Multi-Agent Reinforcement Learning for Problems with Combined Individual and Team Reward

Hassam Ullah Sheikh
Department of Computer Science
University of Central Florida
Orlando, USA
hassam.sheikh@knights.ucf.edu

Ladislau Bölöni
Department of Computer Science
University of Central Florida
Orlando, USA
lboloni@cs.ucf.edu

***Abstract*—Many cooperative multi-agent problems require agents to learn individual tasks while contributing to the collective success of the group. This is a challenging task for current state-of-the-art multi-agent reinforcement algorithms that are designed to maximize either the global reward of the team or the individual local rewards. The problem is exacerbated when either of the rewards is sparse leading to unstable learning. To address this problem, we present *Decomposed Multi-Agent Deep Deterministic Policy Gradient (DE-MADDPG)*: a novel cooperative multi-agent reinforcement learning framework that simultaneously learns to maximize the global and local rewards. We evaluate our solution on the defensive escort team problem and show that our solution achieves better and more stable performance than the direct adaptation of the MADDPG algorithm.**

***Index Terms*—Multi-Agent Reinforcement Learning; Coordination and Collaboration; Dual-Reward Learning**

I. Introduction

Cooperative multi-agent problems are prevalent in real-world settings such as strategic conflict resolution [1] and collaboration of agents in defensive escort teams [2]. Such problems can be modelled as dual-interest: each agent is simultaneously working towards maximizing its own payoff (local reward) as well as the collective success of the team (global reward). For example, autonomous vehicles in double-lane merge conflicts must perform cooperative maneuvers without diverging from their destination-bound nominal trajectories. Similarly, in the case of a defensive escort team, each agent has to maintain a specific distance from the payload to avoid disrupting any social norms without sacrificing the security of the payload. Despite the recent success of multi-agent reinforcement learning (MARL) in multiplayer games like Dota 2 [3], Quake III Capture-the-Flag [4] and Starcraft II [5] or learning to use tools [6], learning multi-agent cooperation while simultaneously maximizing local rewards is still an open challenge. In this learning problem, to which we will refer as "**dual-reward MARL**", the agents are *explicitly* receiving two reward signals: the global team reward and the agent's individual local reward.

Current state-of-the-art MARL algorithms can be categorized in two types. For algorithms such as COMA [7] and QMIX [8], the goal is to maximize the global reward for the success of the group while algorithms such MADDPG [9] and M3DDPG [10] focus on optimizing local rewards without any explicit notion of coordination. As shown in [11] and in our findings in Section V, a direct adaptation of these algorithms to dual-reward problems often leads to poor performance and unstable learning. Generally, these adaptations happen in the reward function space where the local and the global reward signals are combined to form an entangled multi-objective reward function [12]. This coupling of reward functions leads to two problems. First, the entangled reward function becomes unfactorizable during training, causing the learning to oscillate between optimizing either the global or the local reward leading to a sub-optimal and unstable solution. This problem is exacerbated when one of the rewards is sparse, which leads to a bias in favor of the other. The second problem is that maximizing the entangled reward function does not correspond to maximizing the objective function.

To address these issues, we present *Decomposed Multi-Agent Deep Deterministic Policy Gradient (DE-MADDPG)*: a novel cooperative multi-agent reinforcement learning framework built on top of deterministic policy gradients that simultaneously learns to maximize the global and the local rewards without the need of creating an entangled multi-objective reward function. The core idea behind DE-MADDPG is to train two critics. The *global critic*, shared between all the cooperating agents takes as input the observations and actions of these agents and estimates the sum of the global expected rewards. The *local critic* receives as input only the observation and action of the particular agent and estimates the sum of local expected rewards.

To summarize, our contributions are following:

- We develop a dual-critic framework for multi-agent reinforcement learning that learns to simultaneously maximize the decomposed global and local rewards.
- Taking advantage of the decomposition, we treat the global critic as a single-agent critic. This allows us to apply performance enhancement techniques such as Prioritized Experience Replay (PER) [13] and Twin Delayed Deep Deterministic Policy Gradients (TD3) [14] to tackle the overestimation bias problem in Q-functions. This was not previously feasible in the multi-agent RL setting.
- We evaluate our proposed solution on the defensive

This work had been supported in part by the National Science Foundation under grant number IIS-1409823.

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

escort team problem [2], [15] (see Figure 2) and show that it achieves a significantly better and more stable performance than the direct adaptation of the MADDPG algorithm.

II. Related work

Early theoretical work in MARL was limited to discrete state and action spaces [16], [17], [18]. Recent works adopted techniques from single-agent deep RL to develop general algorithms for high-dimensional continuous space environments requiring agent interactions [1], [19], [9].

Cooperative multi-agent learning is important since many real-world problems can be formulated as distributed systems with decentralized agents that must coordinate to achieve shared objectives [20]. Similar to our work, [21] have shown that agents whose rewards depend on all agents' success perform better than agents that optimize for their own success. In the special case when all agents have a single goal and share a global reward, COMA [7] uses a counterfactual baseline. However, the centralized critic in these methods only focuses on optimizing the collective success of the group. When a global objective is the sum of agents' individual objectives, value-decomposition methods optimize a centralized Q-function while preserving scalable decentralized execution [8], but do not address credit assignment. While MADDPG [9] and M3DDPG [10] apply to agents with different reward functions, they do not specifically address the need for cooperation; in fact, they do not distinguish between the problems of cooperation and competition.

To the best of our knowledge, dual-reward MARL was not explicitly addressed in the existing literature. Among papers exploring related topics, [22] explored the multi-goal problem and analyzed its convergence in a special networked setting restricted to fully-decentralized training. In contrast, we are conducting centralized training with decentralized execution. In contrast to multi-*task* MARL, which aims for generalization among *non-simultaneous* tasks [23], and in contrast to hierarchical methods with top-level managers that *sequentially* select subtasks [24], our decentralized agents must cooperate *in parallel* to achieve the global and their local objectives.

