Multi-agent deep reinforcement learning in mobile robotics

Maxime Toquebiau[1,2], Nicolas Bredeche[2], Faïz Benamar[2], and Jae-Yun Jun[1]

[1]ECE Paris, INSEEC U. Research Center, 37 quai de Grenelle, 75015, Paris, France
[2]Sorbonne Université, CNRS, Institut des Systèmes Intelligents et de Robotique, ISIR, F-75005 Paris, France

July 2021

Introduction

Recently, deep reinforcement learning (DRL) has started to be successfully applied to multi-agent systems (MAS), showing the capacity to learn coordinated behaviours in teams of cooperative agents (Lowe et al., 2017; Rashid et al., 2018; OpenAI et al., 2019). Ultimately, the goal is to use these algorithms in multi-robot systems (MRS) such as self-driving cars or swarms of drones. However, applying DRL to real-world robotics is still an on-going issue (Ibarz et al., 2021) and multi-agent deep reinforcement learning (MADRL) research focuses on toy environments such as sequential social dilemmas or video games (Rashid et al., 2018; Foerster et al., 2018; Mahajan et al., 2019; Son et al., 2019; OpenAI et al., 2019; Rashid et al., 2020; Schroeder de Witt et al., 2020a). While these environments capture some aspects of multi-agent interactions, they lack some attributes of robotic tasks. If we want to apply MADRL in mobile robotics, we need first to ask ourselves what does having agents living in the real world imply. Answering this question will help us design simulated tasks that are more relevant to robotics than the ones studied in MADRL research today.

In future tasks, robots will have to comply to pre-defined rules, some of which may be rather abstract (e.g. social rules). They will need to interact with numerous heterogeneous agents: other kinds of robots, human-driven machines and humans. They will have to adapt to changes in their environment, with new agents coming into play or modifications of their working environment. Finally, they will have to communicate with each other and with humans, using interpretable communication protocols (e.g. natural language). They will have to be able not only to communicate what they see and what they are doing, but also what they need. In the same way, they will need to understand the needs of their peers and humans they interact with.

1 Environment

All these issues are part of real-world tasks that need to be addressed when applying MADRL to robotics. Therefore we need to design simulated tasks that encompass all of these issues, and algorithms that are able to learn to complete them efficiently. In figure 1, we present a task that requires cooperation and communication. Agents have to move boxes that randomly spawn in four different zones to a delivery zone in the center. The boxes can have different sizes that correspond to different weights. Heavier boxes will require multiple agents to be pushed efficiently. The goal is to deliver as many boxes as possible in a given number of steps. The environment is partially observable as agents have a limited range of observation. Moreover, we want the execution to be completely decentralised. Agents will thus have to explore their environment and communicate their observations to get help for moving boxes. Finally, we could easily modify the layout of this environment in order to ensure that our agents are able to generalise well to new situations. This environment is challenging as it requires to learn a communication protocol and to find a good coordination policy for delivering a maximum number of boxes.

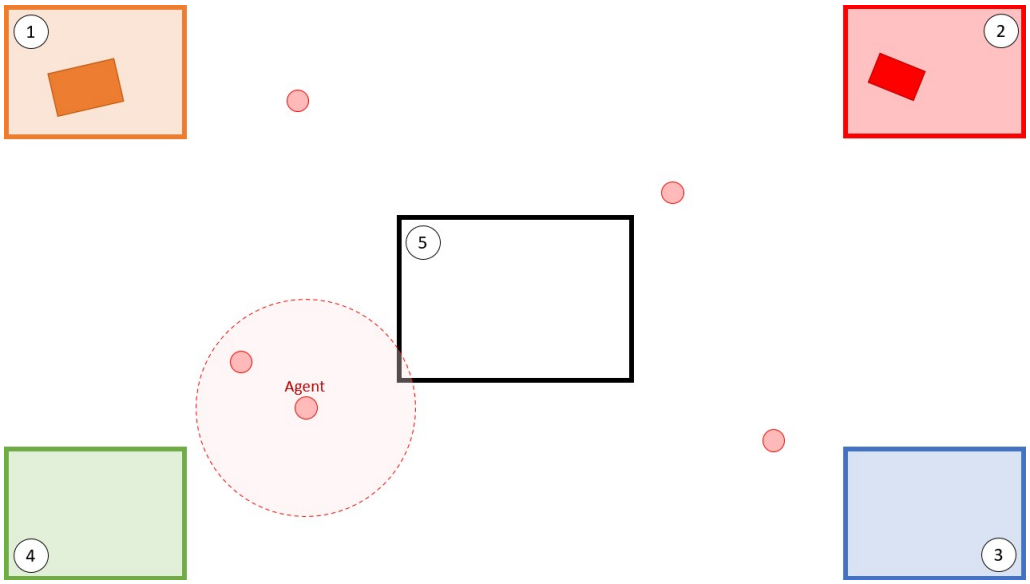

Figure 1: The simulated environment with four zones (1 to 4) where boxes arrive and one delivery zone (5). Agents have a limited field of view, as shown by the red circle area around one agent.

