

The Holy Grail of Multi-Robot Planning: Learning to Generate Online-Scalable Solutions from Offline-Optimal Experts

Blue Sky Ideas Track

Amanda Prorok
University of Cambridge
Cambridge, UK
asp45@cam.ac.uk

Jan Blumenkamp
University of Cambridge
Cambridge, UK
jb2270@cam.ac.uk

Qingbiao Li
University of Cambridge
Cambridge, UK
ql295@cam.ac.uk

Ryan Kortvelesy
University of Cambridge
Cambridge, UK
rk627@cam.ac.uk

Zhe Liu
University of Cambridge
Cambridge, UK
zl457@cam.ac.uk

Ethan Stump
DEVCOM Army Research Laboratory
Maryland, USA
ethan.a.stump2.civ@mail.mil

ABSTRACT

Many multi-robot planning problems are burdened by the curse of dimensionality, which compounds the difficulty of applying solutions to large-scale problem instances. The use of learning-based methods in multi-robot planning holds great promise as it enables us to offload the *online* computational burden of expensive centralized, yet optimal solvers, to an *offline* learning procedure. The hope is that by training a policy to copy an optimal pattern generated by a small-scale (centralized) system, we can transfer that policy to much larger, decentralized systems while maintaining near-optimal performance. Yet, a number of issues impede us from leveraging this idea to its full potential. This blue-sky paper elaborates some of the key challenges that remain.

KEYWORDS

Multi-Robot Planning, Imitation Learning, Graph Neural Networks

ACM Reference Format:

Amanda Prorok, Jan Blumenkamp, Qingbiao Li, Ryan Kortvelesy, Zhe Liu, and Ethan Stump. 2022. The Holy Grail of Multi-Robot Planning: Learning to Generate Online-Scalable Solutions from Offline-Optimal Experts: Blue Sky Ideas Track. In *Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022)*, Auckland, New Zealand, May 9–13, 2022, IFAAMAS, 5 pages.

1 INTRODUCTION

Learning-based methods have proven effective at designing robot control policies for an increasing number of tasks [42, 49]. The application of learning-based methods to multi-robot planning has attracted particular attention due to their capability of handling high-dimensional joint state-space representations, by offloading the online computational burden to an offline learning procedure [25, 50]. We argue that these developments point to a fundamental approach that combines ideas around the application of *learning* to optimization, to produce a flexible framework that

could tackle many hard but important problems in robotics, including multi-agent path planning [57], area coverage [46], task allocation [39, 40], formation control [32], and target-tracking [18]. In this blue-sky paper, we motivate this approach and discuss the crucial challenges and research questions.

The ideas here sit within a larger landscape of the application of machine learning to the solution of optimization problems. Consider Figure 1, where we illustrate how learning is applied to either increase the scale of solvable problems or to increase the ability to deal with practical, partial-information problems. Along the problem scale axis, for example, the operations research community has made use of learned heuristics to solve the Traveling Salesperson Problem (TSP) [11, 15], the Vehicle Routing Problem (VRP) [36], and general Mixed Integer Linear Programming Problems (MILPs) [21]. Along the information axis, which includes dealing with Partially Observable Markov Decision Processes (POMDPs), techniques such as Reinforcement Learning (RL) play a major role, as well as ideas such as tuning Monte-Carlo Tree Search [19], embedding learned components into optimal control frameworks [44], and learning how to bias sampling planners [27].

Practical multi-robot planning and control builds on the progress along both of these axes: the degrees of freedom and environment complexity increase (i.e., *larger problem*), while the ability to communicate and coordinate at scale decreases (i.e., *less information*). Traditional *centralized* approaches would use a planning unit to produce coordinated plans that agents use for real-time on-board control; these have the advantage of producing optimal and complete plans in the joint configuration space, but unfortunately, true optimality is NP-hard in many cases [57] and they will struggle when communications are degraded and frequent replanning is required. By contrast, *decentralized* approaches reduce the computational overhead [10] and relax the dependence on centralized units [51, 55] to deal with challenged communications. However, these approaches account for purely local objectives and cannot explicitly optimize global objectives (e.g., path efficiency).

