Multi-Agent Communication using Difference Rewards Policy Gradients

Simon Vanneste[†][0000−0002−9664−9925], Astrid Vanneste[†][0000−0002−6742−6722], Tom De Schepper[††][0000−0002−2969−3133], Siegfried Mercelis[†][0000−0001−9355−6566], Peter Hellinckx[*][0000−0001−8029−4720], and Kevin Mets[†][0000−0002−4812−4841]

University of Antwerp - imec
IDLab - [†]Faculty of Applied Engineering, [††]Department of Computer Science
Sint-Pietersvliet 7, 2000 Antwerp, Belgium
[*]University of Antwerp, Faculty of Applied Engineering
{simon.vanneste, astrid.vanneste, tom.deschepper, siegfried.mercelis, peter.hellinckx, kevin.mets}@uantwerpen.be

Abstract. Communication learning while learning a behaviour policy is a challenging problem within the multi-agent reinforcement learning domain. In this work, we combine the Multi-Agent Counterfactual Communication (MACC) method with the Difference Reward Policy Gradient (DR.PG) method and propose the novel Difference Reward Multi-Agent Counterfactual Communication (DR.MACC) method. The DR.MACC method enables us to create an agent-specific difference return for the action and communication policy of the agents. This policy-specific difference return minimizes the credit-assignment problem compared to using the team reward directly. The DR.MACC method does not require us to learn a joint Q-function, like the MACC method, but instead operates using the environment's reward function. Alternatively, when the reward function is unavailable, we can learn an approximation of the reward function in the DRR.MACC method. Here, the agent's environment interactions are used to train the approximation of the reward function using supervised learning. In the experiments, we compare the novel DR.MACC method against the MACC method with an individual Q-function and a joint Q-function. The results show that the DR.MACC method can outperform both MACC variants in the different environment configurations.

Keywords: Multi-Agent · Reinforcement Learning · Communication Learning · Difference Return.

1 Introduction

Single-agent Reinforcement Learning (RL) [9–11, 17, 18] has received a lot of attention from the machine-learning community. However, many real-world systems can naturally be described as cooperative multi-agent systems (e.g. industrial robotic or traffic light control [14]). When combining multi-agent systems

with RL, we enter the domain of Multi-Agent Reinforcement Learning (MARL). In cooperative MARL systems, the agents need to learn to work together to achieve a common goal. To achieve this, the agents are trained using a shared team reward which encourages cooperative behaviour and prevents the agents from learning any competitive behaviour. In certain environments, the agents need to share information with the other agents to successfully reach their common objective. This inter-agent communication [4, 20] can simultaneously be learned while training the action policy (e.g. communication for MARL traffic light control [19]). This allows the agents to learn a custom communication protocol which is specifically created for a certain environment.

A general problem within MARL with or without communication is the credit-assignment problem [1, 2, 5, 20]. This problem is caused by only using a single reward to train a set of cooperative agents which makes it difficult for the agents to extract their impact on the team reward. A common method to handle the credit-assignment problem within cooperative MARL is by using the Centralized Training and Decentralized Execution (CTDE) paradigm [1, 4, 5, 13, 20]. Within this paradigm, the agents are trained centralized which allows us to use centralized training structures (like parameter sharing, free communication between the agents, and a centralized critic) while the policies can still be deployed decentralized. This paper tackles the credit-assignment problem in cooperative MARL with inter-agent communication, using the CTDE paradigm, by presenting the novel Difference Reward Multi-Agent Counterfactual Communication (DR.MACC) and Difference Reward with a learned Reward model Multi-Agent Counterfactual Communication (DRR.MACC) methods. These methods combine the DR.PG method [1] (see Section 3.2) and the MACC method [20] (see Section 3.3). These methods allow us to learn a discrete action and discrete communication policy, without the need to learn a joint Q-function which has scalability problems, by using or learning a joint reward function.

2 Related Work

An important problem in cooperative MARL is the credit-assignment problem [1, 5, 20]. This is caused by training a set of cooperative agents using a team reward which makes it difficult for an agent to observe its impact on this reward. A popular method to tackle the credit-assignment problem is by using a centralized critic under the CTDE paradigm. MADDPG [7] is a multi-agent variant of the Deep Deterministic Policy Gradient (DDPG) method [15] where we use a centralized critic allowing it to reduce the credit-assignment problem. However, using the centralized critic requires us to learn a joint Q-function which poses scaling and stability problems for any method that uses a centralized critic. Next, Foerster *et al.* [5] presented the Counterfactual Multi-Agent (COMA) policy gradients method which also uses a centralized critic. Here the critic creates an agent-specific advantage by subtracting a counterfactual baseline from the joint Q-value which is then used to train the individual agents. The COMA method shares a similar disadvantage as MADDPG by requiring a joint Q-function. The

Difference Reward Policy Gradient (DR.PG) [1] method is an alternative to the COMA method where there is no need to train a joint Q-function. The method uses the environment reward function or a learned approximation to create an agent-specific difference return. This difference return is then used to train the individual agents. Section 3.2 discusses this method in more detail.

Next, we describe the related work of communication learning in MARL. Foerster *et al.* [4] presented the Differentiable Inter-Agent Learning (DIAL) which uses a differentiable communication channel. This communication channel allows the gradients to flow back from the receiving agents back towards the sending agent (inter-agent backpropagation) and learn a communication protocol between the agents. The action policies of the agents are trained by using independent Q-learning. Sukhbaatar *et al.* [16] propose a similar method called CommNet which learns a communication protocol by sharing the hidden state of an agent with the other agents. This hidden state is then also trained using inter-agent backpropagation. Targeted Multi-Agent Communication (TarMAC) [3] is a communication learning method where the agents add a signature to each send message which allows the receiving agents to focus on the relevant messages by using an attention mechanism. Similar to the DIAL and CommNet methods, TarMAC also uses a differentiable communication channel to train the communication policy by using inter-agent backpropagation. Additionally, this method uses a centralized critic (joint Q-function) in the policy gradient update for the different agents. Jaques *et al.* [6] presented a method to learn the social influence between agents in a MARL system. Here they investigate social influence through the actions of the other agents and through learning a communication protocol. The agents are trained to learn their social influence by adding a causal influence reward which models the change in distribution when the agents perform an action or send a message. This distribution is obtained by reasoning about counterfactual actions. Using this method it is not necessary to use a differentiable communication channel. Finally, the MACC [20] method is a communication variant of the COMA method. This method learns the communication policy by using counterfactual reasoning to obtain a policy-specific advantage. In Section 3.3, the MACC method is discussed in more detail. However, similar to the COMA method, the method requires us to learn a joint Q-function which again poses scaling and stability problems.