2 Emergent communication in MADRL

In this environment with *partial observability* and *non-stationarity* due to multiple agents concurrently learning, communication is a crucial tool for agents to be efficient. When doing decentralised execution, communication is the only way to entirely leverage the information gathered by all the agents in our system. Despite this clear potential, communication is still not used enough in the MADRL literature. At execution time, agents usually rely only on their observation history to choose their action and try to coordinate with other agents (Lowe et al., 2017; Rashid et al., 2018; Foerster et al., 2018; Schroeder de Witt et al., 2020a). This is unfortunate as communication, when done correctly, can be beneficial to decentralised MAS (Tan, 1993). Thus, this first motivates a study of communication systems in MAS and their impact on MADRL approaches. State-of-the-art MADRL algorithms (Rashid et al., 2018; Foerster et al., 2018; Schroeder de Witt et al., 2020a,b; Wang et al., 2021) can be augmented with an *emergent communication system* (Foerster et al., 2016; Sukhbaatar et al., 2016; Mordatch and Abbeel, 2018; Jiang and Lu, 2018; Das et al., 2019) to see how communication can improve the performance in this task. In emergent communication, agents learn to use a new language to convey information about their observations and their strategy. In our environment, such communication systems can be very helpful for sharing information about boxes that need multiple agents to be pushed, for example. Our contribution will be to study how learning an emergent communication system impacts performance of MADRL algorithms. Specifically, how emergent communication can help alleviate common issues of these algorithms such as relative over-generalisation (Wei and Luke, 2016), and how it can help independent learners to equal or surpass centralised learners (Schroeder de Witt et al., 2020a).

3 Learning an existing language

Next, we will move on to study the learning of a discrete, pre-existing communication system. It is still very hard to interpret emergent languages learnt by teams of agents, or even to measure their impact on the agents' policy (Kottur et al., 2017; Lowe et al., 2019; Lazaridou and Baroni, 2020). But, as we said, robots will need to communicate with human beings with an interpretable language. This first means that the language that we use must be discrete, just like natural language is composed of discrete tokens (i.e. words and symbols). Second, this means that the language already exists beforehand, unlike in Section 2 where the language emerges during training. The tokens all have pre-defined meanings and must be arranged into sentences following strict grammatical rules. This adds a layer of complexity, but it is required if we want

to develop systems that we can interact fully with. Also, ideas from developmental psychology, pioneered by soviet psychologist Vygotsky (1934), state that language is not just a tool for communication, but also plays a part in the cognitive development of humans. Words and phrases, arranged in sentences, are ways for us to understand the world and construct our thoughts. This has led to a whole body of research on language-augmented reinforcement learning (LA-RL) that uses language to guide agents during their task by defining goals (Andreas et al., 2017; Das et al., 2018; Colas et al., 2020; Akakzia et al., 2021). This shows that learning a pre-existing, discrete language can not only help us communicate in a interpretable way, but also drive the learning of multiple agents.

Learning an existing language is hard, even for humans. Traditional natural language methods usually rely on extensive statistical analysis of big corpora of text. This approach lacks of the interactiveness that we would like in a intelligent agent learning to perform a task. In the reinforcement learning (RL) literature, agents have been learning natural language in simple communication games (Lazaridou et al., 2016; Das et al., 2017) and in embodied tasks in previously cited LA-RL approaches. But an existing language has never been learnt in a complex multi-agent environment like ours. The challenge will be to induce the agents to learn to use the language properly, allowing them to improve their performance. This will most likely require a combination of interaction and supervision (Lowe et al., 2020). Agents will interact with one another and discover good ways to use the language. We could enforce this through reward shaping, for example by penalising bad grammar and rewarding good use of words. Supervision can be done in multiple ways. We could draw inspiration from LA-RL approaches by providing descriptions to agents about their observations and actions, from which they will extract some meaning about the words (Andreas et al., 2018; Nguyen et al., 2021; Karch et al., 2021). Or we could look into imitation learning and make the agents learn from demonstrations of pre-programmed agents that use the language in the way we intended (Hester et al., 2018).