The topic discussed here encompasses techniques for learning to coordinate multi-robot systems in real-world applications. It emerges from the trade-off between information availability (from

Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mascardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Auckland, New Zealand. © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaaamas.org). All rights reserved.

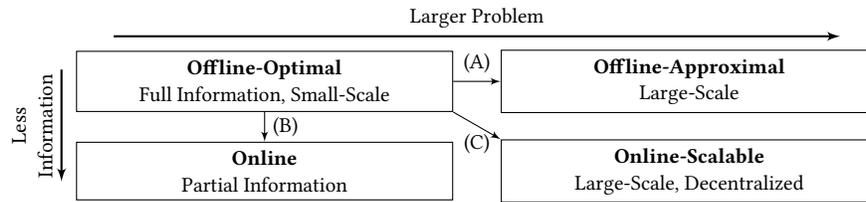


Figure 1: Applications of learning to optimization problems. (A) embodies techniques for learning optimization heuristics; (B) embodies techniques for learning to solve POMDPs; (C) is the emerging topic discussed here, embodying techniques for learning to coordinate large systems in real-world applications

full to partial) and problem size (small to large), as illustrated in Figure 1. What the directions in the figure also indicate is that *success follows from starting with simple problems and using their examples to approach complex ones*. This progression from example to application is reminiscent of *Imitation Learning*. Although we use this crucial observation to understand how learning can play a role in mitigating the shortcomings of decentralized approaches in solving challenging multi-robot problems, it also implies that we need to discover solutions that generalize to larger problems, with less information available.

2 FROM SINGLE-ROBOT LEARNING TO MULTI-ROBOT LEARNING

Learning-based methods promise to find solutions that *balance optimality and real-world efficiency*, thus, bridging the gap between the centralized and decentralized approaches. The process of generating data-driven solutions for multi-robot systems, however, cannot directly borrow from single-robot learning methods because (a) hidden (unobservable) information about other robots must be incorporated through learned communication strategies, and (b), although policies are executed locally, the ensuing actions should lead to plans with a performance near to that of coupled solutions. This agenda means that we need to address the following key points:

- (1) how to generate multi-robot training data,
- (2) how to synthesize decentralizable policies, and
- (3) how to transfer these policies to real-world systems.

The following sections discuss these three key challenges and indicate promising directions.

3 LEARNING DECENTRALIZED POLICIES BY COPYING CENTRALIZED EXPERTS

Although the use of learning-based solutions circumvents some of the problems associated with multi-robot planning, it also introduces new challenges. While it eliminates the issue of an exponentially growing planning space as the time horizon increases, it also introduces a new consideration: coverage of the state-action space. Even though inference can be performed in constant time, the predicted action is unlikely to be accurate unless the state-action space has been sufficiently explored during the training process. Since the size of the joint state-action space grows exponentially in the number of robots in the system, we are still plagued by the curse of dimensionality. This core challenge is the reason why the

development of learning-based multi-robot controllers is a nascent field. While a number of learning paradigms have been applied to this topic (e.g., RL [5, 54]), this position paper focuses on imitation learning strategies, in the first instance.

The following paragraphs discuss three key components of the learning process: data generation, communication strategies, and sim-to-real transfer.

3.1 Experts and Data Generation

How to generate expert data? The works in [25, 50] show that it is possible to train decentralized controllers to learn communication and action policies that optimize a global objective by imitating a centralized optimal expert, in other words, through *behavior cloning*. Li et al. [25] consider the specific case-study of multi-agent path planning, and use Conflict-Based Search (CBS) [47] to find optimal solutions (i.e., sets of optimal, collision-free paths) that compose the training data set. Although the reported results demonstrate unprecedented performance in decentralized systems (i.e., achieving higher than 96% success rates with single-digit flowtime increases, compared to the centralized expert solution), the solution does not generalize well to instances of greater scale than those seen in training. This points to the known fact that simply training the models through behavior cloning leads to bias and over-fitting, since the performance of the network is intrinsically constrained by its training data set. Alternative approaches include learning curricula [3] to optimize the usage of the existing training set, or the introduction of data augmentation mechanisms, which allow experts to teach the learner how to recover from past mistakes.

