other agents change their policy (Buşoniu et al. 2010).

2.2.1 Fully cooperation game

In a fully cooperative stochastic game, agents have the same reward function, and the learning goal is to maximize the common discounted reward. If a centralized structure is available, the goal can be expressed by learning the optimal joint-action values. Q-learning is a common method to learn it:

$$\begin{aligned} &Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) \\ &+\alpha\left[r_{t+1} + \gamma \max_{a'} Q_{t+1}(s_{t+1}, a') - Q_t(s_t, a_t)\right] \end{aligned} \tag{7}$$

Each agent uses a greedy policy to maximize common rewards:

$$h_i(x) = \arg\max_{a_i} \max_{a_i \ldots a_n} Q^*(s, a) \tag{8}$$

It is necessary to consider the cooperation between agents. Team-Q algorithm (Littman 2001) solves cooperation problems by assuming that optimal joint action is unique. Distributed-Q algorithm (Lauer and Riedmiller 2000) solves problems with a limited computational

complexity which is similar to that of Q learning. However, this algorithm is only applicable to deterministic problems with non-negative reward functions.

2.2.2 Fully competitive game

In a fully cooperative stochastic game, agents have the same reward function, and the learning goal is to maximize the common discounted reward. If a centralized structure is available, the goal can be expressed by learning the optimal joint-action values. Q-learning is a common method to learn it:

$$h_{1,t}(s_t, \cdot) = \arg m_1(Q_t, s_t) \tag{9}$$

Each agent uses a greedy policy to maximize common rewards:

$$\begin{aligned} Q_{t+1}(s_t, a_{1,t}, a_{2,t}) &= Q_t(s_t, a_{1,t}, a_{2,t}) + \\ \alpha\left[r_{k+1} + \gamma m_1(Q_t, a_{t+1}) - Q_t(s_t, a_{1,t}, a_{2,t})\right] \end{aligned} \tag{10}$$

It is necessary to consider the cooperation between agents. Team-Q algorithm (Littman 2001) solves cooperation problems by assuming that optimal joint action is unique. Distributed-Q algorithm (Lauer and Riedmiller 2000) solves problems with a limited computational complexity which is similar to that of Q learning. However, this algorithm is only applicable to deterministic problems with non-negative reward functions.

where m_1 is the minimax reward of agent 1

$$m_1(Q, s) = \max_{h_{1(s,\cdot)}} \min_{a_2} \sum_{a_1} h_1(s, a_1) Q(s, a_1, a_2) \tag{11}$$

where the policy of agent 1 in the state s is represented by $h_{1(s,\cdot)}$, and the point represents the parameter of the action.

2.2.3 Mixed game

In mixed dynamic stochastic games, the methods are mainly divided into agent-independent methods and agent-aware methods. Agent-independent methods generally adopt a common structure based on Q learning, where policy and state values are calculated using the game theory solver in SG. One representative approach of agent-independent methods is Nash Q-learning (Hu and Wellman 2003), and there are also Correlated Q-learning (CE-Q) (Greenwald et al. 2003) or Asymmetric Q-learning (Kononen 2004) to solve equilibrium problems by using correlation or Stackelberg (leader–follower) equilibrium respectively. Agent-aware methods typically do consider convergence, in which the representative algorithm is the Win-or-Learn-Fast Policy Hill-Climbing (WoLF-PHC) (Xi et al. 2015). Many methods for mixed SG suffer from scalability challenges and are sensitive to partial observability, the latter one holds for agent-independent algorithms especially.

3 Multi-agent deep reinforcement learning

3.1 Deep reinforcement learning

Deep reinforcement learning algorithms can be roughly divided into two categories: value-based,policy-based. Value-based methods construct optimal policy by gaining an approximation of optimal function $Q_*(\mathrm{s}, \mathrm{a})$ using dynamic programming. In DRL, Q-function is represented with deep neural network. Policy-based algorithms directly optimize policy π^*without additional information about MDP, using gradient approximate estimations relative to policy parameters.

3.1.1 Deep Q-network

Deep Q-Network (DQN) proposed by Mnih et al. (2013) is a representative method of Value-based methods. Concisely, the DQN structure leverages the deep neural network to directly extract a representation of input state from the environment. The output of DQN produces Q-values of all possible actions. Therefore, DQN can be considered as a value network parameterized by β, which is trained continually to the approximate optimal policy. Mathematically, DQN uses the Bellman equation mentioned before to minimize the loss function $\mathcal{L}(\beta)$

$$\mathcal{L}(\beta) = E\left[\left(r + \gamma \max Q\left(s', a'|\beta\right) - Q(s, a|\beta)\right)^2\right] \tag{12}$$

The demerit of using the neural network to approximate value function is unstable and may cause divergence due to the bias of correlative samples. Mnih et al. presented a target network parameterized by $\beta'$to make the samples uncorrelated. The target network is updated in every N step from the estimation network. Besides, generated samples are stored in an experience replay memory and are retrieved randomly from it. Therefore, Eq. (13) can be rewritten as:

$$\mathcal{L}(\beta) = E\left[\left(r + \gamma \max Q\left(s', a'|\beta'\right) - Q(s, a|\beta)\right)^2\right] \tag{13}$$

$$\beta' \leftarrow \beta \; for \; every \; N \; steps \tag{14}$$

On the basis of DQN, a variety of deep reinforcement learning algorithms was proposed. Double Deep Q-network (DDQN) was proposed by Hasselt et al. By applying Double Q-learning to DQN, it can separate action selection and policy evaluation, reducing the risk of overestimating Q value [26~27]. Dueling Deep Q-network was put forward by Wang et al., the model divides the abstract features extracted by CNN into two branches, one of which represents the state value function and the other represents the advantage function. Through this dueling network structure, agents can identify the correct behaviors faster in the process of policy evaluation and network architecture can be better integrated (Wang et al. 2015). Schaul et al. proposed a double deep Q network with proportional prioritization based on DDQN. This method replaces uniform sampling with priority-based sampling, improves the sampling probability of some valuable samples, and thus speeds up the learning of optimal policy (Schaul et al. 2015). Lakshminarayanan et al. proposed Dynamic Frame Skip Deep Q-Network (DFDQN), which uses dynamic frameskip

to replace action repeated k times per moment in DQN. Experiments show that DFDQN achieves better performance in some Atari 2600 games (Lakshminarayanan et al. 2016). Vincent et al. use the adaptive discount factor and learning rate in DQN, the convergence speed of the deep network is accelerated (Francois-Lavet et al. 2015). Tom Schaul et al. developed a framework for prioritizing experience and used it in DQN to replay important transitions more frequently and learn more effectively (Schaul et al. 2015). Fortunato et al. (2017) proposed adding noise to the parameters instead of ϵ-greedy to increase the exploration ability of the model. The success of DQN encouraged full-scale research of value-based methods by studying various demerits of DQN and developing auxiliary extensions. Hessel et al. proposed Rainbow DQN (Hessel et al. 2017), which uniting seven Q-learning-based ideas in one procedure to verify whether these merged extensions are essentially necessary for the RL algorithms.

3.1.2 Deep deterministic policy gradient

The actor-critic algorithm is a widely used policy-based method. The structure consists of two networks, a policy network called the actor network and a value network called the critic network. Actor network input state and output action while critic network input state and action and output Q value (Zhao et al. 2019, 2018; Ding et al. 2019). Lillicrap et al. proposed the Deep Deterministic Policy Gradient (DDPG) method based on the actor-critic structure, which can be used to tackle the problem of continuous action space in DRL. Experiments show that DDPG is not only stable in a series of continuous action space tasks, but also requires far fewer time steps than DQN to obtain the optimal policy. Compared with the DRL method based on value function, the deep deterministic policy gradient method based on the AC framework has higher optimization efficiency and faster speed (Lillicrap et al. 2016; Silver et al. 2014). Besides, various algorithms are derived based on the AC model, such as asynchronous advantage action critic (A3C) algorithm (Mnih et al. 2016) and distributed proximal policy optimization (DPPO) algorithm (Schulman et al. 2017; Heess et al. 2017). A3C algorithm was proposed by Mnih in 2016. In A3C, multi-thread is used to collect data in parallel, and each thread is an independent agent searching for an independent environment. At the same time, each agent can use a different exploration policy to sample in parallel, so that the samples obtained by each thread are naturally unrelated and the sampling speed is faster. Fujimoto et al. (2018) proposed Twin Delayed DDPG (TD3) algorithms to tackle the overestimation problems in actor-critic structure by taking the minimum value between a pair of critics. Tuomas et al. proposed a soft actor-critic method based on the maximum entropy RL framework. The actor network maximizes expected rewards while also maximizing entropy (Haarnoja et al. 2018).