For this approach to be achievable, we need to define a simple and efficient language. Teaching natural language to robots would be a great breakthrough, but in our context this is way out of reach. Indeed, human languages are made to convey information in millions of different situations. In comparison, our task is extremely simple. Therefore, the language we use must be simple as well (Mordatch and Abbeel, 2018; Lazaridou and Baroni, 2020). Table 1 presents a set of tokens that could compose our language. To read and generate such tokens, we could use techniques from natural language processing such as language models. This is a restricted list, but we can add any token we want according to what we want to incorporate in our language. For example, we could add tokens to express orders that, if available only to some agents, would induce social rules in our system. If this approach is successful, it would be a first step towards learning an interpretable language in a robotic task.

Type	Token
Entities	AGENT OPERATOR PACKAGE
Locations	RED_AREA ORANGE_AREA GREEN_AREA BLUE_AREA DELIVERY_AREA
Verbs	GOING_TO NEED_HELP PUSH_PACKAGE

Table 1: Possible tokens used in our language.

4 Multi-agent credit assignment

The environment presented in Section 1 is a highly cooperative one. Agents will have to learn to maximise a global team reward. However, as we have said in previous sections, this environment is both partially observable and non-stationary. These two properties make the global reward unsuitable for agents to learn their optimal policy, as they are not able to deduce the quality of their actions from the reward obtained at a team level (Chang et al., 2004). This essentially means that we are faced with a *multi-agent credit assignment* problem where we need to find a way to know how much each agent's actions have participated in the team reward. Multiple techniques have tackled the multi-agent credit assignment problem. For low-dimensional environments, near-optimal approaches have been developed, like the *shapley value* (Shapley, 1953) or the *wonderful life* and *aristocrat* utilities (Wolpert and Tumer, 2002). More recently, these methods have been adapted to larger environments with the use of DRL techniques (Foerster et al., 2018; Nguyen et al., 2018; Zhou et al., 2020). We need to study these solutions in our context and try to find new ones that yield better results. Specifically, we will have to link

credit assignment to communication. Communication actions must impact the behaviour of other agents and improve the final global reward. Thus, our credit assignment strategy must be designed in relation to the communication system, which has never been studied before.

5 Macro-actions as a sim-to-real approach

Perhaps the most important issue when dealing with DRL and robotics is how to use algorithms trained in simulation in the real world. DRL methods are notoriously hard to apply in the real world (Sünderhauf et al., 2018; Ibarz et al., 2021). But we need to find ways to effectively do it, otherwise we will never be able to leverage the power of DRL in robotics. The main issue here is that DRL is highly sample inefficient: it needs to train on great numbers of steps to converge to a good policy. This makes it impractical to train in reality, as experiences in the real world take a long time to execute, way more than in simulation. In addition to that, RL involves doing numerous series of bad actions in order to learn not to execute them in the future, leading to potentially very bad outcomes for the agents. This is obviously undesirable with real robots, as they are usually quite brittle and very expensive.

One solution to these issues is to use the data gathered in simulation to learn behaviours and try to transfer them on real robots. This is challenging due to the multiple differences between simulated environments and reality, all summarised as the *'reality gap'*. This has led to numerous simulation-to-reality (sim-to-real) approaches that use different tricks to bridge this gap (Tobin et al., 2017; Peng et al., 2018; James et al., 2019; Chebotar et al., 2019; Andrychowicz et al., 2020).

Here, we describe how macro-actions could be used as a sim-to-real approach. Macro-actions are temporally-extended actions that execute multiple low-level actions. Based on the options framework of Sutton et al. (1999), they were used by Amato et al. (2019) and Xiao et al. (2020) to learn efficient robotic policies. They allow us to leverage pre-programmed controllers for solving relatively easy sub-problems like navigating to a location. Navigating from one point to another in a known environment is a task that we know how to perform with a robot. Using RL to learn such tasks with robots can be extremely complicated and time consuming. It is thus more pertinent to focus the learning on the more complex part of the policy we want to learn: when to go where, how to coordinate and distribute sub-tasks. To this end, we can use macro-actions such as *"go to zone X"*, executing a pre-programmed policy to perform this macro-action. The agents would then learn a macro-policy that maximises their reward. Now, considering that we have pre-designed controllers for executing all of our macro-actions in the real-world, we could transfer the macro-policy learnt in a simulated environment to a real environment with the same layout. The macro-policy being more high-level, it is less affected by the difference in executing its actions between simulation and reality.