How to augment existing datasets? One of the major limitations of behavior cloning is that it does not learn to recover from failures, and is unable to handle unseen situations [1]. For example, if the policy has deviated from the optimal trajectory at one-time step, it will fail in getting back to states seen by the expert, hence, resulting in a cascade of errors. One solution (i.e., DAgger [45]) is to introduce the expert *during* training to teach the learner how to recover from mistakes. In [25], the authors demonstrate the utility of this approach by making use of a novel dataset aggregation method that leverages an online expert to resolve hard cases during training. Other approaches are to directly extract a policy from training data, such as GAIL [14]. More broadly speaking, with data augmentation, one can produce arbitrary amounts of training data from arbitrary probability distributions to account for a variety of factors, such as roadmap structure, local environment, obstacle density, motion characteristics, and local robot configurations.

Such carefully controlled distributions enable us to introduce different levels of local coordination difficulties and generate the most challenging instances at each training stage, inherently achieving a form of curriculum learning. More work on data augmentation methods would allow us to better understand the ability boundary of the trained model, to analyze the correlation between different factors, and to identify factors that have the strongest effect on system performance.

3.2 Communication for Decentralized Control

What, how and when to send information? While effective communication is key to decentralized control, it is far from obvious *what information is crucial to the task, and what must be shared among agents*. This question differs from problem to problem and the optimal strategy is often unknown. Hand-engineered coordination strategies often fail to deliver the desired performance, and despite ongoing progress in this domain, they still require substantial design effort. Recent work has shown the promise of Graph Neural Networks (GNNs) to learn explicit communication strategies that enable complex multi-agent coordination [22, 23, 25, 50]. In the context of multi-robot systems, individual robots are modeled as nodes, the communication links between them as edges, and the internal state of each robot as graph signals. By sending messages over the communication links, each robot in the graph indirectly receives access to the global state. One key attribute of GNNs is that they compress data as it flows through the communication graph. In effect, this compresses the global state, affording agents access to global data without inundating them with the entire raw global state. Since compression is performed on local networks (with parameters that can be shared across the entire graph), GNNs are able to compress previously unseen global states. In the process of learning how to compress the global state, GNNs also learn which elements of the signal are the most important, and discard the irrelevant information [23]. This produces a non-injective mapping from global states to latent states, where similar global states ‘overlap’, further improving generalization. Although GNNs clearly promise to deliver generalizable policies as systems scale, more work needs to be done in order to guarantee said property.

Are all messages equally important? If communication happens concurrently and equivalently among many neighboring robots, it is likely to cause redundant information, burden robots’ computational capacity and adversely affect overall team performance. Hence, new approaches towards *communication-aware planning* are required. A potential approach is to introduce *attention mechanisms* to actively measure the relative importance of messages (and their senders). Attention mechanisms have been actively studied and widely adopted in various learning-based models [52], which can be viewed as dynamically amplifying or reducing the weights of features based on their relative importance computed by a given mechanism. Hence, the network can be trained to focus on task-relevant parts of the graph [53]. Learning attention over static graphs has shown to be efficient. Liu et al. [28] developed a learning-based communication model that constructs a communication group on a static graph to address what to transmit and which agent to communicate to for collaborative perception. More recently, Li et al. [26] integrated an attention mechanism with a GNN-based communication strategy to allow for *message-dependent attention* in a

multi-agent path planning problem. A key-query-like mechanism determines the relative importance of features in the messages received from various neighboring robots. They demonstrate that it is possible to achieve performance close to that of a coupled centralized expert algorithm, while scaling to problem instances that are 100× larger than the training instances. An interesting avenue of research that has yet to be explored is that of attention to messages that originate multiple hops away (while ignoring irrelevant messages from ‘nearer’ agents).