So in this work, we present the DR.MACC method which combines the DR.PG method with the MACC method. This allows us to learn a communication protocol using a counterfactual baseline without the need to learn a joint Q-function.

3 Background

In this section, we describe the background information on the Multi-Agent Markov Decision Process, Difference Reward Policy Gradient (DR.PG) [1], and Multi-Agent Counterfactual Communication (MACC) [20]. This section and the following sections use a shared notation where the superscript is used as an

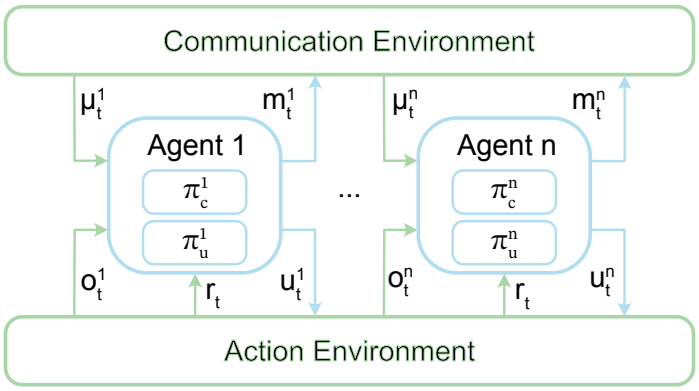


Fig. 1: The Dec-MDP with communication between the agents.

agent identifier. Here, a identifies the current agents and $-a$ identifies the group of all the agents without the current agent. The joint variant is indicated by the omission of the superscript. Next, the subscript indicates if the symbol is used for actions (u) or communication (c). Additionally, the subscript is also used to define the current time step t where $t : T$ is used to show the group of times steps from t until the terminal time step T ($s_{t:T} = \cup_{k=0}^{T} s_{t+k}$). Finally, the notation $\mathbb{E}[X; P]$ represents the expected value of X under the distribution P.

3.1 Markov Decision Process

In this work, we use the Decentralized Markov Decision Process (Dec-MDP) [12] framework. Here, n agents are trained using a shared team reward r_t to achieve cooperative behaviour between the agents. At time step t, every agent a selects an action u_t^a based on the action policy of the agent π_u^a using the observation o_t^a and the received messages μ_t^a. Additionally, every agent also includes a communication policy π_c^a which generates an output message m_t^a by using the observation o_t^a and input messages μ_t^a The input messages are created by the communication environment (defined by the communication function M) based on the output messages of the communication policies $\mu_{t+1} = M(m_t)$ (see Figure 1). A Dec-MDP is a jointly observable environment which means that the global environment state s_t can be uniquely identified by the joint observation o_t. When this property does not hold, we end up in the Decentralized Partially Observable Markov Decision Process [12] framework (see Section 4.5 where we describe the DR.MACC equations to use in a Dec-MDP).

3.2 Difference Reward

Castellini *et al.* [1] described the Difference Reward Policy Gradient (DR.PG) method which is a method that is closely related to the COMA method [5]. Both these methods use the aristocrat utility [21] to create a baseline which enables

them to create an agent-specific advantage or difference reward. However, the COMA method requires us to learn a joint Q-function while the DR.PG can use the environment reward function directly when possible or learn a model of the reward function in the Difference Reward Policy with a learned Reward model Gradient (DRR.PG) method. The agent-specific action difference reward ΔR_u^a is the difference between the reward and the expected reward $\bar{R}_u^a$ under policy π_u^a. The expected reward can be calculated by using the reward function R and the policy of the agents (see Equation 1). Here, the joint action is defined by the concatenation of the actions of the other agents u_t^{-a} and the alternative actions of the current agent $u_t'^a$.

$$\begin{aligned}\Delta R_u^a(s_t, u_t) &= r_t - \bar{R}_u^a(s_t) \\ &= r_t - \sum_{u_t'^a} \pi_u^a(u_t'^a | o_t^a) R(s_t, \langle u_t'^a, u_t^{-a} \rangle)\end{aligned} \tag{1}$$

The difference return ΔG_u^a is obtained by calculating the discounted difference reward as shown in Equation 2.

$$\Delta G_u^a(s_{t:T}, u_{t:T}) \triangleq \sum_{k=t}^{T} \gamma^{k-t} \Delta R^a(s_k, u_k) \tag{2}$$

Next, the gradient g_u^a for the action policy is obtained by using the difference return (see Equation 3). The gradient can then be used to update the parameters of the action policy for agent a.

$$g_u^a = \mathbb{E}\Big[\sum_{t=0}^{T-1} \nabla_{\theta_u^a} \ln \pi_{u,\theta_u^a}^a(u_t^a | o_t^a, \mu_t^a) \Delta G_u^a(s_{t+1:T}, u_{t:T-1}) \Big] \tag{3}$$

Finally, the reward function is not available in every situation, so Castellini *et al.* [1] also presented the DRR.PG method which learns the reward function after which the previously described DR.PG method is used to train the action policy.

