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

Hassan 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 either maximize 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 challenging defensive escort team problem and show that our solution achieves a significantly 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], coordination between autonomous vehicles [2] and collaboration of agents in defensive escort teams [3]. 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 [4], Quake III Capture-the-Flag [5] and Starcraft II [6] or learning to use tools [7], 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.

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

Current state-of-the-art MARL algorithms can be categorized in two types. For algorithms such as COMA [8] and QMIX [9], the goal is to maximize the global reward for the success of the group while algorithms such MADDPG [10] and M3DDPG [11] focus on optimizing local rewards without any explicit notion of coordination. As shown in [12] 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 [13]. 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 either of the rewards is sparse, thus, leading to a bias towards 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 reward. The local critic receives as input only the observation and action of the particular agent and estimates the sum of local expected reward. The advantage of training two critics is that the step of designing an entangled multi-objective reward function can be skipped altogether.

To summarize, our contributions in this paper are the 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) \[14\] and Twin Delayed Deep Deterministic Policy Gradients (TD3) \[15\] 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 escort team problem \[3, 16\] (see Figure \[1\]) 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 \[17, 18, 19\]. Recent work have adopted techniques from single-agent deep RL to develop general algorithms for high-dimensional continuous space environments requiring complex agent interactions \[11, 20, 10\].

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 \[21\]. Similar to our work, \[22\] have shown that agents whose rewards depend on all agents’ success perform better than agents who optimize for their own success. In the special case when all agents have a single goal and share a global reward, COMA \[8\] 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 \[9\], but do not address credit assignment. While MADDPG \[10\] and M3DDPG \[11\] apply to agents with different reward functions, they do not specifically address the need for cooperation; in fact, they do not distinguish the problems of cooperation and competition, despite the fundamental difference.

To the best of our knowledge, dual-reward MARL was not explicitly addressed in the existing literature. Among the related problems, \[23\] 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 \[24\], and in contrast to hierarchical methods with top-level managers that sequentially select subtasks \[25\], our decentralized agents must cooperate in parallel to successfully achieve the global and their respective 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 that instead of parameterizing the Q-function to extract the policy, we parameterize the policy using the parameters θ to maximize the objective represented as $J(θ) = E[|R|^2]$ by taking a step in the direction

$$\nabla J(θ) = E[∇_θ \log \pi_θ(a|s) Q^π(s, a)]$$

Policy gradient methods are prone to the high variance problem. Several methods such as \[26, 27\] 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. \[28\] has shown that it is possible to extend the policy gradient framework to deterministic policies i.e. $π_θ : S → A$. In particular we can write $∇J(θ)$ as

$$\nabla J(θ) = E[∇_θ π(a|s) ∇_a Q^π(s, a)|a_π(s)]$$

A variation of this model, Deep Deterministic Policy Gradients (DDPG) \[29\] is an off-policy algorithm that approximates the policy π and the critic Qπ 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) \[15\] improves on DDPG by addressing the overestimation bias of the Q-function, similarly to Double Q-learning \[30\]. 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 $Q_φ_1, Q_φ_2$, along with two target networks. The Q-functions are updated with the target $y = r^t + γ \min_{i=2} Q_φ_1(s', a')$, while updating the policy with $Q_φ_1$. Additionally, they introduce target policy smoothing by adding noise in the determination of the next action for the critic target $a' = ρ_{θ} (s') + ϵ$, with $ϵ$ being clipped Gaussian noise $ϵ = c_{lip}(N(0, σ), -c, c)$, 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) \[10\] extends DDPG for the multi-agent setting where each agent has its own policy. The gradient of each policy is written as

$$\nabla J(θ_i) = E[∇_θ_i π_i(a_i|o_i) ∇_a_i Q^π_i(s, a_1, \ldots, a_N)|a_i = π_i(o_i)]$$

where $s = (o_1, \ldots, o_N)$ and $Q^π_i(s, a_1, \ldots, a_N)$ 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 primary insight behind MADDPG is that knowing all the actions of other agents makes the environment stationary, even though their policy changes.

Very recently \[31\] proposed Multi-Agent TD3 (MATD3) extending MADDPG by replacing the deterministic policy gradients with twin delayed deterministic policy gradient to tackle the overestimation bias problem.

IV. DECOMPOSED MULTI-AGENT DEEP DETERMINISTIC POLICY GRADIENT

We propose Decomposed Multi-Agent Deep Deterministic Policy Gradient: a multi-agent deep reinforcement learning
The idea of MADDPG by introducing a local critic. Now the modified policy gradient for each agent $i$ can be written as

$$
\nabla J(\theta_i) = \mathbb{E}_{s,a \sim D} \left[ \nabla_{\theta_i} \pi_i(a_i | o_i) \nabla_{a_i} Q_{\phi_i}^g(s, a_1, \ldots, a_N) \right]
$$

$$
+ \mathbb{E}_{o_i,a_i \sim D} \left[ \nabla_{\theta_i} \pi_i(a_i | o_i) \nabla_{a_i} Q_{\psi_i}^l(o_i, a_i) \right]
$$

$D$ is the experience replay buffer. The global critic is $Q_{\psi}^g$ and $Q_{\psi}^l$.