3.3 Multi-Robot Sim-to-Real

Expert data is typically generated in a simulation, yet policies trained in simulation often do not generalize to the real world. This is referred to as the *reality gap* [16].

Why is sim-to-real transfer difficult? Even though simulations have become more realistic and easily accessible over recent years [8, 12], it is often computationally infeasible to replicate all aspects of real-world physics in a simulation since the uncertainty and randomness of complex robot-world interactions are difficult to model. Domain randomization is an intuitive solution to this problem, but also makes the task to learn harder than necessary and therefore results in sub-optimal policies. While the reality gap is a major challenge in computer vision, robotics also deals with physical interaction with the real world and dynamic factors such as inertia, for example in robotic grasping [6, 17], drone flight [20, 29] or robotic locomotion [35, 48].

Why is sim-to-real transfer even more difficult for multi-robot systems? While sim-to-real in the single-robot domain typically deals with robot-world interaction, the multi-robot domain is also concerned with robot-robot interactions. An example of this is a swarm of drones flying closely to each other and turbulence affecting the motions of other drones in the vicinity. Above, we established that communication is key to efficient multi-robot interaction, but it is not yet clear how such communications are affected by the reality gap. Multi-robot coordination is typically trained in a synchronous manner, but when deploying these policies to the real-world, decentralized communication is *asynchronous* [4]. Furthermore, randomness such as message dropouts and delays are typically not considered during synchronous training. There is a dearth of research that evaluates the *robustness of models* to such factors, and the impact that they have on the performance of policies. Decentralization is key to successful multi-agent systems, therefore decentralized mesh communication networks are required to operate multi-robot systems in the real world, which may pose additional challenges to the sim-to-real transfer. Lastly, during cooperative training it is typically assumed that all agents are being truthful about their communications, but faulty and malicious agents can be part of the real world and cause additional problems [5, 33, 41].

How can we close the reality gap? We see a few possible avenues to tackle the sim-to-real transfer for multi-robot communication. Domain randomization facilitates the process of making the real-world a permutation of the training environment, and likely improves performance, potentially even against faulty agents and adversarial attacks [5, 33], yet leading to sub-optimal policies. More realistic (network) simulations [7] are always helpful, but also costly alternatives. Methods such as *sim-to-real via real-to-sim* [58]

or training agents in the real-world in a *mixed reality* setting [34] and federated, decentralized learning where individual robots collect data and use it to update a local model that is then aggregated into a global model can benefit the sim-to-real transfer [31, 56].

4 RESEARCH DIRECTIONS

The sections above lay out the challenges entailed by the described approach. We discussed how more research in the areas of data augmentation, synthesis of effective communication strategies, and multi-robot sim-to-real would help us propel the frontier of solvable problems. Nonetheless, we conclude this position paper with two more questions that give a broader perspective of the problem:

Is imitation learning the right paradigm? There are two main approaches to training a controller for a multi-robot system: imitation learning (IL), e.g., [24], and reinforcement learning (RL), e.g., [30]. The most obvious benefit to RL is that it does not require an expert algorithm, as it simply optimizes a reward function. Moreover, there has recently been significant progress within the field of Multi-Agent RL (MARL), which has provided solutions to the training problem, e.g., through Centralized Training Decentralized Execution (CTDE) paradigms [37], as well as through credit assignment mechanisms [13, 43]. While RL is used to solve complex problems that do not have an existing solution, usually, IL is used to provide fast solutions (i.e., policies) that approximate computationally expensive algorithms. When deciding which scheme to use, it is important to consider the fact that performance in IL can never surpass that of the expert algorithm. On the other hand, there is no theoretical upper bound to the performance of a policy trained with RL—in some cases, this can lead to non-intuitive behavior which exploits the reward function or the environment [2]. Also, the reward function requires careful consideration to guarantee that the learned controller does not exploit it by using unsafe or inappropriate actions. Conversely, IL is often biased around regions that can be reached by the expert and, consequently, if the controller ever finds itself in a previously unseen situation, it might exhibit unpredictable behavior.