3.3 Multi-Agent Counterfactual Communication

Multi-Agent Counterfactual Communication (MACC), as described by Vanneste *et al.* [20], learns a centralized action Q-function Q_u that is used to learn both the action policy π_u (similar to COMA) and communication policy π_c by calculating a policy-specific advantage. The communication policy gradient is shown in Equation 4 which is used to update the parameters of the communication policy.

$$g_c^a = \mathbb{E}\Big[\sum_{t=0}^{T-1} \nabla_{\theta_c^a} \ln \pi_{c,\theta_c^a}^a(m_t^a | o_t^a, \mu_t^a) A_c^a(s_{t+1:T}, m_{t:T-1}) \Big] \tag{4}$$

The communication policy advantage A_c^a is used to learn the communication policy and can be obtained by using counterfactual reasoning. These calculations are shown in Equation 5 and 6.

$$A_c^a(s_{t,t+1}, \boldsymbol{\mu}_t^a, m_t) = Q_c(s_{t,t+1}, m_t) - V_c^a(s_{t,t+1}, \boldsymbol{\mu}_t^a, m_t^{-a}) \tag{5}$$

$$V_c^a(s_{t,t+1}, \boldsymbol{\mu}_t^a, m_t^{-a}) = \sum_{m_t'^a} \Big(Q_c(s_{t,t+1}, \langle m_t'^a, m_t^{-a} \rangle) \pi_c^a(m_t'^a | o_t^a, \boldsymbol{\mu}_t^a) \Big) \tag{6}$$

The communication Q-function Q_c cannot be learned directly because of the non-stationarity of the communication utility due to the changing policy of the other agents. While the communication Q-function Q_c cannot be learned, it can be calculated using the policies of the other agents and the action Q-function Q_u. To do so, the communication Q-function Q_c is split into the Communication Policy to Action Policy (CU) Q-function Q_{cu} and the discounted Communication Policy to Communication Policy (CC) Q-function Q_{cc} as shown in Equation 7.

$$Q_c(s_{t,t+1}, m_t) = Q_{cu}(s_{t,t+1}, m_t) + \gamma_c Q_{cc}(s_{t,t+1}, m_t) \tag{7}$$

The CU Q-function Q_{cu} is defined as the expected action Q-value under the action policy of the agents given the output messages as shown in Equation 8. The CC Q-function Q_{cc} can be calculated by using a specialized algorithm for which we refer to the work of Vanneste *et al.* [20].

$$Q_{cu}(s_{t,t+1}, m_t) = \mathbb{E}\Big[Q_u(s_{t,t+1}, u'_{t+1}); \pi_u(u'_{t+1} | o_{t+1}, M(m_t)) \Big] \tag{8}$$

Vanneste *et al.* [20] investigate several methods to approximate the CU Q-function Q_{cu} to reduce the computational cost. Based on these results, we use Agent-Based Sampling (ABS) as shown in Equation 9.

$$\begin{aligned} &\hat{Q}_{cu}(s_{t,t+1}, m_t) = \frac{1}{n} \sum_{a'=0}^{n} \mathbb{E}\Big[Q_u(s_{t+1}, (u_{t+1}'^{a'}, \widetilde{u}_{t+1}^{-a'})); \pi_u^{a'}(u_{t+1}'^{a'} \mid o_{t+1}^{a'}, Ma'(m_t)) \Big] \\ &\text{with: } \widetilde{u}_{t+1}^{-a'} \sim \pi_u^{-a'}(o_{t+1}^{-a'}, M{-}a'(m_t)) \end{aligned} \tag{9}$$

4 Method

In this section, we describe our novel method which takes the advantages of the DR.PG method [1] and applies it to the MACC method [20]. This combination allows us to learn a communication policy using difference rewards while using the DR.PG method to learn the action policy.

4.1 Difference Reward Multi-Agent Counterfactual Communication

In our novel Difference Reward Multi-Agent Counterfactual Communication (DR.MACC) method, the agent-specific communication policy is trained using the agent-specific communication difference reward. The global architecture of the DR.MACC method is shown in Figure 2. In this architecture, the reward critic creates an action and communication difference return based on the reward function. Next, the communication difference return is used to calculate the gradient g_c^a (see Equation 10) of the communication policy for agent a which is used to update the neural network parameters θ_c^a.

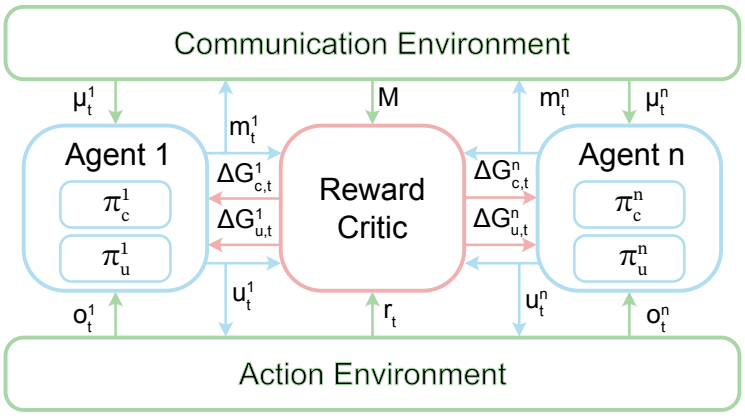


Fig. 2: The DR.MACC architecture where a centralized reward critic calculates a difference reward for the action and communication policies.

$$g_c^a = \mathbb{E}\Big[\sum_{t=0}^{T-1} \nabla_{\theta_c^a} \ln \pi_{c,\theta_c^a}^a(m_t^a|o_t^a,\mu_t^a) \Delta G_c^a(s_{t+1:T}, m_{t:T-1})\Big] \tag{10}$$

The agent-specific communication difference return ΔG_c^a is composed of the Communication Policy to Action Policy (CU) difference return ΔG_{cu}^a and the Communication Policy to Communication Policy (CC) difference return ΔG_{cc}^a (see Equation 11).

$$\Delta G_c^a(s_{t+1:T}, m_{t:T-1}) = \Delta G_{cu}^a(s_{t+1:T}, m_{t:T-1}) + \gamma_c \Delta G_{cc}^a(s_{t+1:T}, m_{t:T-1}) \tag{11}$$