3.2 MDRL research progress: challenges and structures

Many MDRL methods suffer from scalability issues and are sensitive to partial observability (Partial observability means that the agents do not know complete information of states pertaining to the environment when they interact with the environment). There are two mainstream structures used in multi-agent settings. The first structure is decentralizing training and decentralizing execution structure (DTDE), each agent is trained independently of the other agents. This structure can handle the scalability problems caused by the growth of the number of agents, but other problems are emerging, including environment non-stationary, reward distribution, and independent agents is sensitive to partial

observability. The paradigm of centralized training and centralized execution (CTCE) can tackle these problems, in which agents are modeled together to learn a joint policy. The disadvantage of this centralized structure is the huge input and output space dimension. With the increase in the number of agents, the space dimension of the output joint policy renders an exponential increase. Another main structure is centralized training and decentralized execution (CTDE), which can solve partial observability problems meanwhile avoid huge input and output space dimensions caused by centralized execution.

3.2.1 Scalability and DTDE

Scalability is one of the core issues of MDRL domains. Scalability mainly refers to the extension from single-agent environment to multi-agent environment, including the expansion of state dimension and action dimension and the number of agents.

Tampuu et al. (2017) first proposed playing the Atari Pong game with two independent DQN agents. The result indicates that DQNs can be extended for the decentralized learning of multi-agent systems. It is the earliest work to adopt the DTDE framework to solve the scalability problem. Leibo et al. (2017) applied independent DQN to Sequential Social Dilemmas issues and they analyzed the dynamics of multiple independent agent learning policy, each using its deep Q network. Foerster et al. (2017) proposed two methods to improve the experience replay method to make multi-agents more stable and compatible. On the one hand, it adopted importance sampling to naturally attenuate outdated data. On the other hand, each agent can infer the action of other agents by observing the policy of other agents.

Song et al. (2018) proposed a new multi-agent policy gradient algorithm, which solved the high variance gradient estimation problem and effectively optimized multi-agent cooperative tasks in a highly complex particle environment. Wai et al. (2018) proposed a decentralized local exchange scheme to make the method more extensible and robust. Each agent communicates only with its neighbors through the network and iterates spatially and temporally to combine adjacent gradient information and local reward information respectively. Abouheaf and Gueaieb (2017) proposed an online adaptive reinforcement learning method for multi-agent settings based on graph interaction. In this method, the bellman equation of multi-agent setting is solved by the reduced value function, reducing the computational complexity and solving the large-scale optimization problem. Palmer et al. (2018) proposed LDQN algorithm, which introduced the lenient policy into the deep Q network and adopted the leniency treatment method for the update of negative policies to improve the convergence and stability.

The weakness of independent agents is that by treating other agents as part of the environment, it ignores the fact that their policies change over time. While independent agent avoids the scalability caused by centralized learning, it brings a new problem that the environment becomes non-stationary from the point of view of each agent. In this situation, the optimal policy of an agent can be affected by the learning among agents. Meanwhile, the convergence theory of Q-learning applied in single-agent reinforcement is not suitable for most multi-agent settings as the Markov property is no longer valid in the non-stationary environment.

3.2.2 Partial observability and CTDE

In real-world tasks, there are many situations where the environment is partially observable. In other words, when agents interact with the environment, they do not know all the information about the state of the environment. This type of problem is typically modeled using a partially observable Markov decision process (POMDP).

In single-agent deep reinforcement learning, there are many models and algorithms dealing with POMDP, among which deep recurrent Q-network (DRQN) is the most representative. Hausknecht and Stone (2015) modified DQN combined a Long short-term Memory with DQN to tackle the noisy observations problems of POMDP. Although the DRQN sees only one frame at a time, it can still combine information across frames to detect relevant information, such as the speed of objects on the screen. Foerster et al. (2016) proposed Reinforced Inter-Agent Learning (RIAL) which is also based on DRQN to address the partial observability problems and they proposed two variations on it. In one variant, agents learn policy using their own network parameters as independent Q-learning, treating the others as part of the environment. In another variant, agents use the same network in which parameters are shared among all agents.

Gupta et al. (2017) introduced parameter sharing (PS) method to improve learning in homogeneous partially observable multi-agent environments where agents have the same action space. The idea is that a globally shared policy network can still perform differently through different inputs (individual agent observation). They tested three different methods through parameter sharing: PS-DQN, PS-DDPG and PS-TRPO, and the results showed that PS-TRPO was superior to the other two methods.

In fact, in many multi-agent settings, partial observability needs the learning of decentralized policy, because on the one hand it only necessitates local action observation history of each agent. On the other hand, decentralized policies naturally address the problem that joint action spaces grow significantly with the number of agents. Fortunately, decentralized policies are often learned in a centralized way. Partial observability problem can be tackled by centralized training and decentralized execution structure. Lowe et al. (2017) adopt the CTDE paradigm, allowing the policies to use others information to improve training. They proposed an extension of actor-critic setting where the critics added extra information about other agents' policies, while the actor only has local information. After training is completed, only the local actors are used at the execution phase, acting in a decentralized manner. Concretely, consider a game having N agents whose policies are $\mu =$ and policies parameters are $\theta=\{\theta_1 \ldots, \theta_N\}$.Then the gradient of the expected reward for agent i with policy $\mu_i, J(\theta_i) = E[R_i]$ is present as:

$$\nabla_{\theta_i} J(\theta_i) = E_{x,a\sim D}\left[\mu_i(o_i)\nabla a_i Q_i^{\mu}(x, a_1, \ldots, a_N)\middle| a_i = \mu_i(o_i)\right] \tag{15}$$

where $Q_i^{\mu}(x, a_1, \ldots, a_N)$ is a centralized action-value function that takes all agents' action $a_1, \ldots, a_N$ and some state information x (i.e., $\mathrm{x}=(o_1, \ldots o_N)$) as input, and outputs the Q-value for agent i. The experience replay buffer D contains the tuples $(x, x', a_1, ..., a_N, r_1 ... r_N)$ recording experiences of all agents, where x' represents the next state after taking actions $a_1, ..., a_N$ The centralized action-value function Q_i^{μ} is updated as:

$$L(\theta_i) = E_{x,a,r,x'}\left[Q_i^{\mu}(x, a_1, \ldots, a_N) - y\right]^2 \tag{16}$$

$$y = r_i + \gamma Q_i^{\mu'}\left(x', a'_1, \ldots, a'_N\right)\Big|_{a'_j = \mu'_j(o_j)} \tag{17}$$

where $\mu' = \left\{\mu_{\theta'_1}, \ldots, \mu_{\theta'_N}\right\}$ is the set of target policies with delayed parameters θ'_i. It is worth noting that the centralized Q function is only used during training, while each policy μ_{θ_i} only takes local information o_i to produce an action during decentralized execution. (Fig. 3).

Many works have emerged based on this framework. Li et al. (2019) studied the problem of training robust DRL agents with continuous action in multi-agent learning settings, so that the trained agents still have the generalization ability when the opponent's policy changes. They proposed the MiniMax Multi-agent Deep Deterministic Policy Gradient (M3DDPG) method with two main contributions: on the one hand, they introduced a minimax extension of the popular MADDPG algorithm for robust policy learning. On the other hand, because the continuous action space leads to the computational difficulty of minimax learning objectives, they proposed Multi-Agent Adversarial Learning (MAAL) to efficiently tackle that problem.