Despite the clear benefits of RL, practitioners often opt for IL simply because of the additional challenges that RL presents. RL is significantly less sample-efficient than IL because in many cases, it requires a critic to be trained simultaneously with the actor (as opposed to IL, which only necessitates training an actor). Furthermore, the training process of on-policy algorithms can be confounded by a perceived non-stationary environment, which occurs when a policy receives a different reward for the same action in the same state, due to the policy itself changing throughout the learning process. Such situations are common when multiple agents are learning simultaneously, as in multi-robot learning [38]. Many RL algorithms are also sensitive to hyper-parameters, and require time-consuming parameter tuning. Also, as mentioned above, even though methods to solve the credit assignment problem exist, they rely on assumptions (such as agent independence) that often do not reflect reality. Clearly, reinforcement learning is a promising field, but more needs to be done for it to become practical and practicable. Possible future directions should also explore the combination of both IL and RL (e.g., [54]) in the context of decentralized multi-robot systems.

Is it possible to learn small-scale coordination patterns for large-scale systems? Our premise (and hope) is that controllers

trained on only a few robots can then be deployed on large-scale systems with hundreds and even thousands of robots (an approach that not only facilitates data generation, but also accelerates the training process). We posit that achieving this expectation is within our reach. A recent example can be found in [9], where the local coordination behaviors and conventions learned in a partially observable world successfully scales up to 2048 mobile robots in crowded and highly-structured environments. In [26], a promising demonstration shows that the policy trained in 20×20 maps with only 10 robots obtains a success rate above 80% in 200×200 maps with 1000 robots, and more impressively, the learned policy only spends roughly one tenth of the computation time compared to the centralized expert. Overall, these preliminary results give us confidence that we should continue leveraging methods, such as IL, to distill offline-optimal algorithms to online-scalable controllers.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of ARL grant DCIST CRA W911NF-17-2-0181, Engineering and Physical Sciences Research Council (grant EP/S015493/1), and European Research Council (ERC) Project 949940 (gAIA).

REFERENCES

- [1] Alexandre Attia and Sharone Dayan. 2018. Global overview of imitation learning. *arXiv preprint arXiv:1801.06503* (2018).
- [2] Bowen Baker, Ingmar Kanitscheider, Todor Markov, Yi Wu, Glenn Powell, Bob McGrew, and Igor Mordatch. 2019. Emergent tool use from multi-agent autocurricula. *arXiv preprint arXiv:1909.07528* (2019).
- [3] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*. 41–48.
- [4] Jan Blumenkamp, Steven Morad, Jennifer Gielis, Qingbiao Li, and Amanda Prorok. 2022. A Framework for Real-World Multi-Robot Systems Running Decentralized GNN-Based Policies. *IEEE International Conference Robotics and Automation (ICRA)* (2022).
- [5] Jan Blumenkamp and Amanda Prorok. 2020. The Emergence of Adversarial Communication in Multi-Agent Reinforcement Learning. *Conference on Robot Learning (CoRL)* (2020).
- [6] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, et al. 2018. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 4243–4250.
- [7] Miguel Calvo-Fullana, Daniel Mox, Alexander Pyattaev, Jonathan Fink, Vijay Kumar, and Alejandro Ribeiro. 2021. Ros-netsim: A framework for the integration of robotic and network simulators. *IEEE Robotics and Automation Letters* 6, 2 (2021), 1120–1127.
- [8] Erwin Coumans and Yunfei Bai. 2016–2021. PyBullet, a Python module for physics simulation for games, robotics and machine learning. <http://pybullet.org>.
- [9] Mehul Damani, Zhiyao Luo, Emerson Wenzel, and Guillaume Sartoretti. 2021. PRIMAL₂: Pathfinding Via Reinforcement and Imitation Multi-Agent Learning - Lifelong. *IEEE Robotics and Automation Letters* 6, 2 (2021), 2666–2673.
- [10] Vishnu R Desaraju and Jonathan P How. 2012. Decentralized path planning for multi-agent teams with complex constraints. *Autonomous Robots* 32, 4 (2012), 385–403. Publisher: Springer.
- [11] Michel Deudon, Pierre Cournut, Alexandre Lacoste, Yossiri Adulyasak, and Louis-Martin Rousseau. 2018. Learning heuristics for the TSP by policy gradient. In *International conference on the integration of constraint programming, artificial intelligence, and operations research*. Springer, 170–181.
- [12] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. CARLA: An Open Urban Driving Simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*. 1–16.
- [13] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 32.
- [14] Jonathan Ho and Stefano Ermon. 2016. Generative adversarial imitation learning. *Advances in neural information processing systems* 29 (2016), 4565–4573.
- [15] Benjamin Hudson, Qingbiao Li, Matthew Malencia, and Amanda Prorok. 2022. Graph Neural Network Guided Local Search for the Traveling Salesperson Problem. *International Conference on Learning Representations (ICLR)* (2022).