First, the CU return G_{cu}^a models the expected difference return of a certain message under the joint action policy of the different agents (see Equation 12). This CU return is based on the CU Q-function from the MACC method as shown in Equation 8. Intuitively, this CU return represents the impact of a message on the selection of the joint action and how it impacts the expected return. It is important to note that this equation uses the communication function M to determine which messages are received by which agents.

$$G_{cu}(s_{t+1:T}, m_{t:T-1}) = \mathbb{E}\big[G_u(s_{t+1:T}, u'_{t+1:T}); \pi_u(u'_{t+1:T}|o_{t+1:T}, M(m_{t:T-1}))\big] \tag{12}$$

Equation 13 shows how we can obtain the CU difference return based on the CU return.

$$\Delta G_{cu}^a(s_{t+1:T}, m_{t:T-1}) = G_{cu}(s_{t+1:T}, m_{t:T-1}) - \mathbb{E}\big[G_{cu}(s_{t+1:T}, \langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle); \pi_c^a(m'^a_{t:T-1}|o_{t+1:T}, \mu^a_{t+1:T})\big] \tag{13}$$

Next, the CC return G_{cc} is the expected CU return under the joint communication policy which is also similar to the CC Q-function from the MACC method.

$$G_{cc}(s_{t+1:T}, m_{t:T-1}) = \mathbb{E}\big[G_{cu}(s_{t+2:T}, m'_{t+1:T-1}); \pi_c(m'_{t+1:T-1}|o_{t+1:T-1}, \mu_{t+1:T-1})\big] \tag{14}$$

The CC difference return is the difference between the CC return and the expected CC return over the communication policy (see Equation 15).

$$\Delta G^a_{cc}(s_{t+1:T}, m_{t:T-1}) = G_{cc}(s_{t+1:T}, m_{t:T-1}) - \mathbb{E}[G_{cc}(s_{t+1:T}, \langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle); \pi^a_c(m'^a_{t:T-1}|o_{t+1:T}, \mu^a_{t+1:T})] \tag{15}$$

Finally, the DR.MACC method uses the same loss function as described in the MACC method. The social loss function (see Equation 16) is an additional loss function for the action and communication policy to promote social behaviour. In our context more social behaviour results in adapting the action and communication distribution depending on which message is presented. This is achieved by increasing the loss when the distribution does not change when different input messages are presented. This change in distribution is important when learning a communication protocol because when the other agents do not have a change in distribution, the resulting communication difference return will be zero. We refer to the work of Vanneste *et al.* [20] for a more detailed description of the social loss.

$$\mathcal{L}^s(\theta^i_{\pi^a_u}) = -\frac{\lambda}{k}\sum_{x=0}^{k}\left|\pi^a_u(o^a, \mu, \theta^i_{\pi^a_u}) - \pi^a_u(o^a, (\neg\mu_x, \mu_{-x}), \theta^i_{\pi^a_u})\right| \tag{16}$$

4.2 Memory-Improved Equations for the Communication Returns

Calculating the different returns to obtain the CU difference return can be computationally expensive and memory-intensive. So, we investigated memory-improved equations to acquire the different communication returns. These equations allow us to reuse the action difference returns which are used to train the action policy and discard the action returns after the difference return is obtained. The memory-improved method to calculate the CU difference return is shown in Equation 17.

$$\Delta G^a_{cu}(s_{t:T}, m_t) = -\mathbb{E}\Big[\mathbb{E}\big[\Delta G_u(s_{t+1:T}, u'_{t+1:T}, M(m_{t:T-1}))); \pi_u(u'_{t+1:T}|o_{t+1:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))\big]; \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})\Big] \tag{17}$$

In this equation, the global difference return is created as the sum of the agent-specific difference returns. This is because the communication policy attempts to improve the difference return for all the agents. However, this assumption is only valid in a cooperative setting and not in a competitive or mixed cooperative-competitive setting.

$$\Delta G_u(s_{t+1:T}, u_{t+1:T}) = \frac{1}{n}\sum_a \Delta G^a_u(s_{t+1:T}, u^a_{t+1:T}) \tag{18}$$

Equation 17 can be unintuitive because of the negative sign, so we provide a formal proof for this equation in Theorem 1 which shows how this equation is obtained.

Theorem 1. *The CU difference reward ΔG^a_{cu} can be defined as the negative expected action difference reward ΔG^a_u.*

Proof.

$$
\begin{aligned}
&\Delta G^a_{cu}(s_{t:T}, m_t) \\
&\quad = G^a_{cu}(s_{t:T}, m_t) \\
&\qquad - \mathbb{E}[G^a_{cu}(s_{t:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))\pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = \mathbb{E}[G_u(s_{t+1:T}, u''_{t+1:T}); \pi_u(u''_{t+1:T}|o_{t+1:T}, M(m_{t:T-1}))] \\
&\qquad - \mathbb{E}[\mathbb{E}[G_u(s_{t+1:T}, u'_{t+1:T}); \pi_u(u'_{t+1:T}|o_{t+1:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \\
&\qquad\quad \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = \mathbb{E}[\sum_{k=t+1}^{T-t+1} \gamma^{k-t} R(s_k, u''_k); \pi_u(u''_{t+1:T}|o_{t+1:T}, M(m_{t:T-1}))] \\
&\qquad - \mathbb{E}[\mathbb{E}[\sum_{k=t+1}^{T-t+1} \gamma_c^{k-t} R(s_k, u'_k); \pi_u(u'_{t+1:T}|o_{t+1:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \\
&\qquad\quad \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = -\mathbb{E}[\mathbb{E}[\sum_{k=t+1}^{T} \gamma^{k-t}(R(s_k, u'_k) - \mathbb{E}[R(s_k, u''_k); \pi_u(u''_k|o_k, M(m_{k-1}))]); \\
&\qquad\quad \pi_u(u'_{t+1:T}|o_{t+1:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = -\mathbb{E}[\mathbb{E}[\sum_{k=t+1}^{T} \gamma^{k-t} \Delta R(s_k, u'_k, M(m_{k-1})); \\
&\qquad\quad \pi_u(u'_{t+1:T}|o_{t+1:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = -\mathbb{E}[\mathbb{E}[\Delta G_u(s_{t+1:T}, u'_{t+1:T}, M(m_{t:T-1})); \\
&\qquad\quad \pi_u(u'_{t+1:T}|o_{t+1:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})]
\end{aligned}
\tag{19}
$$