For this approach to be possible, we will have to face multiple challenges. First, we need to address the differences in the inputs of our agents between simulation and reality. If we want to base our robots' observations only on their local sensors, then we will have to transfer the model from the simulated inputs to the sensor inputs. Second, even if macro-actions help us shortening the reality gap, differences in executing actions between simulation and reality may still impact the quality of the macro-policy. For example, if we have pre-designed controllers to go from point A to point B, both in simulation and reality, it does not mean that both controllers perform this task in the same fashion (i.e. path, speed). This might require some work on the simulation side to ensure that these differences don't impact the performance. In this regard, we can draw inspiration from domain and dynamics randomisation techniques in other sim-to-real approaches (Tobin et al., 2017; Peng et al., 2018; Andrychowicz et al., 2020). If we manage to overcome these difficulties, this has the potential to be a first sim-to-real approach for mobile robotics.

References

A. Akakzia, C. Colas, P.-Y. Oudeyer, M. Chetouani, and O. Sigaud. Grounding Language to Autonomously-Acquired Skills via Goal Generation. In *ICLR 2021 - Ninth International Conference on Learning Representation*, 2021. URL https://hal.inria.fr/hal-03121146.

C. Amato, G. Konidaris, L. P. Kaelbling, and J. P. How. Modeling and planning with macro-actions in decentralized pomdps. *Journal of Artificial Intelligence Research*, 2019.

J. Andreas, D. Klein, and S. Levine. Modular multitask reinforcement learning with policy sketches. In *Proceedings of the 34th International Conference on Machine Learning*, 2017. URL http://proceedings.mlr.press/v70/andreas17a.html.

J. Andreas, D. Klein, and S. Levine. Learning with latent language. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. doi: 10.18653/v1/n18-1197.

M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, and W. Zaremba. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 2020. URL https://doi.org/10.1177/0278364919887447.

Y.-h. Chang, T. Ho, and L. Kaelbling. All learning is local: Multi-agent learning in global reward games. In *Advances in Neural Information Processing Systems*, 2004. URL https://proceedings.neurips.cc/paper/2003/file/c8067ad1937f728f51288b3eb986afaa-Paper.pdf.

Y. Chebotar, A. Handa, V. Makoviychuk, M. Macklin, J. Issac, N. Ratliff, and D. Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. In *2019 International Conference on Robotics and Automation (ICRA)*, 2019. doi: 10.1109/icra.2019.8793789.

C. Colas, T. Karch, N. Lair, J.-M. Dussoux, C. Moulin-Frier, P. Dominey, and P.-Y. Oudeyer. Language as a cognitive tool to imagine goals in curiosity driven exploration. In *Advances in Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/file/274e6fcf4a583de4a81c6376f17673e7-Paper.pdf.

A. Das, S. Kottur, J. M. F. Moura, S. Lee, and D. Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Neural modular control for embodied question answering. In *Proceedings of The 2nd Conference on Robot Learning*, 2018. URL http://proceedings.mlr.press/v87/das18a.html.

A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau. TarMAC: Targeted multi-agent communication. In *Proceedings of the 36th International Conference on Machine Learning*, 2019. URL http://proceedings.mlr.press/v97/das19a.html.

J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11794.

J. N. Foerster, Y. M. Assael, N. de Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2016.

T. Hester, M. Vecerik, O. Pietquin, M. Lanctot, T. Schaul, B. Piot, D. Horgan, J. Quan, A. Sendonaris, I. Osband, G. Dulac-Arnold, J. Agapiou, J. Leibo, and A. Gruslys. Deep q-learning from demonstrations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018. URL https://ojs.aaai.org/index.php/AAAI/article/view/11757.

J. Ibarz, J. Tan, C. Finn, M. Kalakrishnan, P. Pastor, and S. Levine. How to train your robot with deep reinforcement learning: lessons we have learned. *The International Journal of Robotics Research*, 2021. doi: 10.1177/0278364920987859. URL https://doi.org/10.1177/0278364920987859.

S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

J. Jiang and Z. Lu. Learning attentional communication for multi-agent cooperation. In *Advances in Neural Information Processing Systems*, 2018. URL https://proceedings.neurips.cc/paper/2018/file/6a8018b3a00b69c008601b8becae392b-Paper.pdf.

T. Karch, L. Teodorescu, K. Hofmann, C. Moulin-Frier, and P.-Y. Oudeyer. Grounding spatio-temporal language with transformers. 2021.