- [16] Nick Jakobi, Phil Husbands, and Inman Harvey. 1995. Noise and the Reality Gap: The Use of Simulation in Evolutionary Robotics, In *Advances in Artificial Life. Lecture Notes in Artificial Intelligence* 929.
- [17] S. James, P. Wohlhart, M. Kalakrishnan, D. Kalashnikov, A. Irpan, J. Ibarz, S. Levine, R. Hadsell, and K. Bousmalis. 2019. Sim-To-Real via Sim-To-Sim: Data-Efficient Robotic Grasping via Randomized-To-Canonical Adaptation Networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, Los Alamitos, CA, USA, 12619–12629. <https://doi.org/10.1109/CVPR.2019.01291>
- [18] Boyoon Jung and Gaurav S Sukhatme. 2006. Cooperative multi-robot target tracking. In *Distributed autonomous robotic systems 7*. Springer, 81–90.
- [19] Sammie Katt, Frans A Oliehoek, and Christopher Amato. 2017. Learning in POMDPs with Monte Carlo tree search. In *International Conference on Machine Learning*. PMLR, 1819–1827.
- [20] Elia Kaufmann, Antonio Loquercio, René Ranftl, Matthias Müller, Vladlen Koltun, and Davide Scaramuzza. 2020. Deep Drone Acrobatics. In *Proceedings of Robotics: Science and Systems*. Corvallis, Oregon, USA. <https://doi.org/10.15607/RSS.2020.XVI.040>
- [21] Elias B Khalil, Bistra Dilkina, George L Nemhauser, Shabbir Ahmed, and Yufen Shao. 2017. Learning to Run Heuristics in Tree Search.. In *IJCAI* 659–666.
- [22] Arbaaz Khan, Ekaterina Tolstaya, Alejandro Ribeiro, and Vijay Kumar. 2020. Graph policy gradients for large scale robot control. In *Conference on Robot Learning*, 823–834.
- [23] Ryan Kortvelesy and Amanda Prorok. 2021. ModGNN: Expert Policy Approximation in Multi-Agent Systems with a Modular Graph Neural Network Architecture. *International Conference on Robotics and Automation (ICRA)* (2021).
- [24] Hoang M Le, Yisong Yue, Peter Carr, and Patrick Lucey. 2017. Coordinated Multi-Agent Imitation Learning. In *International Conference on Machine Learning*. <http://proceedings.mlr.press/v70/le17a/le17a.pdf>
- [25] Qingbiao Li, Fernando Gama, Alejandro Ribeiro, and Amanda Prorok. 2020. Graph neural networks for decentralized multi-robot path planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11785–11792.
- [26] Qingbiao Li, Weizhe Lin, Zhe Liu, and Amanda Prorok. 2021. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics and Automation Letters* 6, 3 (2021), 5533–5540.
- [27] Katherine Liu, Martina Stadler, and Nicholas Roy. 2020. Learned Sampling Distributions for Efficient Planning in Hybrid Geometric and Object-Level Representations. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9555–9562.
- [28] Yen-Cheng Liu, Junjiao Tian, Nathaniel Glaser, and Zsolt Kira. 2020. When2com: multi-agent perception via communication graph grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*. 4106–4115.
- [29] Antonio Loquercio, Elia Kaufmann, René Ranftl, Alexey Dosovitskiy, Vladlen Koltun, and Davide Scaramuzza. 2019. Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics* 36, 1 (2019), 1–14.
- [30] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In *Proceedings of the 31st International Conference on Neural Information Processing Systems (Long Beach, California, USA) (NIPS’17)*. Curran Associates Inc., Red Hook, NY, USA, 6382–6393.