III. Background

A. Policy Gradients

Policy gradient methods have been shown to learn the optimal policy in a variety of reinforcement learning tasks. The main idea behind policy gradient methods is to directly parameterize the policy using the parameters θ to maximize the objective represented as $J(\theta) = \mathbb{E}[\mathbf{R}^t]$ by taking a step in the direction

$$\nabla J(\theta) = \mathbb{E}\left[\nabla_\theta \log \pi_\theta(a|s)\, Q^\pi(s,a)\right]$$

Policy gradient methods are prone to the high variance problem. Several methods such as [25], [26] have been shown to reduce the variability by introducing a *critic*, a Q-function that tells about the goodness of a reward by working as a baseline. [27] has shown that it is possible to extend the policy gradient framework to deterministic policies *i.e.* $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$. In particular we can write $\nabla J(\theta)$ as

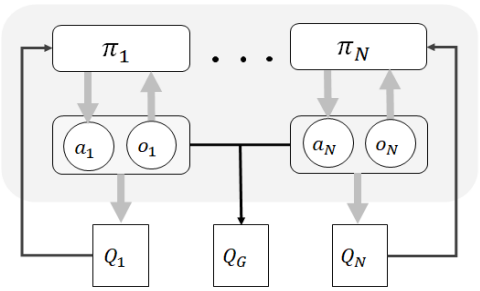


Fig. 1: An overview of the Decomposed Multi-Agent Deep Deterministic Policy Gradient architecture. In contrast to MADDPG which uses a single centralized critic, DE-MADDPG has a global centralized critic shared between all cooperating agents and a local critic specific to the agent.

$$\nabla J(\theta) = \mathbb{E}\left[\nabla_\theta \pi(a|s)\, \nabla_a Q^\pi(s,a)\,|_{a=\pi(s)}\right]$$

A variation of this model, Deep Deterministic Policy Gradients (DDPG) [28] is an off-policy algorithm that approximates the policy π and the critic Q^π with deep neural networks. DDPG uses an experience replay buffer alongside a target network to stabilize the training. Twin Delayed Deep Deterministic Policy Gradients (TD3) [14] improves on DDPG by addressing the overestimation bias of the Q-function, similarly to Double Q-learning [29]. They find that approximation errors of the neural network, combined with gradient descent make DDPG tend to overestimate the Q-values, leading to a slower convergence. TD3 addresses this by using two Q-networks Q_{ψ_1}, Q_{ψ_2}, along with two target networks. The Q-functions are updated with the target $y = r^t + \gamma \min_{1,2} Q_{\psi'_i}(s', a')$, while updating the policy with Q_{ψ_1}. Additionally, they introduce target policy smoothing by adding noise in the determination of the next action for the critic target $a' = \mu_{\theta'_\pi}(s') + \epsilon$, with ϵ being clipped Gaussian noise $\epsilon = \texttt{clip}(\mathcal{N}(0,\sigma), -c, c)$, where c is a tunable parameter. Additionally, they use delayed policy updates and only update the policy π and target network parameters once every d critic updates.

Multi-agent deep deterministic policy gradients (MADDPG) [9] extends DDPG for the multi-agent setting where each agent has it's own policy. The gradient of each policy is written as

$$\nabla J(\theta_i) = \mathbb{E}\left[\nabla_{\theta_i} \pi_i(a_i|o_i)\, \nabla_{a_i} Q_i^\pi(s, a_1, \ldots, a_N)\,|_{a_i=\pi_i(o_i)}\right]$$

where $s = (o_1, \ldots, o_N)$ and $Q_i^\pi(s, a_1, \ldots, a_N)$ is a centralized action-value function that takes the actions of all the agents in addition to the state of the environment to estimate the Q-value for agent i. Since every agent has its own Q-function, the model allows the agents to have different action space and reward functions. The insight behind MADDPG is that knowing all the actions of other agents makes the environment stationary, even though their policy changes.

IV. Decomposed Multi-Agent Deep Deterministic Policy Gradient

We propose *Decomposed Multi-Agent Deep Deterministic Policy Gradient*: a multi-agent deep reinforcement learning algorithm that learns to simultaneously maximize the group's global reward and the agent's local rewards. Our approach uses a two critic approach to train policies and value functions that are optimal to maximize the global and local rewards.

The main idea is to combine MADDPG (or MATD3) for maximizing global rewards with a standard single agent DDPG (or TD3). Intuitively, the goal is to move the policy in the direction that maximizes both the global and the local critic. The resulting learning paradigm is similar to the centralized training with decentralized execution during testing used by [9]. In this setting, additional information is provided for the agents during training that is not available during test time.

Concretely, we consider an environment with N agents with policies $\pi = \{\pi_1, \ldots, \pi_N\}$ parameterized by $\theta = \{\theta_1, \ldots, \theta_N\}$. The *multi-agent deep deterministic policy gradient* for agent i can written as

$$\nabla J(\theta_i) = \mathbb{E}\left[\nabla_{\theta_i} \pi_i\left(a_i|o_i\right) \nabla_{a_i} Q_i^{\pi}\left(s, a_1, \ldots, a_N\right)|_{a_i=\pi_i(o_i)}\right]$$

where $s = (o_1, \ldots, o_N)$ and $Q_i^{\pi}\left(s, a_1, \ldots, a_N\right)$ is a centralized action-value function parameterized by ϕ_i that takes the actions of all the agents in addition to the state of the environment to estimate the Q-value for agent i. We extend the idea of MADDPG by introducing a local critic. Now the modified policy gradient for each agent i can be written as

$$\begin{aligned}\nabla J(\theta_i) = \overbrace{\mathbb{E}_{s,a\sim\mathcal{D}}\left[\nabla_{\theta_i} \pi_i\left(a_i|o_i\right) \nabla_{a_i} Q_{\psi}^{g}\left(s, a_1, \ldots, a_N\right)\right]}^{MADDPG} \\ + \mathbb{E}_{o_i,a_i\sim\mathcal{D}}\left[\nabla_{\theta_i} \pi_i\left(a_i|o_i\right) \nabla_{a_i} Q_i^{\pi}\left(o_i, a_i\right)\right]\Bigg\} DDPG\end{aligned} \tag{1}$$

where $a_i = \pi_i\left(o_i\right)$ is action from agent i following policy π_i and $\mathcal{D}$ is the experience replay buffer. The global critic is Q_{ψ}^{g} is updated as:

$$\mathcal{L}\left(\psi\right) = \mathbb{E}_{s,a,r,s'}\left[\left(Q_{\psi}^{g}\left(s, a_1, \ldots, a_N\right) - y_g\right)^2\right]$$

where y_g is defined as:

$$y_g = r_g + \gamma Q_{\psi'}^{g}\left(s', a'_1, \ldots, a'_N\right)|_{a'_i=\pi'_i(o'_i)}$$

where $\pi' = \{\pi'_1, \ldots, \pi'_N\}$ are target policies parameterized by $\theta' = \{\theta'_1, \ldots, \theta'_N\}$. Similarly, The local critic is Q_i^{π} is updated as:

$$\mathcal{L}\left(\phi_i\right) = \mathbb{E}_{o,a,r,o'}\left[\left(Q_i^{\pi}\left(o_i, a_i\right) - y_l\right)^2\right]$$

where y_l is defined as:

$$y_l = r_l^i + \gamma Q_{\phi'_i}^{\pi'}\left(o'_i, a'_i\right)|_{a'_i=\pi'_i(o'_i)}$$

Overestimation bias in Q-functions have been thoroughly studied in [14], [30]. This overestimation bias can be problemsome in multi-agent settings especially in real time autonomous systems. For example, in resolving the double lane merge conflict in autonomous vehicles, the vehicles might consider the current state to be near conflict resolution thus taking a dangerous turn. To solve this problem [14] have proposed a double critic approach to minimize the overestimation bias. Motivated from the results in [14], we replace the Multi-Agent Deterministic Policy Gradient of the global critic in Equation (1) with Twin Delayed Deterministic Policy Gradient. Therefore our updated policy gradient becomes

$$\begin{aligned}\nabla J(\theta_i) = \mathbb{E}_{s,a\sim\mathcal{D}}\left[\nabla_{\theta_i} \pi_i\left(a_i|o_i\right) \nabla_{a_i} Q_{\psi_1}^{g_1}\left(s, a_1, \ldots, a_N\right)\right] \\ + \mathbb{E}_{o_i,a_i\sim\mathcal{D}}\left[\nabla_{\theta_i} \pi_i\left(a_i|o_i\right) \nabla_{a_i} Q_i^{\pi}\left(o_i, a_i\right)\right]\end{aligned} \tag{2}$$

The twin global critics are updated as

$$\mathcal{L}\left(\psi_i\right) = \mathbb{E}_{s,a,r,s'}\left[\left(Q_{\psi_i}^{g_i}\left(s, a_1, \ldots, a_N\right) - y_g\right)^2\right]$$

where y_g is defined as:

$$y_g = r_g + \gamma \min_{i=1,2} Q_{\psi'_i}^{g_i}\left(s', a'_1, \ldots, a'_N\right)|_{a'_i=\pi'_i(o'_i)}$$

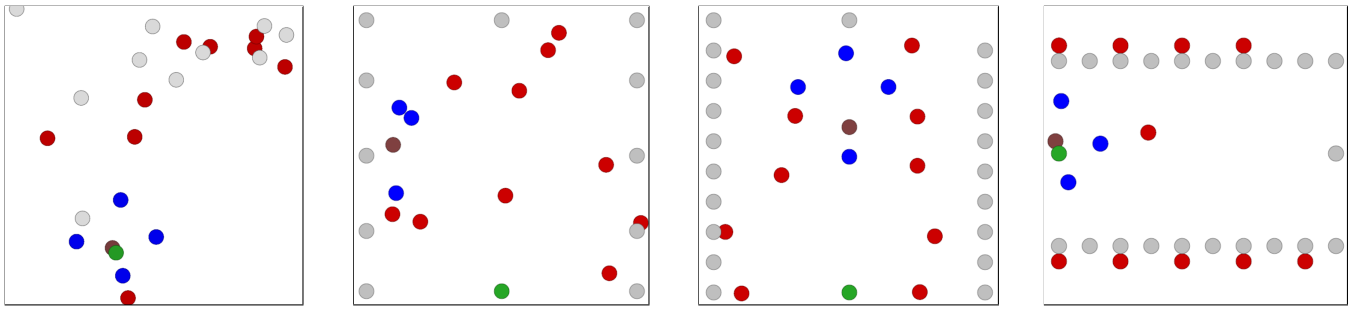
Similarly, the local critics can be updated using TD3 update style but for simplicity, we will use the standard DDPG to update the local critics. The overall algorithm to which we refer as *Decomposed Multi-Agent Deterministic Policy Gradient (DE-MADDPG)* is described in Algorithm 1. The overview of the architecture can be seen in Figure 1.

V. Experiments

A. Environments

We perform our experiments using the defensive escort problem on four VIP protection environments [15], [2]. This is a medium-size collaborative problem where a defensive escort team of agents is learning to maintain an optimal formation around the VIP (payload). The objective of the defenders is to minimize the potential physical attacks as the VIP is moving in a variety of different real world scenarios and are implemented in the Multi-Agent Particle Environment [19]. An illustration of the environments can be seen in Figure 2.

The state of the environment is given by the locations of the landmarks, bystanders, VIP and the defensive team. To closely represent a real world bodyguard that has a limited range of perception, the observation of each agent is the relative physical state of the nearest M bystanders, the VIP and the remaining members of the defensive escort team and represented as $o_i = [x_{j,\ldots N+M}] \in \mathcal{O}_i$ where x_j is the observation of the entity j from the perspective of agent i. In our experiments, we used $M = 5$.



(a) Random Landmarks (b) Shopping Mall (c) Street (d) Pie-in-the-Face

Fig. 2: Four environments for the defensive escort team problem [15]. The team of bodyguards (blue) need to protect the VIP (brown) in the environments, from left to right: "Random Landmarks", "Shopping Mall", "Street" and "Pie-in-the-Face".

Decomposing the Reward Function

In this section, we review the entangled multi-objective reward function defined in [12], [2], explain the problems with it and decompose it to be used by DE-MADDPG.