Foerster et al. (2018) proposed a novel multi-agent actor-critic method like MADDPG called counterfactual multi-agent policy gradients (COMA) to learn decentralized policies for cooperative agents. In addition, COMA addresses credit assignment problems in multi-agent setting by using a counterfactual baseline. That baseline keeps the other agents' actions fixed and marginalizes out the action of a single agent. Then, by comparing the current Q value with the baseline, an advantage function can be calculated. This counterfactual was inspired by difference rewards (Tumer and Agogino 2007), which is a way to capture individual contributions from agents in a coordinated multi-agent setting.

Coordinated MDRL application problems such as coordinating self-driving vehicles often could be treated using a centralized approach. However, Sunehag et al. (2017) found that the centralized approach consistently fails on these simple cooperative MDRL tasks in practice. Specifically, the centralized approach learned inefficient policies with only one agent behaving actively and the other being "lazy". It occurs when an agent learns an effective policy, then the other agents are discouraged from learning. The authors proposed a new value decomposition network (VDN) architecture to solve these problems and trains

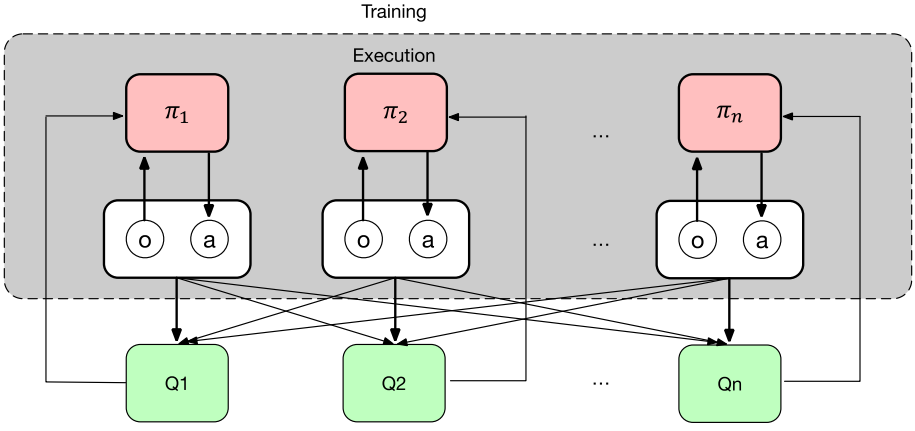

Fig3 Architecture of Multi-Agent Deep Deterministic Policy Gradient

individual agents to decompose team value functions into agent-wise value functions. They proved that value-decomposition has a much better performance than centralization or fully independent learners.

Rashid et al. (2018) proposed a novel value-based method called QMIX. They demonstrated that the full decomposition of VDN is unnecessary to extract decentralized policies. QMIX is an extension of VDN, which gets the joint action-value function by summarizing local action-value functions of each agent. QMIX adopts a hybrid network to merge local value functions of single- agent and adds global state information in the training and learning process to improve algorithm performance(Fig. 4).

However, VDN and QMIX achieve the value decomposition heuristically without valid theoretical groundings. Yang et al. (2020) theoretically derived a linear decomposing formation from Q_{tot} to each Q_i. Based on this theoretical discovery, they introduced multi-head attention mechanisms to approximate the decomposition of each term in the formula and provided theoretical explanations. Son et al. (2019) proposed a new factorization method named QTRAN, which is not subject to structural constraints in the factorization process of VDN and QMIX. Yang et al. (2020) proposed the Q-value Path Decomposition (QPD) method to decompose global Q-values of the system into Q-values of individual agents leveraging the integrated gradient attribution technique (Table 1).

In addition to the above classification, the next sections describe two current mainstream branches, communication learning and agent modeling, in which most of the methods are also used to solve scalability, partial observability, non-stationary problem in MDRL. Actually, some of them can also be categorized into the table above.

3.3 Communication learning

Communication between agents is one of the hotspots in the MDRL field in recent years. Communication learning setting usually considers a set of cooperated agents in the partially observable environment where agents exchange information through communication to promote the cooperation between agents.

Foerster et al. (2016) first proposed the mechanism of updating the communication model through the backpropagation method, namely Differentiable Inter-Agent Learning (DIAL). DIAL takes information from other agents as input, providing a reference for agent decision-making. Gradient flow will update the communication generator layer at the

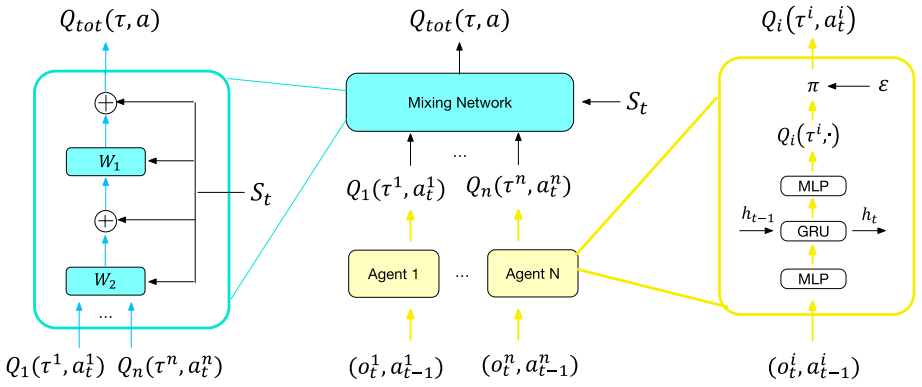

Fig.4 Architecture of QMIX

Table 1 Multi-agent deep reinforcement learning main Challenges and Structure

Challenges	Structure	Value-based	Policy-based
Scalability	Mostly use decentralized training, decentralized execution	Tampuu et al. (Tampuu et al. 2017)	Song et al. (Song et al. 2018)
		Leibo et al. (Leibo et al. 2017)	Wai et al. (Wai et al. 2018)
		Foerster et al. (Foerster et al. 2017)	Abouheaf et al. (Abouheaf and Gueaieb 2017)
		Lenient-DQN (Palmer et al. 2018)	
Partial observability	Mostly use centralized training and decentralized execution	DRQN (Hausknecht and Stone 2015)	MADDPG (Lowe et al. 2017)
		RIAL (Foerster et al. 2016)	M3DDPG (Li et al. 2019)
		Gupta et al. (Gupta et al. 2017)	COMA (Foerster et al. 2018)
		VDN (Sunehag et al. 2017)	CommNet (Sukhbaatar and Fergus 2016)
		QMIX (Rashid et al. 2018)	BiCNet (Peng et al. 2017)
		Yang et al. (Yang et al. 2020)	ACCNET (Mao et al. 2017)
		Son et al. (Son et al. 2019)	ATOC (Jiang and Lu 2018)
		Yang et al. (Yang et al. 2020)	

same time. Foerster et al. proposed the Deep Distributed Recurrent Q-network (DDRQN) based on DRQN, which enables the agent teams to learn to solve communication problems and establish coordination tasks. The agents do not receive any pre-designed communication protocols in these tasks so that they must automate the development first and agree on their communication protocols. This is the first success of deep reinforcement learning in learning communication protocols.

Sukhbaatar et al. (2016) proposed the CommNet method in a similar way. A single agent extracts information from other agents through broadcasting. The CommNet model is showed in Fig. 5. The left one is a view of the model for single agent that the parameters are shared across all agents. The middle one is a single communication step where each agent modules propagate their internal state, as well as broadcasting a communication vector on a common channel (shown in yellow). The Right one is the full model, showing input states for each agent, two communication steps and the output actions for each agent.

Peng et al. (2017) came up with a new communication network, Multi-agent Bidirectional-Coordinated Nets (BicNet). BicNet uses bi-directional RNN as a communication channel and stores local state memory. The BicNet is different from DIAL, CommNet in that BicNet can adjust the order of agents joining communication networks and support continuous action space. However, BicNet requires each agent to obtain a global state.