S. Kottur, J. Moura, S. Lee, and D. Batra. Natural language does not emerge 'naturally' in multi-agent dialog. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

A. Lazaridou and M. Baroni. Emergent multi-agent communication in the deep learning era. 2020.

A. Lazaridou, A. Peysakhovich, and M. Baroni. Multi-agent cooperation and the emergence of (natural) language. 2016.

R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017. URL https://dl.acm.org/doi/abs/10.5555/3295222.3295385.

R. Lowe, J. Foerster, Y.-L. Boureau, J. Pineau, and Y. Dauphin. On the pitfalls of measuring emergent communication. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, 2019.

R. Lowe, A. Gupta, J. Foerster, D. Kiela, and J. Pineau. On the interaction between supervision and self-play in emergent communication. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rJxGLlBtwH.

A. Mahajan, T. Rashid, M. Samvelyan, and S. Whiteson. Maven: Multi-agent variational exploration. *Advances in Neural Information Processing Systems, 32*, 2019.

I. Mordatch and P. Abbeel. Emergence of grounded compositional language in multi-agent populations. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17007.

D. T. Nguyen, A. Kumar, and H. C. Lau. Credit assignment for collective multi-agent rl with global rewards. In *Advances in Neural Information Processing Systems 31 (NeurIPS 2018)*, pages 8113–8124, 2018. URL http://papers.nips.cc/paper/8033-credit-assignment-for-collective-multiagent-rl-with-global-rewards.

K. Nguyen, D. Misra, R. Schapire, M. Dudík, and P. Shafto. Interactive learning from activity description. 2021.

OpenAI, C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, R. Józefowicz, S. Gray, C. Olsson, J. Pachocki, M. Petrov, H. P. d. O. Pinto, J. Raiman, T. Salimans, J. Schlatter, J. Schneider, S. Sidor, I. Sutskever, J. Tang, F. Wolski, and S. Zhang. Dota 2 with large scale deep reinforcement learning. 2019.

X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. doi: 10.1109/icra.2018.8460528.

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*, 2018. URL http://proceedings.mlr.press/v80/rashid18a.html.

T. Rashid, G. Farquhar, B. Peng, and S. Whiteson. Weighted qmix: Expanding monotonic value function factorisation for deep multiagent reinforcement learning. 2020.

C. Schroeder de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. S. Torr, M. Sun, and S. Whiteson. Is independent learning all you need in the starcraft multi-agent challenge? 2020a.

C. Schroeder de Witt, B. Peng, P.-A. Kamienny, P. Torr, W. Böhmer, and S. Whiteson. Deep multi-agent reinforcement learning for decentralized continuous cooperative control. 2020b.

L. S. Shapley. A value for n-person games. *Contributions to the Theory of Games*, 1953. URL https://ci.nii.ac.jp/naid/10013542751/en/.

K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International Conference on Machine Learning*, 2019.

S. Sukhbaatar, A. Szlam, and R. Fergus. Learning multiagent communication with backpropagation. *Advances in Neural Information Processing Systems*, 2016.

R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 1999. doi: https://doi.org/10.1016/S0004-3702(99)00052-1. URL https://www.sciencedirect.com/science/article/pii/S0004370299000521.

N. Sünderhauf, O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, and P. Corke. The limits and potentials of deep learning for robotics. *The International Journal of Robotics Research*, 2018. URL https://doi.org/10.1177/0278364918770733.

M. Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. In *Proceedings of the Tenth International Conference on Machine Learning*, 1993.

J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017. doi: 10.1109/IROS.2017.8202133.

L. S. Vygotsky. *Thought and Language*. 1934.

J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang. Qplex: Duplex dueling multi-agent q-learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Rcmk0xxIQV.

E. Wei and S. Luke. Lenient learning in independent-learner stochastic cooperative games. *The Journal of Machine Learning Research*, 2016.

D. H. Wolpert and K. Tumer. *Optimal Payoff Functions for Members of Collectives*, pages 355–369. 2002. doi: 10.1142/9789812777263_0020. URL https://www.worldscientific.com/doi/abs/10.1142/9789812777263_0020.

Y. Xiao, J. Hoffman, T. Xia, and C. Amato. Learning multi-robot decentralized macro-action-based policies via a centralized q-net. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020. doi: 10.1109/ICRA40945.2020.9196684.

M. Zhou, Z. Liu, P. Sui, Y. Li, and Y. Y. Chung. Learning implicit credit assignment for cooperative multi-agent reinforcement learning. 2020.