Next, a memory-improved method to calculate the CC difference return is shown in Equation 20 which shows similar properties to the memory-improved CU difference return calculations by allowing us to discard the CU returns. The proof for this CC difference return equation is shown in Proof 2.

$$
\begin{aligned}
\Delta G^a_{cc}(s_{t:T}, m_t) = -\mathbb{E}\Big[&\mathbb{E}\big[\Delta G_{cu}(s_{t+2:T}, m'_{t+1:T-1}); \\
&\pi_c(m'_{t+1:T-1}|o_{t+1:T-1}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))\big]; \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})\Big]
\end{aligned}
\tag{20}
$$

Theorem 2. *The CC difference reward ΔG^a_{cc} can be defined as the negative expected CU difference reward ΔG^a_{cu}.*

Proof.

$$
\begin{aligned}
&\Delta G^a_{cc}(s_{t:T}, m_t) \\
&\quad = G^a_{cc}(s_{t:T}, m_t) \\
&\qquad - \mathbb{E}[G^a_{cc}(s_{t:T}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))\pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = \mathbb{E}[G_{cu}(s_{t+2:T}, m''_{t+1:T-1}); \pi_c(m''_{t+1:T-1}|o_{t+1:T-1}, \mu_{t+1:T-1})] \\
&\qquad - \mathbb{E}[\mathbb{E}[G_{cu}(s_{t+2:T}, m'_{t+1:T-1}); \\
&\qquad\qquad \pi_c(m'_{t+1:T-1}|o_{t+1:T-1}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \\
&\qquad\quad \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = -\mathbb{E}[\mathbb{E}[G_{cu}(s_{t+2:T}, m'_{t+1:T-1}) - \mathbb{E}[G_{cu}(s_{t+2:T}, m''_{t+1:T-1}); \\
&\qquad\qquad\qquad\qquad\qquad\qquad \pi_c(m''_{t+1:T-1}|o_{t+1:T-1}, \mu_{t+1:T-1})]; \\
&\qquad\qquad \pi_c(m'_{t+1:T-1}|o_{t+1:T-1}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \\
&\qquad\quad \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})] \\
&\quad = -\mathbb{E}[\mathbb{E}[\Delta G_{cu}(s_{t+2:T}, m'_{t+1:T-1}); \\
&\qquad\qquad \pi_c(m'_{t+1:T-1}|o_{t+1:T-1}, M(\langle m'^a_{t:T-1}, m^{-a}_{t:T-1}\rangle))]; \\
&\qquad\quad \pi^a_c(m'^a_{t:T-1}|o^a_{t:T-1}, \mu^a_{t:T-1})]
\end{aligned}
\tag{21}
$$

4.3 Communication Return Approximation

The memory requirements to calculate the communication difference returns have already been reduced by the memory-improved equations. However, calculating the CU and CC difference return can still be very computationally expensive as the expectation is under the joint action and communication policy respectively. This results in $\#M^a * \prod_{a'}^{\boldsymbol{A}} \#U^{a'}$ of inference calls for the CU difference return where $\#M^a$ is the number of messages for agent a and $\#U^{a'}$ is the number of actions for agent a'. In the work of Vanneste *et al.* [20], two approximation methods are compared to the exact method to calculate the CU and CC Q-values with a lower amount of inference calls. The results showed that the Agent-Based Sampling (ABS) approximation has a good balance between training performance and the number of inference calls. An adapted version of the CU return calculations, using the ABS approximation, is shown in Equation 22. The CU Q-value calculations using ABS approximation required $\#M^a * \sum_{a'}^{\boldsymbol{A}} \#U^{a'}$ number of inference calls.

$$
\begin{aligned}
&\Delta G^a_{cu}(s_{t:T}, m_{t:T-1}) = \\
&\quad \frac{1}{n}\sum_{a'=0}^{n} \mathbb{E}\big[G_u(s_{t+1:T}, \langle u'^{a'}_{t+1:T}, \widetilde{u}^{-a'}_{t+1:T}\rangle); \pi^{a'}_u(u'^{a'}_{t+1:T}|o^{a'}_{t+1:T}, Ma'(m_{t:T-1}))\big] \\
&\text{with } :\widetilde{u}^{-a'}_{t+1:T} \sim \pi^{-a'}_u(o^{-a'}_{t+1:T}, \mu_{t+1:T})
\end{aligned}
\tag{22}
$$

The memory-improved CU difference return calculations from Equation 17 can then be combined with Equation 22. This results in the final CU difference return calculation, as shown in Equation 23, which is used for the DR.MACC-ABS method.

$$\begin{aligned}
&\Delta G^a_{cu}(s_{t:T}, m_t) = \\
&\quad - \mathbb{E}\Big[\frac{1}{n}\sum_{a'=0}^{n} \mathbb{E}\big[\Delta G_u(s_{t+1:T}, \langle u'^{a'}_{t+1:T}, \widetilde{u}^{-a'}_{t+1:T}\rangle, M(m_{t:T-1}))); \\
&\qquad \pi_u^{a'}(u'^{a'}_{t+1:T} | o^{a'}_{t+1:T}, Ma'(m_{t:T-1}))\big]; \pi^a_c(m'^a_{t:T-1} | o^a_{t:T-1}, \mu^a_{t:T-1})\Big] \\
&\text{with } :\widetilde{u}^{-a'}_{t+1:T} \sim \pi_u^{-a'}(o^{-a'}_{t+1:T}, \mu_{t+1:T})
\end{aligned} \tag{23}$$