Mao et al. (2017) proposed an Actor-Coordinator-Critic Net (ACCNet) framework for solving learning-to-communicate problems in multi-agent setting. The ACCNet completes the output of the integrated state through communication channel, and then uses integrated state to complete the agent-independent actor-critic learning process. On the other hand, ACCNet carries out communication in the critic part to obtain better estimated Q-values. Jiang et al. (2018) proposed a novel method named ATOC to use attention to complete communication. ATOC uses attention unit to build a communication group (select agents for information sharing) and bi-directional LSTM unit as a communication channel. In addition, ATOC supports an environment of a large number of agents.

3.4 Agent modeling

An important ability of agents is to reason about the other agents' behaviors. By constructing model agents can make predictions about the modeled agents' properties of interest, including actions, goals, beliefs. Typically, a model is a function that takes a portion of

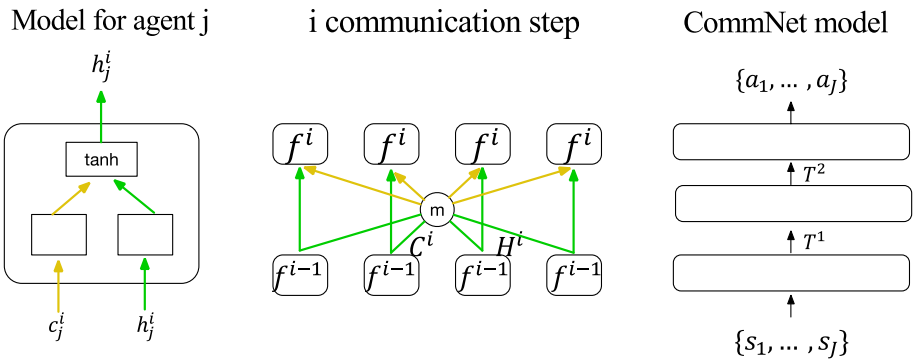

Fig. 5 Architecture of CommNet

the observed interaction history as input and outputs predictions about some of the related properties of the modeled agent. The interaction history contains information such as past actions taken by the modeled agent in various situations. Various modeling approaches now exist in multi-agent environments, with vastly different approaches and underlying assumptions.

He et al. (2016) proposed Deep Reinforcement Opponent Network (DRON) algorithm, which is an early work using deep neural networks for agents modeling. The network architectures have two networks, one evaluates Q values and the other one learns a representation of the opponent policy. In addition, they proposed to have several expert networks. Each expert network captures one type of opponent policy to combine their predictions in order to get the estimated Q value. While DRON defines the opponent network using hand-crafted features, Deep Policy Inference Q-Network (DPIQN) (Hong et al. 2018) learn "policy features" directly from original observations of the other agents. DPIQN consists of three main parts: Q value learning module, feature extraction module, and auxiliary policy feature learning module. The first two modules are responsible for learning the Q value, while the last module is mainly concerned with learning hidden representations from other agent policies.

A variety of opponent modeling approaches are learned from observations. Raileanu et al. (2018) proposed a different approach Self Other Modeling (SOM). SOM uses agents' policies to predict the actions of opponents. The authors present a new method for inferring hidden states from the behavior of other agents and using these estimates to select actions. SOM uses two networks, one to calculate the agent's own policy and the other to infer the opponent's target. This approach does not require any additional parameters to model other agents. These networks have the same input parameters but the different values of the agent or the opponent. Compared with previous methods, the focus of SOM is not to learn the opponent's policy, but to estimate the opponent's target. (Fig. 6)

Rabinowitz et al. (2018) came up with a new neural network, Theory of Mind Network (ToMnet), that is capable of understanding the mental states of itself and the agents around it. Theory of mind refers broadly to the ability of humans to represent the mental states of others, such as desires, beliefs, and intentions. Theory of mind is part of recursive reasoning approaches (Gmytrasiewicz and Doshi 2005; Gmytrasiewicz and Durfee 2000; Camerer et al. 2004; Carmel and Markovitch 1996). In these approach agents have explicit beliefs about the other agents' mental states. ToMnet is composed of three networks, the first one is a character network that learns from past information, the second one is a mental state network that takes the character output and the most recent trace as its input; the third one is the prediction network, its input are the current state and the outputs of the other two networks. The output is to predict the opponent's next action in general. ToMnet can learn a generic model of agents in a training distribution and build agent-specific models whilst observing the actions of new agents.

Yang et al. (2019) proposed Deep Bayesian Theory of Mind Policy (Bayes-ToMoP), which takes inspiration from theory of mind. Their setup assumes that the opponent has a

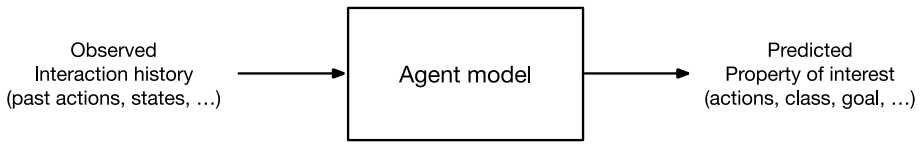

Fig.6 general agent model

different set of policies to act on, and changes over time. Earlier work such as BPR + (Hernandez-Leal et al. 2016) extends the Bayesian policy reuse framework (Rosman et al. 2016) to multi-agent domains to deal with this setup. Deep Bayes-ToMoP provide a higher-level reasoning policy than BRP + by using theory of mind. In addition, Deep BPR + Zheng et al. (2018) method is also inspired by BPR + . It uses not only environmental rewards but also the idea of online learning opponent model (Hernandez-Leal and Kaisers 2017) to construct a rectified belief on the opponent policy. In addition, it extracts ideas from policy distillation (Rusu et al. 2015; Hinton et al. 2015) and extends them to multi-agent settings to create a policy network of distillation.

Lanctot et al. (2017) quantified a serious problem with independent reinforcement agents, joint policy Correlation (JPC), which limits the versatility of these methods. They presented a generalized multi-agent reinforcement learning algorithm that includes several previous algorithms. They demonstrated that PSRO/DCH produces a general policy of significantly reducing JPC in partially observable coordinated games in their experiment. (Table 2).

4 Applications and prospect of MDRL

In recent years, MDRL methods have been applied in various fields to solve complex real-world tasks. This section outlines the application of these methods in different domains (Mao et al. 2019; Wang et al. 2019; Tan 1993; Duan et al. 2016).

MDRL is used in various domains such as autonomous driving, Internet marketing, resource management, and traffic control. Shalev-shwartz et al. (2016) improved and optimized the safety and environmental unpredictability of autonomous driving. They demonstrated policy gradient iterations can be used without Markovian assumptions. In addition, they decomposed the problem into components of Policy for Desires and trajectory planning with hard constraints, which enabled the comfort of driving and the safety of driving respectively. Jin et al. (2018) proposed the Distributed Coordinated Multi-Agent Bidding (DCMAB) algorithm which combined the idea of clustering with the MDRL method to optimize the performance of real-time online bidding in the face of a large number of advertisers. In order to balance the competition and cooperation between advertisers, a practical distributed coordinated multi-agent bidding algorithm was proposed and

Table 2 Multi-agent deep reinforcement learning main approachs

Approachs	Value-based	Policy-based
Communication learning	RIAL and DIAL (Foerster et al. 2016)	CommNet (Sukhbaatar and Fergus 2016)
		BiCNet (Peng et al. 2017)
		ACCNET (Mao et al. 2017)
		ATOC (Jiang and Lu 2018)
Agent modeling	DRON (He et al. 2016)	SOM (Raileanu et al. 2018)
	DPIQN and DRPIQN (Hong et al. 2018)	ToMnet (Rabinowitz et al. 2018)
	Deep Bayes-ToMoP (Yang et al. 2019)	
	Deep BPR + (Zheng et al. 2018)	
	PSRO/DCH (Lanctot et al. 2017)	

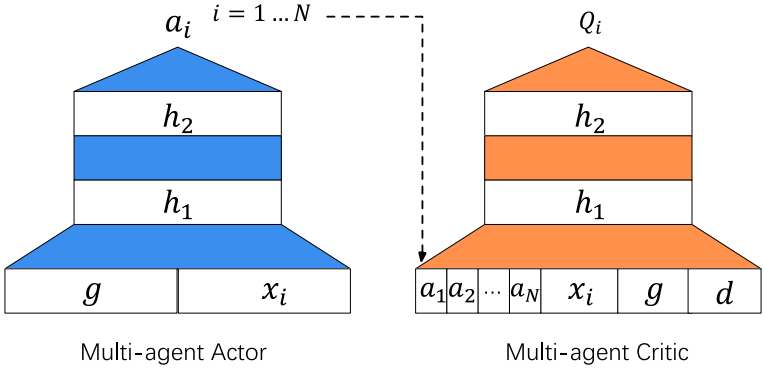

Fig.7 Structure of DCMAB algorithm

implemented. As shown in Fig. 7, the model use structure of separate actor and Q network for each agent. a_i is calculated through u_i using g and x as input. Where g represents general information and x represents the clustering feature. In addition to states and actions, consumer distribution d is collected as the input of all agents' Q function.

