Mixed Cooperative-Competitive Communication Using Multi-Agent Reinforcement Learning

Astrid Vanneste, Wesley Van Wijnsberghe, Simon Vanneste, Kevin Mets, Siegfried Mercelis, Steven Latré, Peter Hellinckx

Abstract By using communication between multiple agents in multi-agent environments, one can reduce the effects of partial observability by combining one agent's observation with that of others in the same dynamic environment. While a lot of successful research has been done towards communication learning in cooperative settings, communication learning in mixed cooperative-competitive settings is also important and brings its own complexities such as the opposing team overhearing the communication. In this paper, we apply differentiable inter-agent learning (DIAL), designed for cooperative settings, to a mixed cooperative-competitive setting. We look at the difference in performance between communication that is private for a team and communication that can be overheard by the other team. Our research shows that communicating agents are able to achieve similar performance to fully observable agents after a given training period in our chosen environment. Overall, we find that sharing communication across teams results in decreased performance for the communicating team in comparison to results achieved with private communication.

1 Introduction

Research has demonstrated that single-agent reinforcement learning (RL) can successfully be applied to tackle challenging problems in a variety of applications [10, 14]. However, several applications (e.g. multiplayer games) require interaction

Astrid Vanneste, Wesley Van Wijnsberghe, Simon Vanneste, Kevin Mets, Siegfried Mercelis, Steven Latré, Peter Hellinckx
University of Antwerp - imec, IDLab, Faculty of Applied Engineering, Belgium, Sint-Pietersvliet 7, 2000 Antwerpen, e-mail: astrid.vanneste@uantwerpen.be, wesley.vanwijnsberghe@student.uantwerpen.be, simon.vanneste@uantwerpen.be, kevin.mets@uantwerpen.be, siegfried.mercelis@uantwerpen.be, steven.latre@uantwerpen.be, peter.hellinckx@uantwerpen.be

between multiple agents in order to reach a certain goal. The tasks that we consider in our application environment are mixed cooperative-competitive, partially observable multi-agent decision making problems in which the agents of a team share a common goal. These goals are competitive across teams. Due to the joint observability of the underlying Markov state, each individual agent only knows its own observation and consequently lacks knowledge of the full state of the dynam+ic environment. With a view to optimise the decision-making process of each individual member of a team, we need a technique for these agents to share their state information with each other in order to achieve a greater knowledge about the full underlying state. Recently, interest has been directed to a research domain within multi-agent reinforcement learning (MARL) that opens the door to a new perspective on coordinated behaviour of multiple RL agents in a dynamic environment [3, 15]. This domain focuses its research on agents that use machine learning principles to learn to communicate amongst themselves while simultaneously learning the behaviour needed to achieve a pre-defined common goal. In real-world environments, individual agents generally do not have full observability of the surrounding environment. Therefore, it is essential that these agents learn a protocol to share their individual observations, so that each agent can generate a more complete representation of the environment. The more information an agent has about its immediate environment, the better it is able to make good decisions in order to achieve the common goal. The aim of this paper is to investigate current state-of-the-art communication learning methods in a mixed cooperative-competitive setting. We want to look into the effects of sharing these communication messages with the opposing team. We continue this paper by discussing the state of the art on multi-agent communication in section 2. Section 3 contains some background information regarding RL. In section 4, we describe the used methods. Our experiments are presented in section 5. We discuss the results of our research in section 6. To finalise this paper, we discuss our conclusions and some future work regarding multi-agent communication in section 7.

2 Related Work

In this section, we focus on the related work concerned with mixed cooperative-competitive MARL and the learning of communication protocols in MARL.

Research within MARL is mostly focused on cooperative tasks. However, Lowe et al. [8] presented a set of tasks which are fully cooperative or mixed cooperative-competitive. They present their method, multi-agent deep deterministic policy gradient (MADDPG). This method is able to tackle these tasks, including some communication learning tasks.

Foerster et al. [3] and Sukhbaatar et al. [15] presented the foundations in the field of communication learning. Sukhbaatar et al. [15] explores a neural model, called CommNet, that uses continuous communication for fully cooperative tasks. In this model, communication between agents is achieved by sharing the agents'