The calculations for the CC return require $\#M^a * \prod_{a'}^{\boldsymbol{A}} \#M^{a'}$ number of inference calls. The CC return ABS approximation calculation reduces the number of inference calls to $\#M^a * \sum_{a'}^{\boldsymbol{A}} \#M^{a'}$ and is shown in Equation 24.

$$\begin{aligned}
&\Delta G_{cc}(s_{t+1:T}, m_{t:T-1}) = \\
&\frac{1}{n}\sum_{a'=0}^{n} \mathbb{E}\big[G_{cu}(s_{t+2:T}, \langle m'^{a'}_{t+1:T-1}, \widetilde{m}'^{-a'}_{t+1:T-1}\rangle); \pi^a_c(m'^{a'}_{t+1:T-1} | o^{a'}_{t+1:T-1}, \mu^{a'}_{t+1:T-1})\big] \\
&\text{with } :\widetilde{m}'^{-a'}_{t+1:T-1} \sim \pi_c^{-a'}(o^{-a'}_{t+1:T}, \mu_{t+1:T})
\end{aligned} \tag{24}$$

Equation 24 is then combined with Equation 20 to achieve Equation 25 which is used in the DR.MACC-ABS method to calculate the CC difference reward.

$$\begin{aligned}
\Delta G^a_{cc}(s_{t:T}, m_t) = -\mathbb{E}\Big[\frac{1}{n}\sum_{a'=0}^{n} \mathbb{E}\big[G_{cu}(s_{t+2:T}, \langle m'^{a'}_{t+1:T-1}, \widetilde{m}'^{-a'}_{t+1:T-1}\rangle); \\
\pi^a_c(m'^{a'}_{t+1:T-1} | o^{a'}_{t+1:T-1}, \mu^{a'}_{t+1:T-1})\big]; \\
\pi^a_c(m'^a_{t:T-1} | o^a_{t:T-1}, \mu^a_{t:T-1})\Big] \\
\text{with } :\widetilde{m}'^{-a'}_{t+1:T-1} \sim \pi_c^{-a'}(o^{-a'}_{t+1:T}, \mu_{t+1:T})
\end{aligned} \tag{25}$$

4.4 Learning the Reward Function

The DR.MACC method assumes that we have access to the environment's reward function. However, we cannot always make this assumption. So, similar to the DRR.PG method, we will learn the reward function and use this to calculate the difference reward. When we learn the reward function for the DR.MACC method, we denote it as the DRR.MACC method. The reward model is trained in a supervised manner using the data generated by the interactions of the agents with the environment. Equation 26 shows the loss function that is used to train the reward function $\bar{R}(s_t, u_t)$. The training of the reward model occurs simultaneously with the training of the agents. It is important to note that learning the

reward function is significantly easier than learning a global Q-function which is required for the MACC method. This is because the reward function does not depend on the policy of the other agents when we have access to the joint action.

$$\mathcal{L}(\theta_r) = \mathbb{E}_{s_t,u_t,r_t}\Big[\big(r_t - \bar{R}(s_t,u_t,\theta_r)\big)^2\Big] \tag{26}$$

4.5 DR.MACC in a Dec-POMDP

Additionally, we also provide the difference return equations which are more suitable for a Dec-POMDP by including a history τ of the past observations and actions. It is important to note that the reward function is only a function of the current state and the current joint action because the reward function is independent of the policy of the agents when we have access to the joint action of the agents.

$$\Delta R^a(o_t,\tau_t,u_t) = r_t - \sum_{u_t'^a} \pi_u^a(u_t'^a|\tau_t^a) R(o_t,\langle u_t'^a, u_t^{-a}\rangle) \tag{27}$$

$$\Delta G_u^a(o_{t:T},\tau_{t:T},u_{t:T}) \triangleq \sum_{k=t}^{T} \gamma^{\,k-t} \Delta R^a(o_k,\tau_k,u_k) \tag{28}$$

These equations can then be used by the memory-improved CU difference return equation as shown in Equation 29.

$$\begin{aligned}\Delta G_{cu}^a(o_{t:T},\tau_{t:T},m_t) = -\mathbb{E}\Big[\mathbb{E}\big[\Delta G_u(o_{t+1:T},\tau_{t:T},u'_{t+1:T},M(m_{t:T-1}))\big);\\ \pi_u(u'_{t+1:T}|\tau_{t+1:T},M(\langle m_{t:T-1}'^a, m_{t:T-1}^{-a}\rangle))\big];\pi_c^a(m_{t:T-1}'^a|\tau_{t:T-1}^a,\mu_{t:T-1}^a)\Big]\end{aligned} \tag{29}$$

Similarly, the memory-improved CC difference return equation can be adapted to include the history as shown in Equation 30.

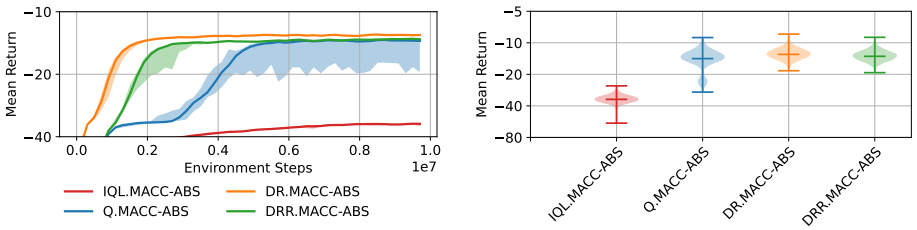
$$\begin{aligned}\Delta G_{cc}^a(o_{t:T},\tau_{t:T},m_t) = -\mathbb{E}\Big[\mathbb{E}\big[\Delta G_{cu}(o_{t+2:T},\tau_{t+2:T},m'_{t+1:T-1});\\ \pi_c(m'_{t+1:T-1}|\tau_{t+1:T-1},M(\langle m_{t:T-1}'^a, m_{t:T-1}^{-a}\rangle))\big];\pi_c^a(m_{t:T-1}'^a|\tau_{t:T-1}^a,\mu_{t:T-1}^a)\Big]\end{aligned} \tag{30}$$

5 Results

In this section, the DR.MACC and DRR.MACC methods are compared to the Q-variant of MACC [20], using a centralized critic, which we will denote as Q-learning MACC (Q.MACC). Additionally, we also compared with an independent variant of MACC which uses an independent Q-function. Here, every agent uses a decentralized critic based on its observation and action. This variant of the MACC method is denoted as Independent Q-Learning MACC (IQL.MACC).