In the field of resource management, Xi et al. (2018) proposed a novel MDRL algorithm named PDWoLF-PHC to solve the stochastic disturbance problem of the power grid system caused by the integration of the distributed energy and new energy. It can realize the application of stochastic games in non-Markovian environments effectively. That model has a faster convergence speed and stronger robustness, so that the power grid system can improve the utilization rate of new energy under more complex conditions. Perolat et al. (2017) studied the emergent behavior of groups of independent-agents in partially observable Markov games. It modeled the occupant of common-pool resources, revealed the relationship between exclusivity, sustainability and inequality, and proposed solutions to improve resource management ability. Kofinas et al. (2018) proposed the fuzzy Q learning method to effectively improve the energy management ability of decentralized micro grid. Noureddine et al. (2017) proposed a DRL cooperative task allocation method, which enables multiple agents to interact and effectively allocate resources and tasks. In a loosely coupled distributed multi-agent setting, agents can benefit from collaborative neighbors.

In the traffic control domains, Chen et al. (2016) proposed a cooperative multi-agent reinforcement learning framework to alleviate bus congestion on bus lanes in real time. They adopted coordination graphs to automatically selecting the coordinated holding actions when multiple buses are stationed at the stops. In addition, for the particularity of the sparsely structured graphs, they developed sparse collaborative Q learning algorithms for coordinated holding actions. Simulations experiments proved the method could be applied in an advanced public transportation system to improve the performance of bus operation. Vidhate et al. (2017) proposed a traffic control model based on cooperative multi-agent reinforcement learning for the control and optimization of the traffic systems, which can deal with unknown complex states. The model extends the traffic value of the vehicle, including delay time, the newly arriving vehicles and the number of vehicles parked at a signal to learn and set the optimal actions. The model makes a great improvement to traffic control, which proves that it can realize real-time dynamic traffic control. As shown in Fig. 8, in a real-world environment, the flow of four signals with eight flows is considered. Calvo et al. (2018) proposed a novel IDQN algorithm to solve

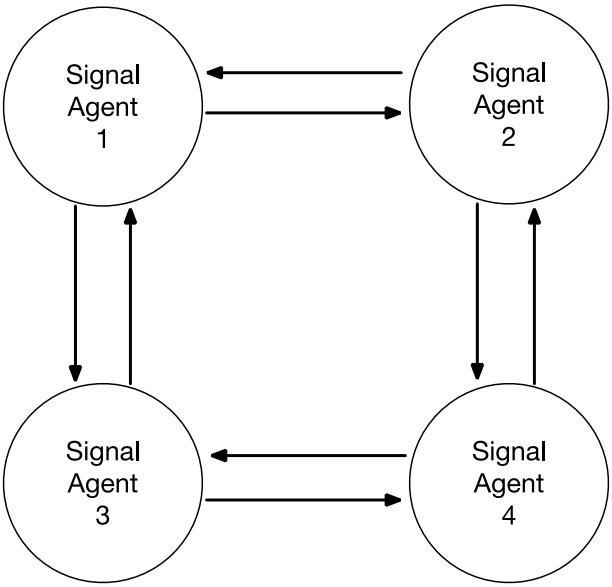

Fig. 8 Traffic flow and control of four intersections with eight flow directions

the heterogeneity problem of urban traffic signal control in multi-agent environment. Each agent learns through dueling double deep Q-network (DDDQN), which integrates dueling networks, DDQN and priority experience replay. As shown in Fig. 9, agents learn from local experience and the exchange of information between them. Agents interact in two ways: one is an intra-level way that agents interact with each other at the same level and another is an inter-level way where agents interact with each other at different hierarchy

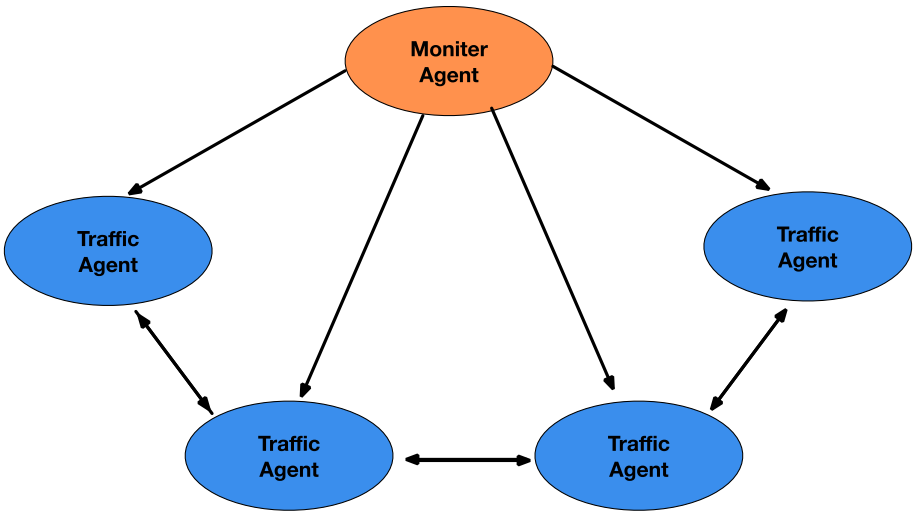

Fig. 9 Architecture of traffic signal control

levels. Traffic agents use inter-level ways to communicate with the monitor-agent. Sharing information between traffic agents is intra-level.

The agents interact in two manners: (1) intra-level or horizontal interaction; where the agents interact with each other at the same level, and (2) inter-level; where the agents interact with each other in different levels of hierarchy. Traffic agents use intra-level interaction by sharing information between them, and the communication between the monitor-agent and the traffic agents is inter-level. (Table 3).

5 Conclusion and research directions

Deep reinforcement learning has shown success in many single-agent fields, and the next step is to focus on multi-agent scenarios. However, deep reinforcement learning for multi-agent settings is fundamentally more difficult due to non-stationarity, the increase of dimensionality, the partial observability and the credit assignment problem, among other factors (Stone and Veloso 2000; Panait and Luke 2005; Hernandez-Leal et al. 2017, 2018; Albrecht and Stone 2018; Silva and Costa 2019; Palmer et al. 2019; Wei et al. 2018; Nguyen et al. 2018; Xu et al. 2016).

This paper makes a systematic review of multi-agent deep reinforcement learning, including the background, classical algorithm, research progress and practical application. In many practical problems and fields, multi-agent deep reinforcement learning has shown great potential, with endless research results and various algorithms emerging. In this paper, the theoretical background and classical algorithms of multi-agent reinforcement learning are introduced firstly, including the Markov framework and stochastic game model. Then the recent innovations and improvements of multi-agent deep reinforcement learning algorithms from different perspectives are reviewed in detail. Finally, the practical application and future prospect of multi-agent deep reinforcement learning are discussed. There are some research directions that are not detailed in this paper but are still promising, including learning from demonstration, model-free deep RL and transfer learning.