As mentioned in the the previous section that the goal of the defensive escort team is to learn an optimal formation around the VIP to minimize the physical threat while simultaneously maintain a certain distance from the VIP to follow the social norms. To achieve both of these objectives, the multi-objective reward function for each agent i is defined as:

$$r_{total} = \overbrace{\alpha\left(-1+\prod_{k=1}^{M}\left(1-RT\left(VIP,b_k,R\right)\right)\right)}^{r_{global}} + (1-\alpha)\underbrace{\left(\mathcal{D}\left(VIP,x_i\right)\right)}_{r_{local}} \tag{3}$$

where r_{global} represents the reward that each agent receives given the formation of the team around the VIP at time-step t, therefore, represents the main objective of the team and r_{local} represents the reward that the agent receives for maintaining a certain distance from the VIP and is defined as

$$\mathcal{D}\left(VIP,x_i\right) = \begin{cases} 0 & m \leq \|x_i - VIP\|_2 \leq d \\ -1 & \text{otherwise} \end{cases} \tag{4}$$

where m is the minimum distance the agent has to maintain from VIP and d is the safe distance. This formulation of r_{total} raises two problems. The first problem is the stability issue since the policy oscillates between optimizing the global reward and the local reward. This stability problem exacerbates when either of the rewards are sparse and the other reward is dense. The second problem is the α hyper-parameter. The α hyper-parameter assigns weight to both rewards. Given the various difficulty settings of the simulations, security should be prioritized in some simulations as compared to the others. Moreover, as MARL experiments take substantial amount of time, finding an optimal α can be time consuming.

We solve this problem by having a dual critic architecture where the task of the global centralized critic is to approximate the cumulative global reward while each agent's local critic approximates its own local reward. In this particular scenario, Q^g_ψ learns to approximate r_{global} while $Q^\pi_{\theta_i}$ approximates r_{local}. The benefit of this decomposition is more stable learning and exclusion of the α hyper-parameter.

B. Evaluations

To evaluate the efficacy of DE-MADDPG and its variants we compare our results with baseline MADDPG and DDPG on the defensive escort team problem in the environments described above. We trained the global critic with two different approaches. In the first approach, we updated the policy by using the standard Multi-Agent Deep Deterministic Policy Gradient mentioned in Equation (1) while in the second approach, we updated the policy using the Twin Delayed Deep Deterministic Policy Gradient mentioned in Equation (2). Additionally, to mitigate the global sparse reward problem, we replace the standard replay buffer with prioritized experience replay buffer (PER). We performed our experiments on 8 different seeds. We used neural networks with three layers for both the critic and actor networks. For each environment, we trained each approach for $20,000$ episodes except the *Pie-in-the-Face* environment which was trained for $10,000$ episodes.

Figure 3 shows the learning curves of our experiments. Figure 3a corresponds to the *Random Landmark* environment and it can be seen that the DE-MADDPG based approaches outperform MADDPG and DDPG by a significant margin. Similar observations can be made for the *Shopping Mall*, and *Street* environments. For the *Pie-in-the-Face*, there is little difference between the performance of the different approaches - all approaches converge very quickly. A possible reason for this is that this environment, focusing on a single attacking bystander, makes the positioning choice less complex.

Though, Figure 3 shows the learning curves that visually represents the performance of the different approaches, it does not quantitatively explain the improvement in performance across different approaches. To that end, we test our trained policies across all environments on 8 different seeds. Table I shows the average returns of the global reward over 1000 episodes. We notice that in complex environments such as *Shopping Mall* and *Random Landmarks*, DE-MADDPG aug-

Algorithm 1 Decomposed Multi-agent Deep Policy Gradient

Initialize main global critic networks $Q_{\psi}^{g_1}$ and $Q_{\psi}^{g_2}$.
Initialize target global critic networks $Q_{\psi'}^{g_1}$ and $Q_{\psi'}^{g_2}$.
Initialize each agents policy and critic networks.
for episode = 1 to T **do**
  for t = 1 to episode–length **do**
    Get environment state s^t.
    For each agent i, select action $a_i^t = \pi_{\theta_i}(o_i^t)$
    Execute actions $\mathbf{a}^t = [a_1^t, \ldots, a_N^t]$
    Receive global r_g^t and local rewards $\mathbf{r}_l^t$.
    Store $\left(s^t, \mathbf{a}^t, \mathbf{r}_l^t, r_g^t, s^{t+1}\right)$ in replay buffer.
  end for
  /* Train global critic*/
  Sample minibatch of size S $\left(\mathbf{s}^j, \mathbf{a}^j, \mathbf{r}_g^j, \mathbf{s}'^j\right)$ from buffer.
  $(a'_1, \ldots, a'_N) := \left(\pi'_{\theta_i}(o_i'^j), \ldots, \pi'_{\theta_N}(o_N'^j)\right)$
  Set $y_g^j = r_g^j + \gamma \min_{i=1,2} Q_{\psi'_i}^{g_i}\left(s'^j, a'_1, \ldots, a'_N\right)$
  Update global critics by minimizing

$$\frac{1}{S}\sum_j \left(y_g^j - Q_{\psi_i}^{g_i}\left(s^j, a_1^j, \ldots, a_N^j\right)\right)^2$$

  Update target network parameters

$$\psi'_i \leftarrow \tau\psi_i + (1-\tau)\,\psi'_i$$

  if episode $\mathrm{mod}\ d$ **then**
    /* Train local critics and update agent policies*/
    for agent $i = 1$ to N **do**
      Sample minibatch of size S $\left(\mathbf{s}^j, \mathbf{a}^j, \mathbf{r}_l^j, \mathbf{s}'^j\right)$
      Set $y^j = r_{il}^j + \gamma Q_{\phi'_i}^{\pi'}\left(o'^j, \pi'_{\theta_i}(o_i'^j)\right)$
      Update local critic by minimizing

$$\frac{1}{S}\sum_j \left(y^j - Q_{\phi_i}^{\pi}\left(o^j, a_i^j\right)\right)^2$$