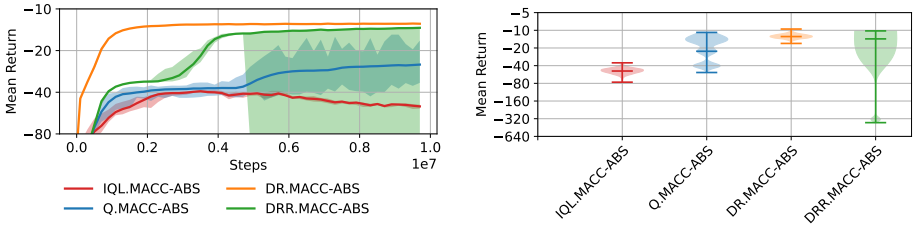
The IQL.MACC variant allows us to compare both the centralized and decentralized critic variants. This variant enables us to demonstrate the importance of a centralized critic in learning to communicate because this is not the case for every environment as described by Lyu *et al.* [8]. The different methods all use the Agent-Based Sampling (ABS) method [20] to reduce the computational cost. We used the speaker with multiple listener's environment from the work of Vanneste *et al.* [20] which is based on the speaker listener setting from the Particle environment [7]. In this environment, n listeners need to go towards one of three target landmarks in a 2D world. However, the listeners do not know to which landmark they need to go. This information is only available to the speaker, so it needs to share this information with the listener agents by learning a certain communication protocol. The speaker agent can use two bits of information to share this information with the listener agents. The agents are trained using a shared reward which is the negative average distance of the different listeners to their target landmark. This means that a set of optimal agents can achieve an average return of -15 independently of the number of listeners in the environment. No specific reward is presented to the agents for sharing information between the speaker and listeners, so the agents need to learn a communication protocol to minimize the distance between the listeners and their target landmarks. This environment allows us to easily test the scalability of the different methods by increasing the number of listeners and comparing the average return. We tested the different methods in three different configurations with One, two and four listeners. Additionally, the difference reward variants are also evaluated with eight listeners. In our experiments, we used the same agent configuration and implementation for the different methods and only adapt the method to calculate the advantage or difference reward. The different agents are trained under the CTDE paradigm, so we allowed the agents to share network parameters during training. Every method and number of listeners combination is trained five times after which the results are combined into a training graph and violin graph of the average return of the final episodes. The data of the training graphs are processed to show the interquartile mean and the bootstrapped 90% confidence interval. These results are shown in Figures 3, 4, 5, and 6.

The results show that the IQL.MACC variant is not able to learn a communication policy for any number of listeners. This is because a decentralized critic is not able to properly estimate the expected discounted reward without the observation of the listener. The Q.MACC can learn a valid communication protocol in the configuration with one listener because an average return of -15 is achieved. However, when the number of listeners is increased to two, the Q.MACC method cannot achieve this. When we increase the number of listeners to four, the Q.MACC method cannot learn a valid action policy and achieves an average return of -400 in the final episodes. The results of the DRR.MACC method, which uses a learned reward function, can outperform the Q.MACC method in every tested configuration. Although the DRR.MACC method achieves better results than the Q.MACC method, the method cannot learn a set of valid policies for the configuration with eight listeners where it achieves an average



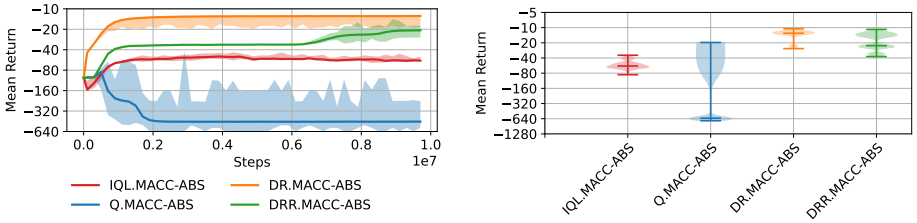
(a) Training Graph (b) Return of the Final Episodes

Fig. 3: Speaker with One Listener.



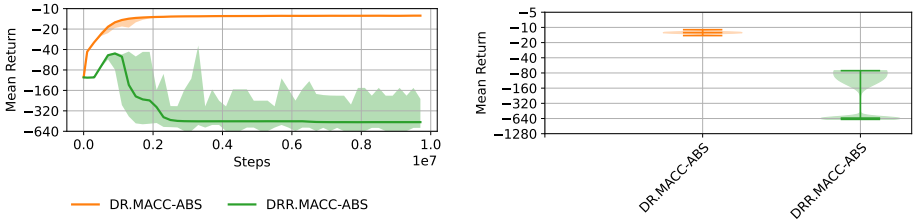
(a) Training Graph (b) Return of the Final Episodes

Fig. 4: Speaker with Two Listeners.



(a) Training Graph (b) Return of the Final Episodes

Fig. 5: Speaker with Four Listeners.



(a) Training Graph (b) Return of the Final Episodes

Fig. 6: Speaker with Eight Listeners.

return of -400. The DR.MACC method, which uses the environment's reward function, can learn a valid communication policy for every configuration. These results show that the main factor which determines the agents' performance, is the accuracy of the Q-function or Reward function. Since the reward function is much easier to learn, the DRR.MACC method can learn with a higher number of agents compared to the Q.MACC method. When using the reward function directly, which is perfectly accurate by definition, the DR.MACC method can learn within all the tested configurations.