Learning from demonstration which consists of imitation learning and inverse RL has made significant progress in single-agent deep RL (Piot et al. 2016). Imitation learning attempts to map states to actions in a supervised way. It extends the expert policy directly to the unvisited state, making it closer to the multi-class classification problem in the case of finite action sets. Inverse RL agents try to infer a reward function based

Table 3 Typical MDRL applications in different fields

Application field	Based structure	Research
Autonomous driving	Policy Gradient	Shalev-shwartz et al. (Shalev-Shwartz et al. 2016)
Internet marketing	Actor-Critic	Jin et al. (Jin et al. 2018)
Resource management	PDWoLF-PHC	Xi et al. (Xi et al. 2018)
	DQN	Perolat et al. (Perolat et al. 2017)
	Fuzzy DQN	Kofinas et al. (Kofinas et al. 2018)
	IDQN	Noureddine et al. (Noureddine et al. 2017)
Traffic control	DQN	Chen et al. (Chen et al. 2016)
	DQN	Vidhate et al. (Vidhate and Kulkarni 2017)
	IDQN	Calvo et al. (Calvo and Dusparic 2018)

on expert demonstrations (Hadfield-Menell et al. 2016, 2017). However, these methods have not been fully studied in multi-agent settings. In MAS, these applications create a very direct challenge that requires multiple experts who can demonstrate tasks collaboratively. Moreover, the communication and reasoning abilities of experts are difficult to be described and modeled by autonomous agents in MAS. These raise important questions for the extensions of imitation learning and inverse RL to MDRL (Christiano et al. 2017; Nguyen et al. 2018, 2018). Zhang et al. combined human prior suboptimal knowledge with RL to introduce a knowledge guided policy network (Zhang et al. 2020). In addition, model-free deep RL has been applied to solve many complex problems in the field of single-agent and multi-agent domains. However, such methods require a large number of samples and a long learning time to achieve good performance. Model-based deep learning extensions have made great progress in single-agent field, such as (Finn and Levine 2017; Gu et al. 2016; Levine et al. 2016), but these extensions have not been widely studied in multi-agent setting.

Many studies have promoted transfer learning to improve the performance of MDRL models during training and reduce computational cost. Yin et al. (2017) introduced policy distillation framework to apply knowledge transfer to DRL. This method reduces training time and has better performance than DQN, but its exploration policy is not effective enough. Egorov et al. (2016) reconstructed the multi-agent environment as an image-like representation and used CNNs to estimate the Q value of each agent. When the transfer learning method can be used to speed up the training process, it can solve the scalability problem in MAS. Parisotto et al. (2015) proposed a role simulation method for multi-task transfer learning to improve the learning speed of deep policy networks. The network is not very complex, but it can achieve expert performance on multiple games at the same time.

Although MDRL is a recent area, there are still many open-source benchmarks that can be used for different characteristics, such as Starcraft Multi-agent Challenge, Hanabi and so on. Starcraft Multi-agent Challenge (Samvelyan et al. 2019) is based on StarCraft II which is a real-time strategy game. It focuses on the challenges of micromanagement, which means the fine-grained control of individual units. Each unit is controlled by an independent agent that acts according to local observations. Hanabi is a cooperative multiplayer card game that includes two to five players.

The game is designed that players can't see their own cards, but other players can reveal their information. This presents an interesting challenge to learning algorithms, especially in the case of self-play learning and ad-hoc teams (Bard et al. 2020; Hessel et al. 2018; Bowling and McCracken 2005). Pommerman (Resnick et al. 2018) is another multi-agent benchmark that supports partial observability and communication learning among agents. It can be used to test cooperative, competitive and mixed tasks. Pommerman is a very challenging field because the rewards are very sparse and delayed (Gao et al. 2019). The Apprentice Firemen Game (Palmer et al. 2019) and Fully Cooperative Multiagent Object Transporation Problems (CMTOPs) (Palmer et al. 2018) are both two-agent pixel-based environment. In addition to the above, there are many benchmarks such as MuJoCo Multi-agent Soccer (Liu et al. 2019), Neural MMO (Suarez et al. 1903), Arena (Song et al. 2019), MARLO competition (Johnson et al. 2016), MAgent (Zheng et al. 2017).

Acknowledgements This work is supported by the National Natural Science Foundations of China (Nos. 61672522, 61976216, and 61379101).

References

Abouheaf M, Gueaieb W (2017) Multi-agent reinforcement learning approach based on reduced value function approximations. In 2017 IEEE International Symposium on Robotics and Intelligent Sensors (IRIS) pp 111–116. IEEE

Albrecht SV, Stone P (2018) Autonomous agents modeling other agents: a comprehensive survey and open problems. Artif Intell 258:66–95

Bard N, Foerster JN, Chandar S, Burch N, Lanctot M, Song HF, Dunning I (2020) The hanabi challenge: a new frontier for ai research. Artif Intell 280:103216

Bowling M, McCracken P (2005) Coordination and adaptation in impromptu teams. In: 1995 AAAI conference on artificial intelligence, vol 5, pp 53–58

Buşoniu L, Babuška R, De Schutter B (2010) Multi-agent reinforcement learning: an overview. In: Srinivasan D, Jain LC (eds) Innovations in multi-agent systems and applications-1. Springer, Berlin, Heidelberg, pp 183–221

Calvo JA, Dusparic I (2018) Heterogeneous multi-agent deep reinforcement learning for traffic lights control. In AICS pp 2–13

Camerer CF, Ho TH, Chong JK (2004) Behavioural game theory: thinking, learning and teaching. In Advances in understanding strategic behavior. Palgrave Macmillan, London, pp 120–180

Carmel D, Markovitch S (1996) Incorporating opponent models into adversary search. In AAAI/IAAI, Vol. 1, pp 120–125

Chen W, Zhou K, Chen C (2016) Real-time bus holding control on a transit corridor based on multi-agent reinforcement learning. In 2016 IEEE 19th International conference on intelligent transportation systems (ITSC) pp 100–106. IEEE

Christiano PF, Leike J, Brown T, Martic M, Legg S, Amodei D (2017) Deep reinforcement learning from human preferences. In Advances in Neural Information Processing Systems pp 4299–4307

Da Silva FL, Costa AHR (2019) A survey on transfer learning for multiagent reinforcement learning systems. J Artif Intell Res 64:645–703

Ding S, Du W, Zhao X et al (2019) A new asynchronous reinforcement learning algorithm based on improved parallel PSO. Appl Intell 49(12):4211–4222

Duan Y, Chen X, Houthooft R, Schulman J, Abbeel P (2016) Benchmarking deep reinforcement learning for continuous control. In International Conference on Machine Learning pp 1329–1338

Egorov M (2016) Multi-agent deep reinforcement learning. CS231n: convolutional neural networks for visual recognition

Finn C, Levine S (2017) Deep visual foresight for planning robot motion. In 2017 IEEE International Conference on Robotics and Automation (ICRA) pp 2786–2793. IEEE

Foerster J, Assael IA, de Freitas N, Whiteson S (2016) Learning to communicate with deep multi-agent reinforcement learning. In Advances in Neural Information Processing Systems pp 2137–2145

Foerster J, Nardelli N, Farquhar G, Afouras T, Torr PH, Kohli P, Whiteson S (2017) Stabilising experience replay for deep multi-agent reinforcement learning. In Proceedings of the 34th International Conference on Machine Learning-Volume 70 pp 1146–1155. JMLR. org

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

Fortunato M, Azar MG, Piot B, Menick J, Osband I, Graves A, Blundell C (2017) Noisy networks for exploration. arXiv preprint

Francois-Lavet V, Fonteneau R, Ernst D (2015) How to discount deep reinforcement learning: towards new dynamic strategies. Proceedings of the Workshops at the Advances in Neural Information Processing Systems. Montreal, Canada: pp 107–116

Fu H, Tang H, Hao J, Lei Z, Chen Y, Fan C (2019) Deep multi-agent reinforcement learning with discrete-continuous hybrid action spaces. arXiv preprint

Fujimoto S, Van Hoof H, Meger D (2018) Addressing function approximation error in actor-critic methods. arXiv preprint

Gao C, Kartal B, Hernandez-Leal P, Taylor ME (2019) On hard exploration for reinforcement learning: a case study in pommerman. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment Vol. 15, No. 1, pp 24–30