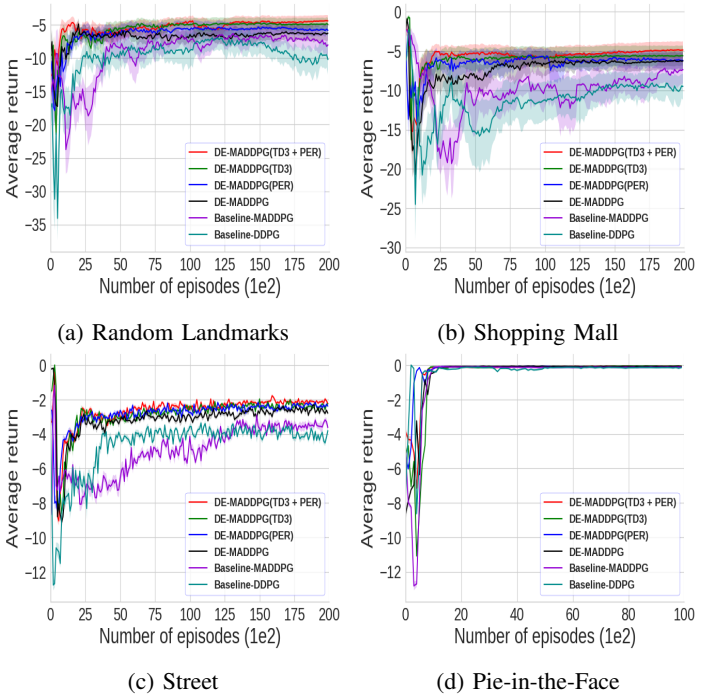
$$\theta_i = \theta_i + \frac{1}{S}\sum_j \nabla_{\theta_i}\pi_i\left(a_i|o_i^j\right)\nabla_{a_i}Q_{\psi}^{g_1}\left(s^j, a_1^j, \ldots, a_N^j\right) + \nabla_{\theta_i}\pi_i\left(a_i|o_i^j\right)\nabla_{a_i}Q_{\phi_i}^{\pi}\left(o^j, a_i^j\right)$$

    end for
    Update target network parameters for each agent i

$$\theta'_i \leftarrow \tau\theta_i + (1-\tau)\,\theta'_i$$
$$\phi'_i \leftarrow \tau\phi_i + (1-\tau)\,\phi'_i$$

  end if
end for

mented with TD3 and PER was able to achieve 55% and 73% better performance than baseline MADDPG. Similarly, on the slightly less complex environments *Street* and *Pie-in-the-Face*, the performance was about 61% and 100% better.



(a) Random Landmarks (b) Shopping Mall

(c) Street (d) Pie-in-the-Face

Fig. 3: Learning curves representing the average cumulative global reward. The higher reward represents higher protection to the VIP (payload).

TABLE I: Average cumulative return of the global reward over 1000 episodes over 8 seeds. Maximum value for each task is bolded. ± corresponds to 95% confidence interval over seeds.

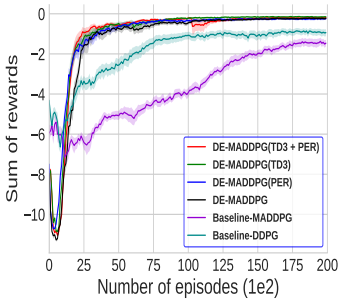
Environment	DE-MADDPG (TD3+PER)	DE-MADDPG (TD3)	DE-MADDPG (PER)	DE-MADDPG	MADDPG	DDPG
Shopping Mall	**-4.87 ± 0.06**	-5.61 ± 0.08	-6.17 ± 0.08	-6.27 ± 0.07	-7.59± 0.13	-9.51± 0.12
Rand. Landm.	**-4.43 ± 0.05**	-4.91 ± 0.06	-5.75 ± 0.04	-6.34 ± 0.08	-7.67± 0.13	-10.22± 0.16
Street	**-2.13 ± 0.06**	-2.38 ± 0.07	-2.36 ± 0.07	-2.66 ± 0.08	-3.43± 0.13	-3.81± 0.11
Pie-in-the-Face	-0.07 ± 0.002	**-0.06 ± 0.002**	-0.11 ± 0.003	-0.07 ± 0.004	-0.12± 0.003	-0.10± 0.006

Figure 4 and Table II shows the learning curves and the test results of the sum of the local rewards. Similar to the global reward, it can be seen in Figure 4 and and Table II that DE-MADDPG based approaches outperforms MADDPG and DDPG results. One point to be noted here is that maintaining a certain distance from a moving payload is a fairly easy problem for standard reinforcement learning algorithms as the reward is dense and the state space is relatively easy. However, adding an additional objective such as global reward maximization not only had catastrophic affect on the global reward maximization but also negatively effected the learning of a trivial task.

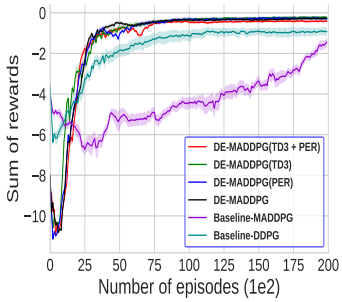
TABLE II: Average cumulative return of the local reward over 1000 episodes over 8 seeds. Maximum value for each task is bolded. ± corresponds to 95% confidence interval over seeds.

Environment	DE-MADDPG (TD3+PER)	DE-MADDPG (TD3)	DE-MADDPG (PER)	DE-MADDPG	MADDPG	DDPG
Shopping Mall	-0.41 ± 0.02	**-0.23 ± 0.03**	-0.28 ± 0.03	-0.30 ± 0.07	-1.44± 0.07	-0.92± 0.05
Rand. Landm.	**-0.15 ± 0.02**	-0.19 ± 0.01	-0.25 ± 0.02	-0.22 ± 0.03	-1.46± 0.06	-0.94± 0.05
Street	**-0.12 ± 0.01**	-0.15 ± 0.02	-0.16 ± 0.02	-0.18 ± 0.08	-1.23± 0.08	-0.47± 0.04
Pie-in-the-Face	**-0.11 ± 0.01**	-0.28 ± 0.01	-0.12 ± 0.01	-0.31 ± 0.01	-0.20± 0.01	-0.31± 0.01

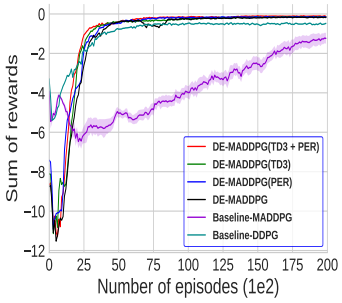
A common pattern can be seen in Figure 3 and Figure 4:



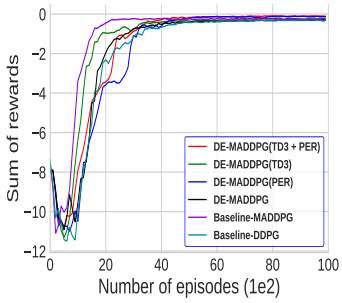
(a) Random Landmarks



(b) Shopping Mall



(c) Street



(d) Pie-in-the-Face

Fig. 4: Learning curves of the experiments representing the sum of local rewards. Notice that similar to global rewards, the local reward learning of MADDPG and DDPG is also unstable and reaches only at sub-optimal performance.