internal hidden states with the other agents. Agents learn a communication protocol by allowing gradients to flow through the communication channel. This results in improved performance over non-communicative agents. Two different approaches for learning discrete communication are proposed by Foerster et al. [3]: Reinforced Inter-Agent Learning (RIAL) and Differentiable Inter-Agent Learning (DIAL). RIAL uses deep Q-learning [10] for both the action and the communication policy. There is no direct feedback from other agents on the communication of a certain agent, which makes it difficult for the agents to agree on a communication policy. DIAL addresses this limitation by allowing gradients to flow between agents through the communication channel. The gradients give richer feedback, reduce the required amount of learning by trial and error, and ease the discovery of effective protocols. They are able to learn discrete communication protocols by applying a discretize regularize unit (DRU) on the messages. Following these fundamental works of Sukhbaatar et al. and Foerster et al. more research in the field of communication learning emerged. BiCNet[13] and ACCNet [9] are both actor-critic approaches to communication learning. Vanneste et al. [18] combined the work of Foerster et al. [3] and Sunehag et al. [16] to improve the performance of DIAL. MACC [19] is an approach with a centralized critic that applies counterfactual reasoning to learn to communicate. Recently, there has also been more attention to directional communication [1][2][6] .

3 Background

In this section we discuss some required background information on RL. In single-agent RL, the agent's environment can be described by a Markov decision process (MDP)[5]. A MDP is a tuple (S, U, f, ρ) where S is the set of environment states s, U is the set of agent actions u, $f : S \times U \times S \rightarrow [0, 1]$ is the state transition probability function, and $\rho : S \times U \rightarrow \mathbb{R}$ is the reward function. As a result of the action $u_t \in U$, the environment changes its state from s_t to some $s_{t+1} \in S$ according to the state transition probabilities defined by f. After this transition, the agent receives a scalar reward $r_{t+1} \in \mathbb{R}$ according to the reward function ρ. A policy π describes the behaviour of the agent, and therefore specifies how the agent chooses its actions given the observed state. The task of the agent is to learn a policy to maximize its long-term expected reward. When agents do not receive the full state of the environment but only a limited state observation o, the multi-agent system is described using a partially observable MDP (POMDP).

MARL problems where all agents receive the full state are modelled using multi-agent MDPs (MMDP). In contrast, problems where agents only receive a limited state observation are described by Oliehoek et al. [12] as decentralized POMDPs (dec-POMDP). Decentralized MDPs (dec-MDP) are a subset of dec-POMDPs for jointly observable problems. In these jointly observable problems, the full state s_t can be composed by combining all individual observations of the agents [4] [12]. In our experiments we try to tackle these dec-MDPs, since the agents in our multi-agent

mixed cooperative-competitive setup lack the necessary information to achieve their goal, but are able to generate full observability by exchanging state information.

4 Methods

In this section, we present our approach for learning communication protocols in a mixed cooperative-competitive environment where two teams compete against each other. In our experiments we will be using a predator-prey environment where the predators need to capture a specific prey. Only the other predator knows which prey needs to be targeted. This results in a need for communication. The environment is explained in more detail in Section 5.1. The agents of the two teams are different. The team of predators will be able to communicate while the team of prey are not.

4.1 Prey Agents

The team of prey agents is not allowed to communicate. We opted to train our prey agents with independent Q-learing (IQL)[17]. All prey agents share the same parameters, which are synchronized after each epoch, to achieve better results. The prey agents will not be able to gain full observability if they cannot overhear the communication of the predators since they do not know which of the predators is targeting them.
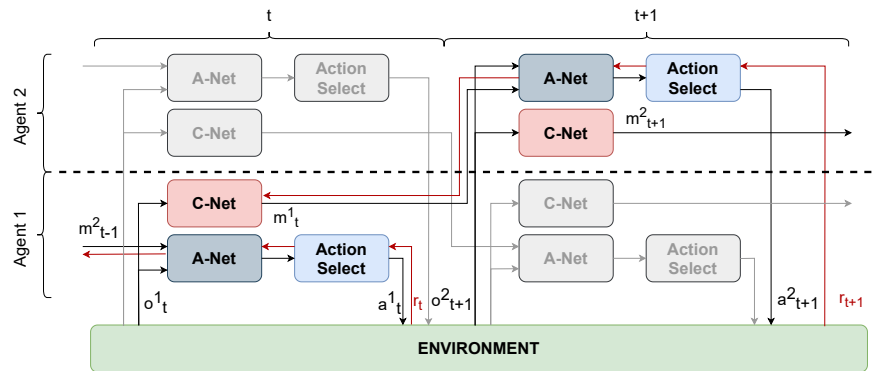
4.2 Predator Agents