Gmytrasiewicz PJ, Doshi P (2005) A framework for sequential planning in multi-agent settings. J Artif Intell Res 24:49–79

Gmytrasiewicz PJ, Durfee EH (2000) Rational coordination in multi-agent environments, autonomous agents and multi-agent systems 3 (4)

Greenwald A, Hall K, Serrano R (2003) Correlated q-learning. In: International conference on machine learning, vol 3, pp 242–249

Gu S, Lillicrap T, Sutskever I, Levine S (2016) Continuous deep q-learning with model-based acceleration. In International Conference on Machine Learning pp 2829–2838

Gu S, Holly E, Lillicrap T et al. (2017) Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates. IEEE International Conference on Robotics and Automation. Singapore: IEEE Press: 3389–3396

Gupta, J. K., Egorov, M., & Kochenderfer, M. (2017). Cooperative multi-agent control using deep reinforcement learning. In International Conference on Autonomous Agents and Multiagent Systems pp 66–83 Springer, Cham

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

Hadfield-Menell D, Russell SJ, Abbeel P, Dragan A (2016) Cooperative inverse reinforcement learning. In Advances in neural information processing systems pp 3909–3917

Hadfield-Menell D, Milli S, Abbeel P, Russell SJ, Dragan A (2017) Inverse reward design. In Advances in neural information processing systems pp 6765–6774

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

He H, Boyd-Graber J, Kwok K, Daumé III H (2016) Opponent modeling in deep reinforcement learning. In International Conference on Machine Learning pp 1804–1813

Heess N, Sriram S, Lemmon J, Merel J, Wayne G, Tassa Y, Silver D (2017) Emergence of locomotion behaviours in rich environments. arXiv preprint

Hernandez-Leal P, Kaisers M (2017) Learning against sequential opponents in repeated stochastic games. In The 3rd Multi-disciplinary Conference on Reinforcement Learning and Decision Making, Ann Arbor

Hernandez-Leal P, Taylor ME, Rosman B, Sucar LE, Munoz de Cote E (2016) Identifying and tracking switching, non-stationary opponents: a bayesian approach, In: Multiagent Interaction without Prior Coordination Workshop at AAAI, Phoenix, AZ, USA, 2016

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

Hernandez-Leal P, Zhan Y, Taylor ME, Sucar LE, de Cote EM (2017) Efficiently detecting switches against non-stationary opponents. Auton Agent Multi-Agent Syst 31(4):767–789

Hernandez-Leal P, Kartal B, Taylor ME (2018) Is multiagent deep reinforcement learning the answer or the question? A brief survey. arXiv preprint

Hessel M, Modayil J, Van Hasselt H, Schaul T, Ostrovski G (2017) Rainbow: combining improvements in deep reinforcement learning

Hessel M, Modayil J, Van Hasselt H, Schaul T, Ostrovski G, Dabney W, Silver D (2018) Rainbow: combining improvements in deep reinforcement learning. In Thirty-Second AAAI Conference on Artificial Intelligence

Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv preprint

Hong ZW, Su SY, Shann, TY, Chang YH, Lee CY (2018) A deep policy inference q-network for multi-agent systems. In Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems pp 1388–1396. International Foundation for Autonomous Agents and Multiagent Systems

Hu J, Wellman MP (2003) Nash Q-learning for general-sum stochastic games. J Mach Learn Res 4:1039–1069

Ivanov S, D'yakonov A (2019) Modern Deep Reinforcement Learning Algorithms. arXiv preprint

Jiang J, Lu Z (2018) Learning attentional communication for multi-agent cooperation. In Advances in Neural Information Processing Systems pp 7254–7264

Jin J, Song C, Li H, Gai K, Wang J, Zhang W (2018) Real-time bidding with multi-agent reinforcement learning in display advertising. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management pp 2193–2201. ACM

Johnson M, Hofmann K, Hutton T (2016) The Malmo platform for artificial intelligence experimentation. In: IJCAI, pp 4246–4247

Kofinas P, Dounis AI, Vouros GA (2018) Fuzzy Q-Learning for multi-agent decentralized energy management in microgrids. Appl Energy 219:53–67

Kononen V (2004) Asymmetric multiagent reinforcement learning. Web Intell Agent Syst: An Int J 2(2):105–121

Kurek M, Jakowski W (2016) Heterogeneous team deep Q-learning in low-dimensional multi-agent environments. In Computational Intelligence and Games (CIG), 2016 IEEE Conference on pp 1–8

Lakshminarayanan AS, Sharma S, Ravindran B (2016) Dynamic frame skip deep q network. Proceedings of the Workshops at the International Joint Conference on Artificial Intelligence

Lanctot M, Zambaldi V, Gruslys A, Lazaridou A, Tuyls K, Pérolat J, Graepel T (2017) A unified game-theoretic approach to multiagent reinforcement learning. In Advances in Neural Information Processing Systemsm pp 4190–4203

Lanctot M, Zambaldi V, Gruslys A et al (2017) A unified game-theoretic approach to multi-agent reinforcement learning. Advances in neural information processing systems. Los Angeles: NIPS Press 2017:4190–4203

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

Leibo JZ, Zambaldi V, Lanctot M, Marecki J, Graepel T (2017) Multi-agent reinforcement learning in sequential social dilemmas. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems pp 464–473. International Foundation for Autonomous Agents and Multiagent Systems

Levine S, Finn C, Darrell T, Abbeel P (2016) End-to-end training of deep visuomotor policies. J Mach Learn Res 17(1):1334–1373

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

Lillicrap TP, Hunt JJ, Pritzel A et al (2016) Continuous control with deep reinforcement learning. Comput Sci 8(6):A187

Littman ML (1994) Markov games as a framework for multi-agent reinforcement learning. New brunswick: machine learning. Elsevier, USA, pp 157–163

Littman ML (2001) Value-function reinforcement learning in Markov games. Cognit Syst Res 2(1):55–66

Liu S, Lever G, Merel J, Tunyasuvunakool S, Heess N, Graepel T (2019) Emergent coordination through competition. arXiv preprint

Lowe R, Wu Y, Tamar A, Harb J, Abbeel OP, Mordatch I (2017) Multi-agent actor-critic for mixed cooperative-competitive environments. Adv Neural Inf Process Syst 30:6379–6390

Mao H, Gong Z, Ni, Y, Xiao Z (2017) ACCNet: Actor-Coordinator-Critic Net for" Learning-to-Communicate" with Deep Multi-agent Reinforcement Learning. arXiv preprint

Mao H, Liu W, Hao J, Luo J, Li D, Zhang Z, Xiao Z (2019) Neighborhood cognition consistent multi-agent reinforcement learning. arXiv preprint

Mnih V, Kavukcuoglu K, Silver D, Graves A, Antonoglou I, Wierstra D, Riedmiller, M (2013) Playing atari with deep reinforcement learning. arXiv preprint

Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Kavukcuoglu K (2016) Asynchronous methods for deep reinforcement learning. In International conference on machine learning (pp 1928–1937

Nguyen ND, Nahavandi S, Nguyen T (2018) A human mixed strategy approach to deep reinforcement learning. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) pp 4023–4028. IEEE

Nguyen TT, Nguyen ND, Nahavandi S (2018) Deep reinforcement learning for multi-agent systems: a review of challenges, solutions and applications. arXiv preprint

Nguyen T, Nguyen ND, Nahavandi S (2018) Multi-agent deep reinforcement learning with human strategies. arXiv preprint

Noureddine D, Gharbi A Ahmed S (2017) Multi-agent deep reinforcement learning for task allocation in dynamic environment. In Proceedings of the 12th International Conference on Software Technologies (ICSOFT), pp 17–26

Palmer G, Tuyls K, Bloembergen D, Savani R (2018) Lenient multi-agent deep reinforcement learning. In Proceedings of the 17th International Conference on Autonomous Agents and Multi-Agent Systems pp 443–451. International Foundation for Autonomous Agents and Multiagent Systems