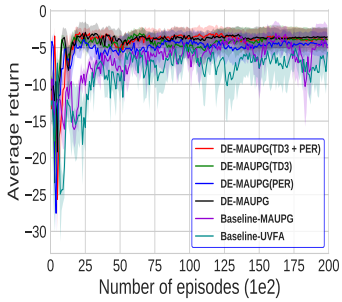
TABLE III: Average cumulative return of the global reward of multi-scenario learning over 1000 episodes over 8 seeds. Maximum value for each task is bolded. $\pm$ corresponds to 95% confidence interval over seeds.

Environment	DE-MAUPG (TD3+PER)	DE-MAUPG (TD3)	DE-MAUPG (PER)	DE-MAUPG	MAUPG	UVFA
Shopping Mall	**-3.85 $\pm$ 0.07**	-4.34 $\pm$ 0.08	-5.70 $\pm$ 0.09	-5.98 $\pm$ 0.10	-6.97$\pm$ 0.11	-6.94$\pm$ 0.11
Rand. Landm.	-3.91 $\pm$ 0.07	-3.86 $\pm$ 0.07	-4.65 $\pm$ 0.08	**-3.54 $\pm$ 0.08**	-5.02$\pm$ 0.14	-5.62$\pm$ 0.11
Street	**-2.54$\pm$ 0.08**	-3.22 $\pm$ 0.09	-3.58 $\pm$ 0.11	-3.51 $\pm$ 0.10	-3.97$\pm$ 0.14	-4.34$\pm$ 0.14
Pie-in-the-Face	**-0.07 $\pm$ 0.003**	-0.13$\pm$ 0.008	-0.08 $\pm$ 0.003	**-0.07 $\pm$ 0.002**	-0.12$\pm$ 0.003	-0.19$\pm$ 0.008

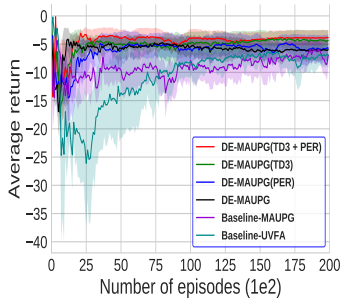
DE-MADDPG based approaches not only achieve higher global and local rewards but they achieve it significantly faster compared to MADDPG and DDPG. For example, in complex environments such as *Random Landmarks* and *Shopping Mall*, DE-MADDPG (TD3 + PER) reaches the maximum performance before 2500 episodes while the baseline MADDPG reaches its best performance for *Random Landmarks* and *Shopping Mall* environment at around $12,000$ episodes and $20,000$ episodes respectively.

TABLE IV: Average cumulative return of the local reward of multi-scenario learning over 1000 episodes over 8 seeds. Maximum value for each task is bolded. $\pm$ corresponds to 95% confidence interval over seeds.

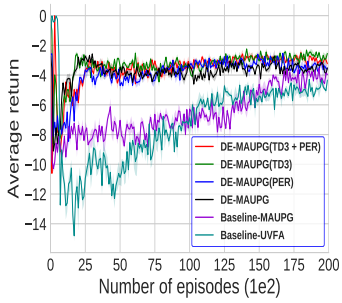
Environment	DE-MAUPG (TD3+PER)	DE-MAUPG (TD3)	DE-MAUPG (PER)	DE-MAUPG	MAUPG	UVFA
Shopping Mall	-0.56 $\pm$ 0.02	**-0.27 $\pm$ 0.03**	-0.22 $\pm$ 0.03	-0.25 $\pm$ 0.07	-1.04$\pm$ 0.06	-0.96$\pm$ 0.05
Rand. Landm.	-0.18 $\pm$ 0.02	-0.12 $\pm$ 0.02	-0.22 $\pm$ 0.02	**-0.11 $\pm$ 0.02**	-0.89$\pm$ 0.08	-0.71$\pm$ 0.08
Street	-0.16 $\pm$ 0.02	-0.15 $\pm$ 0.02	-0.17 $\pm$ 0.02	**-0.13 $\pm$ 0.02**	-0.64$\pm$ 0.07	-0.60$\pm$ 0.07
Pie-in-the-Face	**-0.21 $\pm$ 0.01**	-0.43 $\pm$ 0.01	-0.24 $\pm$ 0.01	-0.27 $\pm$ 0.01	-0.27$\pm$ 0.007	-0.22$\pm$ 0.009



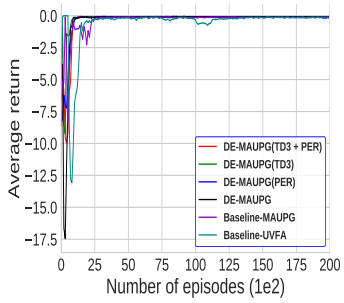
(a) Random Landmarks



(b) Shopping Mall



(c) Street



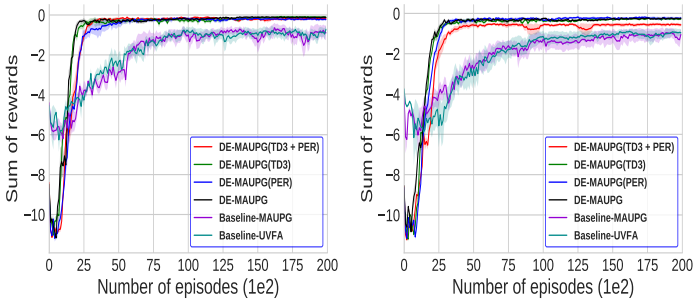
(d) Pie-in-the-Face

Fig. 5: Learning curves representing the average cumulative global reward of multi-scenario learning experiments..