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

Overestimation bias in Q-functions have been thoroughly studied in [15], [31]. This overestimation bias can be problematic 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 turns. To solve this problem [15] have proposed a double critic approach to minimize the overestimation bias. Motivated from the results in [15], 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
The twin global critics are updated as
\[
\nabla J(\theta_j) = \mathbb{E}_{s,a \sim D} \left[ \nabla \theta_j \pi_j(a_i|o_i) \nabla a_i Q^\psi_j(s,o_1,\ldots,o_N) \right] + \mathbb{E}_{o_i,a_i \sim D} \left[ \nabla \theta_j \pi_j(a_i|o_i) \nabla a_i Q^\pi_j(a_i) \right]
\]

(2)

The twin global critics are updated as
\[
\mathcal{L}(\psi_i) = \mathbb{E}_{s,a,r,s'} \left[ (Q^\psi_i(s,o_1,\ldots,o_N) - y_g)^2 \right]
\]

where \( y_g \) is defined as:
\[
y_g = r_g + \gamma \min_{i=1,2} Q^\psi_i(s',o_1',\ldots,o_N') |_{a_i' = \pi_i(o_i')}
\]

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 2.

V. EXPERIMENTS

A. Environments

We perform our experiments using the defensive escort problem on four VIP protection environments [16], [13]. 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 [20]. An illustration of the environments can be seen in Figure 1.

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

**Algorithm 1 Decomposed Multi-agent Deep Policy Gradient**

1. Initialize main global critic networks \( Q^\psi_{o_1}, Q^\psi_{o_2}, \ldots, Q^\psi_{o_N} \)
2. Initialize target global critic networks \( Q^\psi'_{o_1}, Q^\psi'_{o_2}, \ldots, Q^\psi'_{o_N} \)
3. Initialize each agents policy and critic networks.
4. for episode = 1 to \( T \) do
5. for \( t = 1 \) to episode-length do
6. Get environment state \( s_t^i \).
7. For each agent \( i \), select action \( a_t^i = \pi_{\theta_i}(o_t^i) \)
8. Execute actions \( a^i = [a_1^i, \ldots, a_N^i] \)
9. Receive global \( r_t^i \) and local rewards \( r_t^j \).
10. Store \( (s_t^i, a^i, r_t^i, r_t^j, s_{t+1}) \) in replay buffer.
11. end for
12. // Train global critic
13. Sample minibatch of size \( S \) \( (s_t^i, a^i, r_t^i, s_{t+1}) \) from buffer.
14. \( (a_t^1, \ldots, a_t^N) := (\pi_{\theta_1}(o_t^1), \ldots, \pi_{\theta_N}(o_t^N)) \)
15. Set \( y_t^j = r_t^j + \gamma \min_{i=1,2} Q^\psi_j(s_{t+1}, a_{t+1}^1, \ldots, a_{t+1}^N) \)
16. Update global critics by minimizing
\[
\frac{1}{S} \sum_j (y_t^j - Q^\psi_j(s_t^i, a_t^i, \ldots, a_t^i))^2
\]
17. Update target network parameters
\[ \psi_i' \leftarrow \tau \psi_i + (1 - \tau) \psi_i' \]
18. if episode mod \( d \) then
19. // Train local critics and update agent policies
20. for agent \( i = 1 \) to \( N \) do
21. Sample minibatch of size \( S \) \( (s_t^i, a_t^i, r_t^j, s_{t+1}^i) \)
22. Set \( y_t^j = r_t^j + \gamma Q^\pi_{\phi_j}(s_{t+1}^i, a_{t+1}^j, \ldots, a_{t+1}^N) \)
23. Update local critic by minimizing
\[
\frac{1}{S} \sum_j (y_t^j - Q^\pi_{\phi_j}(a_j, a_{t+1}^j))^2
\]
24. \( \theta_i = \theta_i + \frac{1}{S} \sum_j \nabla \theta_i \pi_i(a_i|o_i) \nabla a_i Q^\psi_j(s_t^i, a_t^1, \ldots, a_t^N) \)
25. end for
26. Update target network parameters for each agent \( i \)
\[ \theta_i' \leftarrow \tau \theta_i + (1 - \tau) \theta_i' \] \[ \phi_i' \leftarrow \tau \phi_i + (1 - \tau) \phi_i' \]
27. end if
28. end for
The environment consists of the VIP (payload), defensive escort team and one or more classes of bystanders. The VIP (brown disk) starts from the starting point and moves towards the destination landmark (green disk). The goal of the defensive escort team is learn an optimal formation around the VIP to protect it from potential physical attacks. In order to maintain social norms, the defensive agents need to maintain a certain distance from the VIP. To closely simulate 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 $o_t = [x_{j,...,N+M}] \in O_t$ where $x_j$ is the observation of the entity $j$ from the perspective of agent $i$. In our experiments, we used $M = 5$.