Palmer G, Savani R, Tuyls K (2019) Negative update intervals in deep multi-agent reinforcement learning. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems pp 43–51. International Foundation for Autonomous Agents and Multiagent Systems

Panait L, Luke S (2005) Cooperative multi-agent learning: The state of the art. Auton Agent Multi-Agent Syst 11(3):387–434

Parisotto E, Ba JL, Salakhutdinov R (2015) Actor-mimic: Deep multitask and transfer reinforcement learning. arXiv preprint

Peng P, Yuan Q, Wen Y, Yang Y, Tang Z, Long H, Wang J (2017) Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. arXiv preprint , 2

Perolat J, Leibo JZ, Zambaldi V, Beattie C, Tuyls K, Graepel T (2017) A multi-agent reinforcement learning model of common-pool resource appropriation. In Advances in Neural Information Processing Systems pp 3643–3652

Piot B, Geist M, Pietquin O (2016) Bridging the gap between imitation learning and inverse reinforcement learning. IEEE transactions on neural networks and learning systems 28(8):1814–1826

Rabinowitz NC, Perbet F, Song HF, Zhang C, Eslami SM, Botvinick M (2018) Machine theory of mind. arXiv preprint

Raileanu R, Denton E, Szlam A, Fergus R (2018) Modeling others using oneself in multi-agent reinforcement learning. arXiv preprint

Rashid T, Samvelyan M, De Witt CS, Farquhar G, Foerster J, Whiteson S (2018). QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. arXiv preprint

Resnick C, Eldridge W, Ha D, Britz D, Foerster J, Togelius J et al (2018) Pommerman: a multi-agent playground

Rosman B, Hawasly M, Ramamoorthy S (2016) Bayesian policy reuse. Machine Learning 104(1):99–127

Rusu AA, Colmenarejo SG, Gulcehre C, Desjardins G, Kirkpatrick J, Pascanu R, Hadsell R (2015) Policy distillation. arXiv preprint

Samvelyan M, Rashid T, Schroeder de Witt C, Farquhar G, Nardelli N, Rudner TG, Whiteson . (2019). The starcraft multi-agent challenge. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems pp 2186–2188. International Foundation for Autonomous Agents and Multiagent Systems

Schaul T, Quan J, Antonoglou I, Silver D (2015) Prioritized experience replay. arXiv preprint

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

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

Silver D, Lever G, Heess N et al (2014) Deterministic policy gradient algorithms. Proceedings of the International Conference on Machine Learning. Beijing, China: 387–395

Son K, Kim D, Kang WJ, Hostallero DE, Yi Y (2019) Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. arXiv preprint

Song J, Ren H, Sadigh D, Ermon S (2018) Multi-agent generative adversarial imitation learning. In Advances in Neural Information Processing Systems pp 7461–7472

Song Y, Wang J, Lukasiewicz T, Xu Z, Xu M, Ding Z, Wu L (2019) Arena: a general evaluation platform and building toolkit for multi-agent intelligence. arXiv preprint

Stone P, Veloso M (2000) Multiagent systems: a survey from a machine learning perspective. Auton Robots 8(3):345–383

Suarez J, Du Y, Isola P, Mordatch I, MMO N (1903) A massively multiagent game environment for training and evaluating intelligent agents. arXiv preprint

Sukhbaatar S, Fergus R (2016) Learning multiagent communication with backpropagation. In Advances in neural information processing systems pp 2244–2252

Sunehag P, Lever G, Gruslys A, Czarnecki WM, Zambaldi V, Jaderberg M, Graepel T (2017) Value-decomposition networks for cooperative multi-agent learning. arXiv preprint

Tampuu A, Matiisen T, Kodelja D, Kuzovkin I, Korjus K, Aru J, Vicente R (2017) Multiagent cooperation and competition with deep reinforcement learning. PLoS ONE 12(4):e0172395

Tan M (1993) Multi-agent reinforcement learning: Independent vs. cooperative agents. In Proceedings of the tenth international conference on machine learning pp 330–337

Tumer K, Agogino A (2007) Distributed agent-based air traffic flow management. In Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems pp 1–8

Van Hasselt H, Guez A, Silver D (2016) Deep reinforcement learning with double q-learning. In Thirtieth AAAI conference on artificial intelligence

Vidhate DA, Kulkarni P (2017) Cooperative multi-agent reinforcement learning models (CMRLM) for intelligent traffic control. In 2017 1st International Conference on Intelligent Systems and Information Management (ICISIM) pp 325–331. IEEE

Wai HT, Yang Z, Wang PZ, Hong M (2018) Multi-agent reinforcement learning via double averaging primal-dual optimization. In Advances in Neural Information Processing Systems pp 9649–9660

Wang Z, Schaul T, Hessel M, Van Hasselt H, Lanctot M, De Freitas N (2015) Dueling network architectures for deep reinforcement learning. arXiv preprint

Wang W, Yang T, Liu Y, Hao J, Hao X, Hu Y, Gao Y (2019) From Few to More: Large-scale Dynamic Multiagent Curriculum Learning. arXiv preprint

Wang W, Liu TYY, Hao J, Hao X, Hu Y, Chen Y, Gao Y (2019) Action semantics network: Considering the Effects of Actions in Multiagent Systems. arXiv preprint

Wei E, Wicke D, Freelan D, Luke S (2018) Multiagent soft q-learning. In 2018 AAAI Spring Symposium Series

Xi L, Yu T, Yang B, Zhang X (2015) A novel multi-agent decentralized win or learn fast policy hill-climbing with eligibility trace algorithm for smart generation control of interconnected complex power grids. Energy Convers Manage 103:82–93

Xi L, Chen J, Huang Y, Xu Y, Liu L, Zhou Y, Li Y (2018) Smart generation control based on multi-agent reinforcement learning with the idea of the time tunnel. Energy 153:977–987

Xi L, Yu L, Xu Y, Wang S, Chen X (2019) A novel multi-agent DDQN-AD method-based distributed strategy for automatic generation control of integrated energy systems. IEEE Transactions on Sustainable Energy

Xu D, Si J, Bian W (2016) Fingerprint orientation field extraction using gradient-based weighted averaging. International Journal of collaborative intelligence 1(4):287–297

Yang T, Hao J, Meng Z, Zhang C, Zheng YZZ, Zheng Z (2019) Towards efficient detection and optimal response against sophisticated opponents. In Proceedings of the 28th International Joint Conference on Artificial Intelligence pp 623–629. AAAI Press

Yang Y, Hao J, Liao B, Shao K, Chen G, Liu W, Tang H (2020) Qatten: a general framework for cooperative multiagent reinforcement learning. arXiv preprint .

Yang Y, Hao J, Chen G, Tang H, Chen Y, Hu Y, Wei Z (2020) Q-value path decomposition for deep multiagent reinforcement learning. In International Joint Conference on Artificial Intelligence (IJCAI)

Yin H, Pan SJ (2017) Knowledge transfer for deep reinforcement learning with hierarchical experience replay. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence

Zhang P, Hao J, Wang W, Tang H, Ma Y, Duan Y, Zheng Y (2020) KoGuN: accelerating deep reinforcement learning via integrating human suboptimal knowledge. In Thirty-seventh International Conference on Machine Learning (ICML)s

Zhao Z, Gao Y, Luo B et al (2004) Reinforcement learning technology in multi-agent system. Comput Sci 31(3):23–27

Zhao X, Ding S, An Y, Jia W (2018) Asynchronous reinforcement learning algorithms for solving discrete space path planning problems. Appl Intell 48(12):4889–4904

Zhao X, Ding S, An Y, Jia W (2019) Applications of asynchronous deep reinforcement learning based on dynamic updating weights. Appl Intell 49(2):581–591

Zheng L, Yang J, Cai H, Zhang W, Wang J, Yu Y (2017)s Magent: a many-agent reinforcement learning platform for artificial collective intelligence

Zheng Y, Meng Z, Hao J, Zhang Z, Yang T, Fan C (2018) A deep bayesian policy reuse approach against non-stationary agents. In Advances in Neural Information Processing Systems pp 954–964