C. Multi-Scenario Experiments

Multi-agent reinforcement learning is sensitive to distortions and does not work well if the trained policies are deployed on scenarios other than the scenario on which the policies are trained on. This problem is generally referred as single-task multi-scenario learning. The goal here is to learn a joint policy $\boldsymbol{\pi}$ that performs equally well as scenario-dependant policy. [15] introduced *Multi-Agent Universal Policy Gradients* (MAUPG) to solve the multi-scenario learning and evaluated it on the VIP protection environments similar to our experiments. The main idea behind MAUPG is to replace the standard centralized Q-functions in MADDPG with Universal Value Function Approximators (UVFA). In an analogous way, we replaced all the critics in DE-MADDPG with UVFAs. For brevity, we will refer our multi-scenario solution as *Decomposed-Multi-Agent Universal Policy Gradients* (DE-MAUPG) For our experiments, we kept our settings identical to DE-MADDPG experiments and trained it for 20,000 episodes. Unlike scenario-dependant training, where every scenario was trained for 20,000 episodes, all scenarios are trained in parallel using one joint policy $\boldsymbol{\pi}$.

Figure 5 shows the learning curves of the DE-MAUPG and its variants. Similar to our results from Section V-B, decomposition based learning solutions outperform the baseline MAUPG. The point to be noted here is that not only our solutions learn to achieve a higher reward but they also learn faster. This can be easily seen in Figure 5a and Figure 5b where DE-MAUPG (TD3 + PER) reaches the maximum performance in less than 2500 episodes while the baseline



(a) Random Landmarks (b) Shopping Mall

(c) Street (d) Pie-in-the-Face

Fig. 6: Learning curves representing the sum of local rewards of multi-scenario learning experiments.

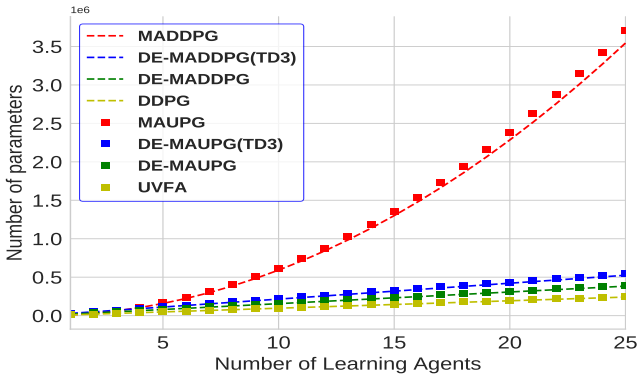


Fig. 7: Number of trainable parameters in the main Q-networks. Notice that 25 agents can be trained by DE-MADDPG approaches using the same number of parameters as 10 MADDPG agents.

TABLE V: Average training time for experiments in minutes over 8 different seeds. Minimum value for each task is bolded. $\pm$ corresponds to standard deviation across seeds

Environment	DE-MADDPG (TD3+PER)	DE-MADDPG (TD3)	DE-MADDPG (PER)	DE-MADDPG	MADDPG
Shopping Mall	168.25 $\pm$ 7.22	152.85 $\pm$ 6.31	167.25 $\pm$ 4.40	**149.5 $\pm$ 1.65**	268.5$\pm$ 21.21
Rand. Landm.	168.87 $\pm$ 0.78	151.25 $\pm$ 1.19	164.14 $\pm$ 1.35	**146.875 $\pm$ 1.16**	262.25$\pm$ 2.86
Street	397.62 $\pm$ 4.71	352.87 $\pm$ 5.63	356.87 $\pm$ 13.97	**340.75 $\pm$ 16.74**	462.75$\pm$ 43.14
Pie-in-the-Face	64.25 $\pm$ 1.23	59.25 $\pm$ 0.59	63.5$\pm$ 1.09	**55.12 $\pm$ 1.05**	72.79$\pm$ 1.16

MAUPG reaches its peak performance at 12,500 episodes in the *Random Landmarks* environment and does not even reach its peak performance before 19,000 episodes for the *Shopping Mall* environment.

Table III quantifies the improvement of DE-MAUPG in maximizing the global reward when compared to MAUPG and UVFA. We find that the decomposition based approaches achieve 81% on the *Shopping Mall* environment and 41% on the *Random Landmarks* environment. Similarly Table IV quantifies the improvement of DE-MAUPG in maximizing the local reward when compared to MAUPG and UVFA.

D. Memory and time complexity

In this section we evaluate the growth of parameter space as the number of agents increases and we empirically evaluate the computation time for DE-MADDPG and compare it with MADDPG and DDPG. The main components of the MADDPG that fuel its learning are the distributed centralized Q-functions. As the number of the agents grow, the input space of those Q-functions increase quadratically. Concretely, assuming all the agents have identical observation and action space, the number of trainable parameters can be represented by $\mathcal{O}(n^2(odim+adim))$, where n is the number of agents, *odim* and *adim* represents the dimensionality of observation and action space respectively. Alternatively, DE-MADDPG solves this scalability problem by having a shared global centralized Q-function whose parameter space increases linearly and can be represented as $\mathcal{O}(n(odim + adim))$. In Figure 7, we show the growth of number of parameters of all the main Q-networks of MADDPG, DE-MADDPG and its TD3 variant. It can be seen that our solution has a significantly smaller number of parameters compared to MADDPG and MAUPG. Note that the figure only represents the number of trainable parameters in the main Q-networks and does not include target or policy networks as the growth of the policy network parameters are identical and no gradient based learning happens in the target networks. All these statements hold true for their UVFA variants.

We empirically verified the benefits of the parameter reduction by measuring the time taken to train the experiments. Table V shows that DE-MADDPG based approaches always train faster than baseline MADDPG. The details of the network architecture are shown in Section VI.

VI. Experimental Details

TABLE VI: The parameters used for various variations of DE-MADDPG and the baseline algorithm MADDPG in the experiments.