We chose the defensive escort team problem with these environments, because it is a representative case of a well structured dual-reward collaborative MARL problem, the environments are continuous with a relatively high dimensional state space. Furthermore, previous work using these environments provide strong baselines against which the proposed algorithms can be compared. Aligning our experimental setup with [3], [16], we chose 4 bodyguards and 10 bystanders.

**Decomposing the Reward Function**

In this section, we review the entangled multi-objective reward function defined in [13], [4]. explain the problems with it and decompose it to be used by DE-MADDPG.

As mentioned in 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:

$$ r_{total} = \alpha \left( -1 + \prod_{k=1}^{M} (1 - RT(VIP, b_k, R)) \right) + (1 - \alpha) \left( D(VIP, x_i) \right) $$

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

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

where $m$ is the minimum distance the agent has to maintain from VIP and $d$ is the safe distance. As per our understanding, $r_{total}$ has 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. This phenomenon is also explained in Section VC.

The second problem is the $\alpha$ hyper-parameter. The $\alpha$ 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 reinforcement learning experiments take substantial amount of time, finding an optimal $\alpha$ 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, $Q^\pi_{\psi}$ learns to approximate $r_{\text{global}}$ while $Q^\pi_{\Theta}$ approximates $r_{\text{local}}$. The benefit of this decomposition is more stable learning and exclusion of the $\alpha$ hyper-parameter.

B. Evaluations

To evaluate the efficacy of DE-MADDPG and its variants we compare our results with baseline MADDPG and DDPPG on 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 three layered neural networks 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 DE-MADDPG based approaches outperforms the MADDPG and DDPPG by a significant margin. Similar observation can be seen for the other two environments i.e., Shopping Mall, and Street. Finally, for the last environment Pie-in-the-Face, there is little difference between the performance of the different approaches. A possible reason for this is that this environment, focusing on a single attacking bystander, makes the positioning choice less complex.

Though, fig. 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 1 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 augmented with TD3 and PER were 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.

| Environment     | DE-MADDPG (TD3 + PER) | DE-MADDPG (TD3) | DE-MADDPG (PER) | MADDPG | DDPPG |
|-----------------|-----------------------|----------------|----------------|--------|-------|
| Shopping Mall   | -4.07 ± 0.06          | -5.61 ± 0.06   | -6.17 ± 0.04   | -8.22 ± 0.07 | -7.99 ± 0.13 | -9.91 ± 0.12 |
| Rand. Landmarks | -4.01 ± 0.03          | -4.01 ± 0.06   | -5.75 ± 0.04   | -6.34 ± 0.08 | -7.67 ± 0.13 | -10.20 ± 0.16 |
| Street          | -2.13 ± 0.06          | -2.38 ± 0.07   | -2.66 ± 0.07   | -2.66 ± 0.08 | -3.43 ± 0.13 | -3.81 ± 0.11 |

A common pattern can be seen in Figure 4 and Figure 1 that DE-MADDPG based approaches outperforms MADDPG and DDPPG results. One point to be noted here is that maintaining a certain distance from a moving payload is fairly an easy problem for standard reinforcement learning algorithms as the reward is dense and the state space is relatively easy but 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.

| Environment     | DE-MADDPG (TD3 + PER) | DE-MADDPG (TD3) | DE-MADDPG (PER) | MADDPG | DDPPG |
|-----------------|-----------------------|----------------|----------------|--------|-------|
| Shopping Mall   | -4.01 ± 0.02          | -4.23 ± 0.03   | -4.26 ± 0.03   | -5.90 ± 0.07 | -4.61 ± 0.07 | -4.92 ± 0.08 |
| Rand. Landmarks | -0.02 ± 0.01          | -0.10 ± 0.01   | -0.25 ± 0.02   | -0.22 ± 0.03 | -1.42 ± 0.06 | -0.94 ± 0.08 |
| Street          | -0.12 ± 0.01          | -0.15 ± 0.02   | -0.16 ± 0.02   | -0.18 ± 0.08 | -1.72 ± 0.08 | -0.47 ± 0.04 |