Fig. 1: DIAL without DRU and with a separate action and communication policy

The team of predator agents will be able to communicate with each other. The method we use is inspired by DIAL as described by Foerster et al. [3] and can be seen in Fig. 1. The key insight behind DIAL is the possibility of pushing gradients from one agent to another through the communication channel. The gradient flow from one agent to the other acts as a form of feedback, which does not only reduce the amount of learning by trial and error but also eases the discovery of effective communication protocols. Comparing our method to the original method presented by Foerster et al., there are two main difference. First, we have split the neural network of the action and the communication policy. This allows us to use a very simple network for the communication policy, making it easier to learn. Secondly, we chose not to include the DRU, since we observed better results when using a ReLU activation instead. This causes communication to be continuous at both execution and training time. The communication network, which is called the C-Net, is a neural network which outputs a vector message based on the environment observation it receives as input. The vector message is passed to the other agent. The action network, called the A-Net, calculates the Q-values based on the received vector message from its teammate and the observation received from the environment. These Q-values are passed on to the action select, which selects the optimal action based on these Q-values. The predator agents share their parameters to improve performance. The parameters are synchronized after each epoch.

5 Experiments

In order to evaluate our approach, we perform a set of experiments in a mixed cooperative-competitive environment. In this section we discuss the specification of the constructed environment and our agent setup.

5.1 Environment

In our research we use a combination of the "simple tag" and "simple reference" environments provided by OpenAi [8, 11] to represent our mixed cooperative-competitive multi-agent environment. The environment is a predator-prey environment where the predators have to catch a specific prey and the prey have to avoid the predator that is targeting them. In our experimental setup, only the two predator agents will be able to communicate with each other. This communication is a key factor in their performance, since every predator agent is assigned a specific enemy target, which is only known by its teammate. In order for these agents to reach the common goal of getting as close as possible to their respective target prey, they have to learn a way to communicate the targets to each other. The goal of the prey agents is to maximise their combined distance from the predators assigned to catch

them. Since there are two possible targets for the predator agents, the size of the communication space is one.

5.2 Model Architecture

The predators use the C-Net to learn the message policy. The C-Net is a fully connected neural network with no hidden layers. In our setup we do not need hidden layers, because our agents have to directly communicate certain information from their observation. It accepts an observation input and outputs the vector message. This network is effectively a linear mapping between the input and output.

The A-net accepts an observation vector input from the environment and a message vector input from the agent's teammate. The inputs are concatenated and passed through two fully connected hidden layers with a size of 256 and 512 respectively. This network is the only network used by the prey agents because they are not required to learn communication.

We perform our experiments in the RLlib framework [7]. In our experiments we perform 30 steps in a single episode. The learning rate is set at 0.0005, while we choose a discount factor of 0.97. The networks are updated every epoch, consisting of 50 episodes. We use a training batch size of 200.

5.3 Experimental Setup

In order to evaluate the effectiveness of the learned communication, we choose to train our agents in four distinct configurations.

- **No Communication**: The predator agents are not allowed to learn a communication protocol. They have to base their actions solely on the environmental observation, which lacks target information for a single predator agent. Here, both teams will be trained using IQL[17] with a partially observable environment.
- **Full Observability**: Instead of observing the teammate target, predators observe their own target. They have all the necessary information required to maximize the team reward without the need to communicate with its teammate. Since there is no need for communication, both teams are trained using IQL [17].
- **Private Communication**: The predator agents communicate with each other, while the prey agents are not able to observe this communication.
- **Public Communication**: The predator agents communicate with each other, while the prey agents can hear this communication. They effectively observe the communication of the predators and if they are able to use the predator communication to learn which prey agent(s) is/are targeted during each episode, they can use this extra information to increase their performance. This is the only setting where the prey are able to determine which prey is being targeted by which predator.

6 Results

In this section, we evaluate the results of the four proposed experiments from the previous section. Because of the fact that only a single team, namely the team of predators, is able to learn communication, we choose to concentrate on their results. In each of our experimental setups the prey merely act as an opponent in order to fulfill the competitive aspect of our research. Across most experiments, they share the same configuration apart from the prey in the public communication experiment, who differ by the ability to observe the predator communication. Due to the competitive setup of our environment and the way the rewards are calculated, the performance of the prey is inversely proportional to the performance of the predators. In Fig. 2 the results from five training runs for the mean predator rewards are shown. It shows the average reward received during an episode per epoch for the entire training run. We choose to smooth the results by applying an exponentially weighted moving average filter with an alpha value of 0.0005 in order to ease the comparison of the different experiments. Table 1 shows the average reward and the average standard deviation across the five training runs. We see that overall the full observability setting performs best with the private communication settings achieving very similar results. They both also show the lowest standard deviation. The public communication setting however shows a much worse performance, close to the performance of the no communication setting. However, the public communication setting is able to perform much better than the no communication setting when we look at the average peak performance accross the different training run of each of the settings. The peak performance shows the maximum reward that is achieved in a training run. This can be seen in Table 2. We see that the private communication setting performs best closely followed by the full observability setting. However, the public communication setting performs a lot better than the no communication setting. This is to be expected since the predators are able to communicate about their target, but they cannot perform as well as in the private communication setting because the prey can overhear the communication and determine which of the predators is targeting them.

