# Cooperative Multi-Agent Deep Reinforcement Learning in Soccer Domains

Extended Abstract

Jim Martin Catacora Ocana
Sapienza University of Rome
Rome, Italy
catacora@diag.uniroma1.it

Francesco Riccio
Sapienza University of Rome
Rome, Italy
riccio@diag.uniroma1.it

Roberto Capobianco
Sapienza University of Rome
Rome, Italy
capobianco@diag.uniroma1.it

Daniele Nardi
Sapienza University of Rome
Rome, Italy
nardi@diag.uniroma1.it

## ABSTRACT

In multi-robot reinforcement learning the goal is to enable a group of robots to learn coordinated behaviors from direct interaction with the environment. Here, we provide a comparison of two main approaches designed for tackling this challenge; namely, independent learners (IL) and joint-action learners (JAL). We evaluate these methods in a multi-robot cooperative and adversarial soccer scenario, called 2 versus 2 free-kick task, with simulated NAO humanoid robots as players. Our findings show that both approaches can achieve satisfying solutions, with JAL outperforming IL.

## KEYWORDS

Multi-Robot; Deep Reinforcement Learning; Robot Soccer

**ACM Reference Format:**
Jim Martin Catacora Ocana, Francesco Riccio, Roberto Capobianco, and Daniele Nardi. 2019. Cooperative Multi-Agent Deep Reinforcement Learning in Soccer Domains. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

## 1 INTRODUCTION

Multi-agent reinforcement learning (MARL) is concerned with the application of reinforcement learning (RL) techniques to situations having multiple agents learning at the same time in the same environment. MARL has the potential to handle challenging domains involving robot teams or swarms [1–3, 6, 7, 9, 10, 14].

Two main MARL approaches have been proposed for handling multi-agent domains. In independent learners (IL), every agent performs standard RL, but in the presence of other agents. IL has the drawback that each individual sees the environment as non-stationary; and hence, guarantees of single-agent RL do not longer hold. Meanwhile, in joint-action learners (JAL), the state and action spaces of all agents are merged together, defining a sort of super agent in the joint-space. Any single-agent RL algorithm can readily learn the optimal joint-policy for such super agent. JAL overcomes the non-stationarity problem, but it presents scalability issues.

*Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.* 

Here, we implement IL and JAL and compare them on a robotic task. We consider a simplified version of soccer, referred to as 2 versus 2 offensive free-kick task. In soccer domains, the number of agents is normally small; hence, JAL is likely to converge within a reasonable amount of time, unlike domains with hundreds or more agents where JAL is impractical. In such domains, JAL may still be slower than IL, but it may also find better policies. Thus, they provide a motivating opportunity for investigating a tradeoff between optimality and convergence rate across MARL approaches.

As contributions: 1) we provide new satisfactory results from IL and JAL, whereas a previous work [4] did not learn coordinated behaviors with neither approach in a similar task (2 versus 1 half-field offense), 2) we compare the performances of IL and JAL given these results, confirming that JAL should not be overlooked when facing analogous domains, and 3) we carry out experiments on a physically realistic 3D simulator, whereas, as far as we know, all previous studies concerning MARL in soccer domains involved simplified 2D environments.

## 2 METHODOLOGY

*Task Specification.* Experiments were carried out within a RoboCup Standard Platform League [11] simulator, called the B-Human framework [12], which allows teams of realistically simulated NAO humanoid robots to compete against each other.

The offensive free-kick task involves two teams; an offensive (2 attackers) and a defending (defender and keeper) team. The game takes place in the half-field belonging to the latter team. An episode starts with the offensive team being granted a free kick. The attackers' goal is to score a goal within a time limit without losing control of the ball. Only the attackers are allowed to learn a strategy, whilst the defending players follow handcrafted policies.

*Markov Decision Process (MDP) for IL.* As all agents have homogeneous state and action spaces, we formulate one MDP [5] and learn a shared policy from their combined experiences. This MDP comprises an 18-dimensional state vector consisting of an agent's and its teammate's poses, the ball's position and velocity, the defender's position, the keeper's y-coordinate and a 5-bit timestamp. Such high-level information is supplied by the simulator.