6 Conclusion

In this work, we presented the DR.MACC and the DRR.MACC methods. These methods combine the benefits of two methods. First, the MACC method [20] enables us to learn a discrete communication policy by using counterfactual reasoning over the policy of the other agents. Next, the DR.PG method [1] allows us to calculate an agent-specific return by the reward function or a trained approximation instead of the more complex Q-function. To reduce the memory requirements of the computations, we provided a set of memory-improved equations for the communication difference return which can reuse the action difference return directly. However, these calculations still depend on the expectation of the joint action and joint communication policy of the agents which is very computationally expensive using the exact method. To reduce this computational cost to a great extent, we used the Agent-Based Sampling (ABS) method from the work of Vanneste *et al.* [20]. The results show that the IQL.MACC method, using the individual Q-function, is not able to learn a communication protocol because the critic cannot access the observation of the speaker. However, the Q.MACC method can learn a communication protocol for one listener but encounters scalability problems in the configuration with two and four listeners because of the need to learn a joint Q-function. The DRR.MACC method outperforms the Q.MACC method in every tested configuration but cannot learn a set of policies for the eight listener's environment. The DR.MACC method, using the reward function directly, can learn a valid action and communication policy in every tested configuration. In future work, the DR.MACC and DRR.MACC methods could be evaluated in a Dec-POMDP using the adaptions to the difference return as presented in Section 4.5. Additionally, other approximation methods to calculate the communication difference return could be investigated.

Acknowledgements

Simon Vanneste and Astrid Vanneste are supported by the Research Foundation Flanders (FWO) under Grant Number 1S94120N and Grant Number 1S12121N respectively.

References

1. Castellini, J., Devlin, S., Oliehoek, F.A., Savani, R.: Difference rewards policy gradients (2021)
2. Chang, Y.H., Ho, T., Kaelbling, L.: All learning is local: Multi-agent learning in global reward games. Advances in neural information processing systems **16** (2003)
3. Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., Pineau, J.: Tarmac: Targeted multi-agent communication. In: International Conference on Machine Learning. pp. 1538–1546. PMLR (2019)
4. Foerster, J., Assael, I.A., De Freitas, N., Whiteson, S.: Learning to communicate with deep multi-agent reinforcement learning. Advances in neural information processing systems **29** (2016)
5. Foerster, J., Farquhar, G., Afouras, T., Nardelli, N., Whiteson, S.: Counterfactual multi-agent policy gradients. In: Proceedings of the AAAI conference on artificial intelligence. vol. 32 (2018)
6. Jaques, N., Lazaridou, A., Hughes, E., Gulcehre, C., Ortega, P., Strouse, D., Leibo, J.Z., De Freitas, N.: Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In: International conference on machine learning. pp. 3040–3049. PMLR (2019)
7. Lowe, R., Wu, Y.I., Tamar, A., Harb, J., Pieter Abbeel, O., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. Advances in neural information processing systems **30** (2017)
8. Lyu, X., Xiao, Y., Daley, B., Amato, C.: Contrasting centralized and decentralized critics in multi-agent reinforcement learning (2021)
9. Mandhane, A., Zhernov, A., Rauh, M., Gu, C., Wang, M., Xue, F., Shang, W., Pang, D., Claus, R., Chiang, C.H., et al.: Muzero with self-competition for rate control in vp9 video compression. arXiv preprint arXiv:2202.06626 (2022)
10. Mnih, V., Badia, A.P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., Kavukcuoglu, K.: Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. pp. 1928–1937. PMLR (2016)
11. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. nature **518**(7540), 529–533 (2015)
12. Oliehoek, F.A., Amato, C.: A concise introduction to decentralized POMDPs. Springer, Switzerland (2016)
13. Oliehoek, F.A., Vlassis, N.: Q-value functions for decentralized pomdps. In: Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems. pp. 1–8 (2007)
14. Van der Pol, E., Oliehoek, F.A.: Coordinated deep reinforcement learners for traffic light control. Proceedings of Learning, Inference and Control of Multi-Agent Systems (at NIPS 2016) (2016)
15. Silver, D., Lever, G., Heess, N., Degris, T., Wierstra, D., Riedmiller, M.: Deterministic policy gradient algorithms. In: International conference on machine learning. pp. 387–395. PMLR (2014)
16. Sukhbaatar, S., Fergus, R., et al.: Learning multiagent communication with backpropagation. Advances in neural information processing systems **29** (2016)
17. Sutton, R.S., Barto, A.G.: Introduction to Reinforcement Learning. MIT Press, Cambridge, MA, USA, 1st edn. (1998)
18. Van Hasselt, H., Guez, A., Silver, D.: Deep reinforcement learning with double q-learning. In: Proceedings of the AAAI conference on artificial intelligence. vol. 30 (2016)

19. Vanneste, S., de Borrekens, G., Bosmans, S., Vanneste, A., Mets, K., Mercelis, S., Latré, S., Hellinckx, P.: Learning to communicate with reinforcement learning for an adaptive traffic control system. In: Advances on P2P, Parallel, Grid, Cloud and Internet Computing: Proceedings of the 16th International Conference on P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC-2021). pp. 207–216. Springer International Publishing (2022)
20. Vanneste, S., Vanneste, A., Mets, K., De Schepper, T., Anwar, A., Mercelis, S., Latré, S., Hellinckx, P.: Learning to communicate using counterfactual reasoning. In: Proc. of the Adaptive and Learning Agents Workshop (ALA 2022). Adaptive and Learning Agents Workshop (ALA 2022) (2022)
21. Wolpert, D.H., Tumer, K.: Optimal payoff functions for members of collectives. Advances in Complex Systems **4**(02n03), 265–279 (2001)