	Average Reward	Standard Deviation
Full Observability	-50.93	27.65
No Communication	-119.81	69.81
Private Communication	-64.49	42.05
Public Communication	-129.20	75.69

Table 1: Overview of the average reward and the average standard deviation across the five training runs for each of the experiment settings

Since there is a significant difference in performance between the private and public communication settings, it is interesting to look at their communication policies. In Fig. 3, the communication protocols are presented as a confusion matrix. In this confusion matrix, we have discretised the communication by applying a thresh-
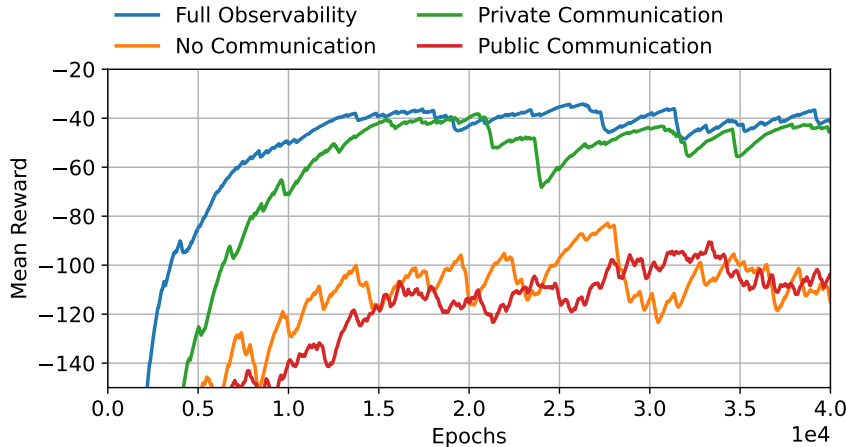
Fig. 2: Mean reward of the predators for each of the experiment settings smoothed with an exponentially weighted moving average filter with an alpha value of 0.0005.

	Average Peak Performance	Standard Deviation of the Peak Performance
Full Observability	-18.6	1.02
No Communication	-43.6	1.96
Private Communication	-17.6	0.49
Public Communication	-24.0	1.26

Table 2: Overview of the average peak performance and the standard deviation on the average peak performance across the five training runs for each of the experiment settings

old at zero for the messages. Here, we can see which message is sent in function of which prey the other predator has to target. We can clearly see that in both settings the agents are able to learn a valid communication protocol. Therefore, we can conclude that the difference in performance is not caused by an invalid communication protocol in the public communication setting but by the fact that the prey can overhear communication.

7 Conclusion and Future Work

In this paper, we investigated the effect of learning to communicate in mixed cooperative-competitive environments where it is possible for communication to be heard by the opposing team. In order to perform the required research, we use a technique for agents of the same team to learn a communication protocol inspired
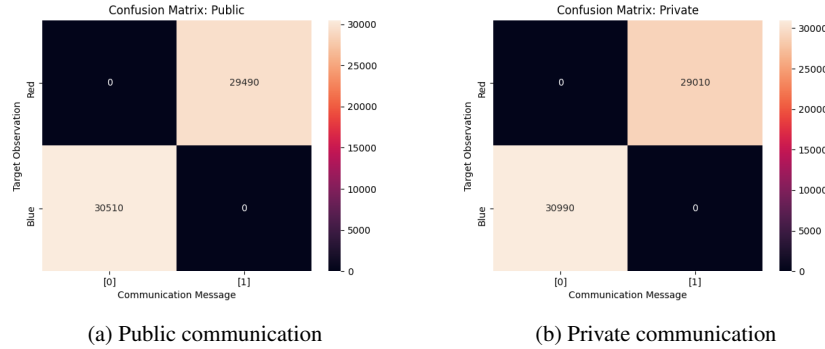
(a) Public communication　　　　(b) Private communication

Fig. 3: Confusion matrix from the communication policy of the predator agents

by the state of the art as presented by Foerster et al. [3]. This technique was tested in a mixed cooperative-competitive environment that acted as a predator-prey environment where the agents are rewarded based on the distance to their respective target. We compare the achieved results using this proposed technique for four distinct configurations. We can conclude that the predators in the private experiment are able to achieve a similar performance to predators with full observability of the environment. When looking at the predators at peak performance, we acquired the knowledge that the predators in the public setting performed worse than in the private setting. Our experiments showed that using DIAL [3], predator agents are able to learn a valid communication protocol for both public and private configurations. This leads to the conclusion that in scenarios where communication is shared across teams, performance will decline significantly. In the future, it would be interesting to look at a scenario with more agents, since we limited the size of the teams to two agents. In our results, we saw that public communication makes it easier for the opponents to anticipate on the actions of the agents. To prohibit this, it would be interesting to look at techniques that will allow the agents to encrypt their messages and to reason about the behaviour of opposing agents as a reaction to the communication.