The action space of the MDP is represented by a 5-dimensional real-valued vector. Two components correspond to action selectors

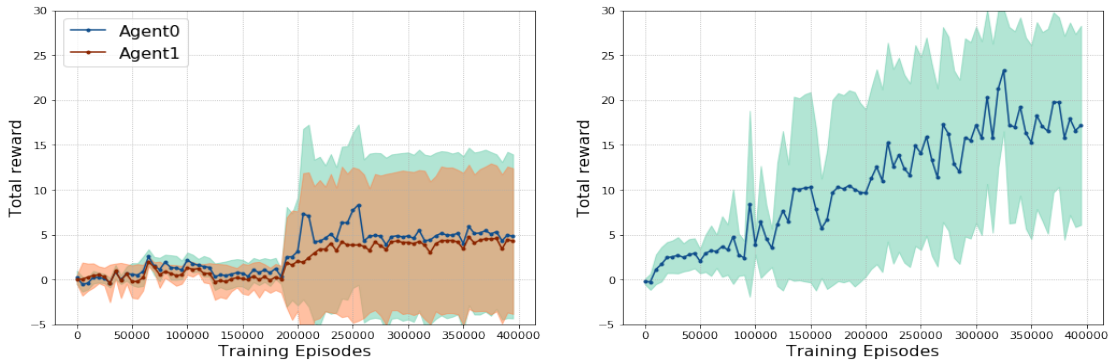


**Figure 1: Mean and standard deviation over 10 runs of the total reward per episode reported by a) IL (left) and b) JAL (right).**

for the only two macro actions a learning agent can execute, namely: walk and kick, which come pre-built with the simulator. The three remaining components specify walking velocities. Both agents are independently rewarded given the same function shown below:

$$R = D_A^B + max(D_B^T, D_B^G) + G \quad (1)$$

Where, $D_A^B$ computes the difference between the distances from an agent A's previous and current positions to the ball B's current position, $D_B^T$ computes the difference between the distances from B's previous and current positions to T's current position (A's teammate), $D_B^G$ computes the difference between the shortest distances from B's previous and current positions to the goal line, and $G$ assigns +20 to all learners only when a goal is scored. $D_B^T$ and $D_B^G$ return zero if A is not responsible for the ball's motion.

*Stochastic Game (SG) for JAL.* Since full observability is assumed in both formulations, the state space of the SG [13] associated with the free-kick task is also 18-dimensional, embedding the same components as the previous MDP. The joint-action space (10D) is constructed by concatenating each agent's individual 5-dimensional action vector, as presented in the precedent sub-section.

The offensive team as a whole is rewarded according to the following function, whose terms mirror those in Equation 1; and where, subscript $A_i$ refers to the i-th agent in the offensive team.

$$R = \sum_i D_{A_i}^B + max(\{D_B^{A_i}, \forall i\}, D_B^G) + G \quad (2)$$

*Deep RL Settings.* The DDPG algorithm [8] is used in IL as well as in JAL. Critics receive as input the concatenation of state and action vectors of the corresponding MDP or SG, while actors are fed only state vectors. Sequential neural networks with 3 hidden layers are employed in all cases. Training relies on the L2-loss and Adam optimizer. Hyper-parameters are set as shown in Table 1.

## 3 RESULTS

IL and JAL were executed 10 runs each over the offensive free-kick task. Each run continued for a maximum of 400K iterations. Policies were validated on 50 new episodes after every 200 iterations.

IL achieved successful team strategies in two out of ten runs. In the remaining runs, IL scored zero goals at every validation step. Figure 1(a) shows that initially IL gets stuck in a bad local minimum, but it eventually manages to converge to a satisfactory policy after 200K iterations on average. Moreover, in policies found by IL, the attacker that receives the ball displays a reactive behavior, i.e. it does not know where to go until its teammate makes a pass; as a result, it spends extra time readjusting and searching for the ball.

**Table 1: DDPG hyper-parameter setting.**

| Parameter | Approach | Value |
|---|---|---|
| Hidden units/layer critic/actor | both | 64,48,32 (RELU) |
| Output units in critic | both | 1 (RELU) |
| Output units in actor | IL | 5 (logistic) |
| Output units in actor | JAL | 10 (logistic) |
| Size of replay buffer | both | 100000 |
| Training batch size | both | 4000 |
| $\epsilon$-greedy control parameter | both | 0.5 |
| Discount rate | both | 0.9 |
| Learning rate critic/actor | both | 0.001 |
| Update rate target networks | both | 0.01 |

JAL accomplished a perfect goal percentage in validation in six out ten runs. Another two ended with percentages of 0.52 and 0.78, and the last two stayed at zero the entire time. Figure 1(b) reveals that JAL can discover effective policies quickly, but it does not truly converge until several steps later (after 300K iterations on average). Policies obtained by JAL demonstrate a beforehand understanding between agents when executing a pass, such that the receiver moves straight to intersect the ball without wasting much time.