C. Stability

One of the focal point of this study was to find a stable solution that solves the dual-reward MARL problem. In this section, we analyse the stability of DE-MADDPG based solutions. Though, Figure 3 shows that DE-MADDPG when augmented with TD3 outperforms MADDPG in maximizing global reward, the average of results across all the seeds make it difficult to analyze the stability. For that purpose, we chose fixed single seed runs of all the solutions on the Shopping Mall environment and plotted their learning curves. It can be seen in Figure 5 that all DE-MADDPG based solutions are stable in learning to maximize global reward as compared to baseline MADDPG. This behavior is common across all seeds and environments except Pie-in-the-Face as it is does not provide any complexity.
D. 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 \( \pi \) that performs equally well as scenario-dependant policy. In [16] have 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). Given the similarity between MADDPG and MAUPG, 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. The point to note here is that unlike scenario-dependant training, where every scenario was trained for 20,000 episodes, all scenarios are trained in parallel using one joint policy \( \pi \).

Figure 6 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 our solutions learn to achieve higher reward but they also learn faster. This can be easily seen in Figure 6a and Figure 6b where DE-MAUPG (TD3 + PER) reaches the maximum performance in less than 2500 episodes where baselines MAUPG reaches its peak performance at 12,500 episodes in Random Landmarks environment and does not even reach its peak performance before 19,000 episodes for the Shopping Mall environment.

Table IV quantifies the improvement of DE-MAUPG in maximizing 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 local reward when compared to MAUPG and UVFA.

E. Computational Evaluations

In this section we evaluate the growth of parametric space as the number of agents increases. Moreover, we empirically evaluate the computational time for DE-MADDPG and compare it with MADDPG and DDPG. The main component of the MADDPG that fuels 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 \( O(n^2(\text{dim} + \text{adim})) \). 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 parametric space increases linearly and can be represented as \(O(n(odim + adim))\). In Figure 8, 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 takes significantly small number of parameters as 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 parametric reduction by measuring the time taken to train the experiments. It can be seen in Table VI that DE-MADDPG based approaches always train faster than baseline MADDPG. The complete details about the network architecture can be seen in Section VI. We do not provide time comparisons for DDPG and UVFA variants as the experiments were run on multiple machines with a variety of hardware thus making it difficult for a fair comparison.

VI. EXPERIMENTAL DETAILS

A. Network Architecture

Both actor and critic networks for all agents consists of 2 hidden layers containing 64 units in each layer. The hidden layers use ReLU activation function while the output layers of both networks use linear activation function. 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) to enable one-step look ahead learning of the critics after each training cycle. To keep experiments as fair as possible, we initialized the network parameters with same seed across all the experiments. Complete hyperparameter details can be seen in Table VI. Note that same configurations were used for multi-scenario learning experiments.

VII. CONCLUSION

In this paper, we focused on the dual-reward MARL: a collaborative setting where a group of learning agents have to 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. Finally, we showed that by replacing the standard critics with UVFAs, our solution also outperforms MAUPG which is a baseline algorithm for single-task multi-scenario learning in multi-agent reinforcement learning.

REFERENCES

[1] J. Z. Leibo, V. Zambaldi, M. Lancot, 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] Y. Cao, W. Yu, W. Ren, and G. Chen, “An overview of recent progress in the study of distributed multi-agent coordination,” IEEE Transactions on Industrial informatics, vol. 9, no. 1, pp. 427–438, 2012.

[3] 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.

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

[5] M. Jaderberg, W. M. Czarnecki, T. Dunning, L. Marsi, 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.

[6] 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. Danilhelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Veznheevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsly, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schauff, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver, “Grandmaster level in starcraft ii using multi-agent reinforcement learning,” Nature, 2019.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] 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.

[13] 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.

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

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

[16] 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.

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

[18] M. L. Littman, “Markov games as a framework for multi-agent reinforcement learning,” in Machine Learning Proceedings, vol. 4, pp. 1039–1069, Nov 2003.

[19] 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.

[20] I. Mordatch and P. Abbeel, “Emergence of grounded compositional language in multi-agent populations,” in Proc. of AAAI Conference on Articial Intelligence (AAAI-2017), 2017.

[21] 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. 378–434, 2005.

[22] 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.

[23] 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.

[24] S. Omidsfahani, 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. 2601–2609, 2017.

[25] A. S. Veznheveets, 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.

[26] 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.

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

[28] 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.

[29] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, M. Jaderberg, A. G. Zanfir, 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.

[30] H. Van Hasselt, A. Guez, and D. Silver, “Deep reinforcement learning with double Q-learning,” in Proc of the Thirty-seventh AAAI Conference on Artificial Intelligence (AAAI-2016), 2016.

[31] 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.

[32] T. Viscic, A. Czirók, E. Ben-Jacob, I. Cohen, and O. Shochet, “Novel type of phase transition in a system of self-driven particles,” Physical Review Letters, vol. 75, no. 6, p. 1226, 1995.