References

1. Das, A., Gervet, T., Romoff, J., Batra, D., Parikh, D., Rabbat, M., Pineau, J.: Tarmac: Targeted multi-agent communication (2018)

2. Ding, Z., Huang, T., Lu, Z.: Learning individually inferred communication for multi-agent cooperation (2021)
3. Foerster, J., Assael, I.A., De Freitas, N., Whiteson, S.: Learning to communicate with deep multi-agent reinforcement learning. In: Advances in neural information processing systems, pp. 2137–2145 (2016)
4. Goldman, C.V., Zilberstein, S.: Optimizing information exchange in cooperative multi-agent systems. In: Proceedings of the second international joint conference on Autonomous agents and multiagent systems, pp. 137–144 (2003)
5. Howard, R.A.: Dynamic Programming and Markov Processes. MIT Press, Cambridge, MA (1960)
6. Jiang, J., Lu, Z.: Learning attentional communication for multi-agent cooperation. In: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett (eds.) Advances in Neural Information Processing Systems 31, pp. 7254–7264. Curran Associates, Inc. (2018). URL http://papers.nips.cc/paper/7956-learning-attentional-communication-for-multi-agent-cooperation.pdf
7. Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Goldberg, K., Gonzalez, J.E., Jordan, M.I., Stoica, I.: RLlib: Abstractions for distributed reinforcement learning. In: International Conference on Machine Learning (ICML) (2018)
8. Lowe, R., Wu, Y., Tamar, A., Harb, J., Abbeel, P., Mordatch, I.: Multi-agent actor-critic for mixed cooperative-competitive environments. Neural Information Processing Systems (NIPS) (2017)
9. Mao, H., Gong, Z., Ni, Y., Xiao, Z.: Accnet: Actor-coordinator-critic net for "learning-to-communicate" with deep multi-agent reinforcement learning (2017)
10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidjeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., Hassabis, D.: Human-level control through deep reinforcement learning. Nature **518**, 529–33 (2015). DOI 10.1038/nature14236
11. Mordatch, I., Abbeel, P.: Emergence of grounded compositional language in multi-agent populations. arXiv preprint arXiv:1703.04908 (2017)
12. Oliehoek, F.A., Amato, C., et al.: A concise introduction to decentralized POMDPs, vol. 1. Springer (2016)
13. Peng, P., Yuan, Q., Wen, Y., Yang, Y., Tang, Z., Long, H., Wang, J.: Multiagent bidirectionally-coordinated nets for learning to play starcraft combat games. CoRR **abs/1703.10069** (2017). URL http://arxiv.org/abs/1703.10069
14. Silver, D., Huang, A., Maddison, C., Guez, A., Sifre, L., Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., Hassabis, D.: Mastering the game of go with deep neural networks and tree search. Nature **529**, 484–489 (2016). DOI 10.1038/nature16961
15. Sukhbaatar, S., Fergus, R., et al.: Learning multiagent communication with backpropagation. In: Advances in neural information processing systems, pp. 2244–2252 (2016)
16. Sunehag, P., Lever, G., Gruslys, A., Czarnecki, W.M., Zambaldi, V., Jaderberg, M., Lanctot, M., Sonnerat, N., Leibo, J.Z., Tuyls, K., Graepel, T.: Value-decomposition networks for cooperative multi-agent learning based on team reward. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '18, p. 2085–2087. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC (2018)
17. Tan, M.: Multi-agent reinforcement learning: Independent vs. cooperative agents. In: In Proceedings of the Tenth International Conference on Machine Learning, pp. 330–337. Morgan Kaufmann (1993)
18. Vanneste, S., Vanneste, A., Bosmans, S., Mercelis, S., Hellinckx, P.: Learning to communicate with multi-agent reinforcement learning using value-decomposition networks. In: L. Barolli, P. Hellinckx, J. Natwichai (eds.) Advances on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 736–745. Springer International Publishing, Cham (2020)
19. Vanneste, S., Vanneste, A., Mercelis, S., Hellinckx, P.: Learning to communicate using counterfactual reasoning (2020)