In an extra experiment, we remove the defender keeping everything else unchanged. No promising solutions were found within 10 runs of 400K iterations each. Just as in [4], IL and JAL normally got stuck in a bad local minimum, where a single agent scores sporadically (about 50% of times) without cooperating with its partner.

## 4 CONCLUSIONS

This work successfully implemented IL and JAL in the offensive free-kick task and within a 3D simulator. The addition of the defender was proven to be a crucial factor leading to this achievement.

JAL was clearly superior to IL. They both converged after a comparable number of iterations due to full observability; however, JAL discovered good strategies more consistently than IL. In addition, policies found by JAL reveal a higher degree of inter-agent coordination than those found by IL.

Hence, we can conclude that for similar robotic domains JAL constitutes a MARL alternative that should not be ignored.

## REFERENCES


[1] Adekunle A. Adepegba, Suruz Miah, and Davide Spinello. 2016. Multi-Agent Area Coverage Control using Reinforcement Learning. In *The 29th International Florida Artificial Intelligence Research Society (FLAIRS) Conference: Autonomous Robots and Agents*. Key Largo, Florida, USA.

[2] Ronny Conde, José Ramón Llata, and Carlos Torre-Ferrero. 2017. Time-Varying Formation Controllers for Unmanned Aerial Vehicles Using Deep Reinforcement Learning. *CoRR* abs/1706.01384 (2017). arXiv:1706.01384 http://arxiv.org/abs/1706.01384

[3] Ravi N. Haksar and Mac Schwager. 2018. Distributed Deep Reinforcement Learning for Fighting Forest Fires with a Network of Aerial Robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2018, Madrid, Spain, October 1-5, 2018*. 1067–1074. https://doi.org/10.1109/IROS.2018.8593539

[4] Matthew John Hausknecht. 2016. *Cooperation and Communication in Multiagent Deep Reinforcement Learning*. Ph.D. Dissertation. University of Texas at Austin, USA.

[5] R. A. Howard. 1960. *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA.

[6] S. Hung and S. N. Givigi. 2017. A Q-Learning Approach to Flocking With UAVs in a Stochastic Environment. *IEEE Transactions on Cybernetics* 47, 1 (Jan 2017), 186–197. https://doi.org/10.1109/TCYB.2015.2509646

[7] M. Knopp, C. AykÄśn, J. Feldmaier, and H. Shen. 2017. Formation control using GQ(Îż) reinforcement learning. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*. 1043–1048. https://doi.org/10.1109/ROMAN.2017.8172432

[8] Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2015. Continuous control with deep reinforcement learning. *CoRR* abs/1509.02971 (2015).

[9] Yugang Liu and Goldie Nejat. 2016. Multirobot Cooperative Learning for Semi-autonomous Control in Urban Search and Rescue Applications. *J. Field Robot.* 33, 4 (June 2016), 512–536.

[10] Huy Xuan Pham, Hung Manh La, David Feil-Seifer, and Luan Van Nguyen. 2018. Cooperative and Distributed Reinforcement Learning of Drones for Field Coverage. *CoRR* abs/1803.07250 (2018). arXiv:1803.07250 http://arxiv.org/abs/1803.07250

[11] RoboCup Technical Committee. 2018. *RoboCup Standard Platform League (NAO) Rule Book*.

[12] Thomas Röfer, Tim Laue, Yannick Bülter, Daniel Krause, Jonas Kuball, Andre Mühlenbrock, Bernd Poppinga, Markus Prinzler, Lukas Post, Enno Roehrig, René Schröder, and Felix Thielke. 2017. *B-Human: Team Report and Code Release 2017*. Technical Report. Deutsches Forschungszentrum für Künstliche Intelligenz, Universität Bremen.

[13] L. S. Shapley. 1953. Stochastic Games. *Proceedings of the National Academy of Sciences* 39, 10 (1953), 1095–1100. https://doi.org/10.1073/pnas.39.10.1095 arXiv:https://www.pnas.org/content/39/10/1095.full.pdf

[14] L. Zhou, P. Yang, C. Chen, and Y. Gao. 2017. Multiagent Reinforcement Learning With Sparse Interactions by Negotiation and Knowledge Transfer. *IEEE Transactions on Cybernetics* 47, 5 (May 2017), 1238–1250. https://doi.org/10.1109/TCYB.2016.2543238