Parameter	DE-MADDPG (TD3+PER)	DE-MADDPG (TD3)	DE-MADDPG (PER)	DE-MADDPG	MADDPG	DDPG
Episodes	20k	20k	20k	20k	20k	20k
Replay buffer	10^6	10^6	10^6	10^6	10^6	10^6
Minibatch size	2048	2048	2048	2048	2048	2048
Steps per train Q_g	4	4	4	4	N/A	N/A
Steps per train Q_l	2	2	2	2	2	2
Max env steps	25	25	25	25	25	25
PER α	0.6	N/A	0.6	N/A	N/A	N/A
PER β	0.4	N/A	0.4	N/A	N/A	N/A
PER ϵ	1e-6	N/A	1e-6	N/A	N/A	N/A
PER β decay	10000	N/A	10000	N/A	N/A	N/A

A. Network Architecture

Both actor and critic networks consist of 2 hidden layers containing 64 units in each layer with a ReLU activation

function while the output layers of both networks use linear activation. Both networks are initialized using Xavier normal initializers however, the output layer of the target critics were initialized with uniform random values between (-0.01 and 0.01)). The hyperparameter details are shown in Table VI. Note that same configurations were used for multi-scenario learning experiments.

VII. Conclusions

In this paper, we focused on the dual-reward MARL: a collaborative setting where a group of agents must simultaneously learn to maximize the collective global reward and individual local reward. To solve the problem, we proposed the *Decomposed Multi-Agent Deep Deterministic Policy Gradient (DE-MADDPG)* algorithm and applied it to the problem of defensive escort team: how can agent learn a policy to maintain an optimal formation around the VIP to protect him/her from possible physical attack. We first demonstrated that decomposing a multi-objective reward function leads to higher and more stable performance. We compared our results with the MADDPG algorithm and achieved at least 50% better performance in terms of the collected reward. Additionally, we showed that our solution is computationally efficient and requires a significantly lower number of parameters while achieving better performance than the baseline.

References

[1] J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," in *Proc. of the 16th Int'l Conf. on Autonomous Agents and Multiagent Systems (AAMAS-2017)*, pp. 464–473, 2017.

[2] H. U. Sheikh, M. Razghandi, and L. Bölöni, "Learning distributed cooperative policies for security games via deep reinforcement learning," in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC-2019)*, vol. 1, pp. 489–494, Jul 2019.

[3] OpenAI, *OpenAI Five*, 2018. https://blog.openai.com/openai-five/.

[4] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, N. Sonnerat, T. Green, L. Deason, J. Z. Leibo, D. Silver, D. Hassabis, K. Kavukcuoglu, and T. Graepel, "Human-level performance in first-person multiplayer games with population-based deep reinforcement learning," *arXiv preprint arXiv: 1807.01281*, 2018.

[5] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, 2019.

[6] B. Baker, I. Kanitscheider, T. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," *arXiv preprint arXiv:1909.07528*, 2018.

[7] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-2018)*, 2018.

[8] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "QMIX: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *Proceedings of the 35th International Conference on Machine Learning*, pp. 4295–4304, 2018.

[9] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Advances in Neural Information Processing Systems 30*, pp. 6379–6390, 2017.

[10] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient," in *Proc. of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-2019)*, 2019.

[11] J. Yang, A. Nakhaei, D. Isele, H. Zha, and K. Fujimura, "CM3: cooperative multi-goal multi-stage multi-agent reinforcement learning," *arXiv preprint arXiv:1809.05188*, 2018.

[12] H. U. Sheikh and L. Bölöni, "Designing a multi-objective reward function for creating teams of robotic bodyguards using deep reinforcement learning," in *Proc. of 1st Workshop on Goal Specifications for Reinforcement Learning (GoalsRL-2018) at ICML 2018*, July 2018.

[13] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *International Conference on Learning Representations (ICLR-2016)*, 2016.

[14] S. Fujimoto, H. van Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. of the 35th International Conference on Machine Learning (ICML-2018)*, pp. 1587–1596, 2018.

[15] H. U. Sheikh and L. Bölöni, "Emergence of scenario-appropriate collaborative behaviors for teams of robotic bodyguards," in *Proc. of the 18th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2019)*, pp. 2189–2191, 2019.

[16] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, pp. 330–337, 1993.

[17] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*, pp. 157–163, Elsevier, 1994.

[18] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, Nov 2003.

[19] I. Mordatch and P. Abbeel, "Emergence of grounded compositional language in multi-agent populations," in *Proc. of AAAI International Conference on Artificial Intelligence (AAAI-2017)*, 2017.

[20] L. Panait and S. Luke, "Cooperative multi-agent learning: The state of the art," *Autonomous Agents and Multi-Agent Systems*, vol. 11, no. 3, pp. 387–434, 2005.

[21] J. L. Austerweil, S. Brawner, A. Greenwald, E. Hilliard, M. Ho, M. L. Littman, J. MacGlashan, and C. Trimbach, "How other-regarding preferences can promote cooperation in non-zero-sum grid games," in *Proceedings of the AAAI Symposium on Challenges and Opportunities in Multiagent Learning for the Real World*, 2016.

[22] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proc. of the 35th International Conference on Machine Learning (ICML-2018)*, pp. 5872–5881, 2018.

[23] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian, "Deep decentralized multi-task multi-agent reinforcement learning under partial observability," in *International Conference on Machine Learning*, pp. 2681–2690, 2017.

[24] A. S. Vezhnevets, S. Osindero, T. Schaul, N. Heess, M. Jaderberg, D. Silver, and K. Kavukcuoglu, "Feudal networks for hierarchical reinforcement learning," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3540–3549, JMLR. org, 2017.

[25] C. Wu, A. Rajeswaran, Y. Duan, V. Kumar, A. M. Bayen, S. Kakade, I. Mordatch, and P. Abbeel, "Variance reduction for policy gradient with action-dependent factorized baselines," in *Proc. of the 6th Int'l Conf. on Learning Representations (ICLR)*, 2018.

[26] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[27] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. of the 31st Int'l Conf. on Machine Learning (ICML-2014)*, pp. 387–395, 2014.

[28] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," in *Proc. of the 3rd Int'l Conf. on Learning Representations (ICLR-2015)*, 2015.

[29] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double Q-learning," in *Proc of the Thirtieth AAAI Conference on Artificial Intelligence (AAAI-2016)*, 2016.

[30] J. Ackermann, V. Gabler, T. Osa, Alec, and M. Sugiyama, "Reducing overestimation bias in multi-agent domains using double centralized critics," *arXiv preprint arXiv:1910.01465*, 2019.