- [31] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
- [32] N Michael, M M Zavlanos, V. Kumar, and G J Pappas. 2008. Distributed multi-robot task assignment and formation control. In *IEEE International Conference Robotics and Automation*. 128–133.
- [33] Rupert Mitchell, Jan Blumenkamp, and Amanda Prorok. 2020. Gaussian Process Based Message Filtering for Robust Multi-Agent Cooperation in the Presence of Adversarial Communication. *CoRR abs/2012.00508* (2020). [arXiv:2012.00508](https://arxiv.org/abs/2012.00508)
- [34] Rupert Mitchell, Jenny Fletcher, Jacopo Panerati, and Amanda Prorok. 2020. Multi-Vehicle Mixed Reality Reinforcement Learning for Autonomous Multi-Lane Driving. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (Auckland, New Zealand) (AAMAS ’20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1928–1930.
- [35] Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S. Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. 2018. Learning to Adapt in Dynamic, Real-World Environments through Meta-Reinforcement Learning. In *ICLR*.
- [36] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V Snyder, and Martin Takáč. 2018. Reinforcement learning for solving the vehicle routing problem. *arXiv preprint arXiv:1802.04240* (2018).
- [37] Frans A Oliehoek, Christopher Amato, et al. 2016. *A concise introduction to decentralized POMDPs*. Vol. 1. Springer.
- [38] Georgios Papoudakis, Filippos Christianos, Arrasy Rahman, and Stefano V Albrecht. 2019. Dealing with non-stationarity in multi-agent deep reinforcement learning. *arXiv preprint arXiv:1906.04737* (2019).
- [39] S S Ponda, L B Johnson, and J P How. 2012. Distributed chance-constrained task allocation for autonomous multi-agent teams. In *American Control Conference (ACC)*. 4528–4533.
- [40] Amanda Prorok. 2019. Redundant Robot Assignment on Graphs with Uncertain Edge Costs. In *Distributed Autonomous Robotic Systems (Springer Proceedings in Advanced Robotics)*, Nikolaus Correll, Mac Schwager, and Michael Otte (Eds.). 313–327.
- [41] Amanda Prorok, Matthew Malencia, Luca Carlone, Gaurav S Sukhatme, Brian M Sadler, and Vijay Kumar. 2021. Beyond Robustness: A Taxonomy of Approaches towards Resilient Multi-Robot Systems. *arXiv preprint arXiv:2109.12343* (2021).
- [42] Aravind Rajeswaran, Kendall Lowrey, Emanuel V. Todorov, and Sham M Kakade. 2017. Towards Generalization and Simplicity in Continuous Control. In *Advances in Neural Information Processing Systems (NIPS)*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Vol. 30. Curran Associates, Inc., 6550–6561.
- [43] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning. In *International Conference on Machine Learning*. PMLR, 4295–4304.
- [44] Charles Richter, John Ware, and Nicholas Roy. 2014. High-speed autonomous navigation of unknown environments using learned probabilities of collision. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6114–6121.
- [45] Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 627–635.
- [46] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. 2009. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research* 28, 3 (2009), 357–375.
- [47] Guni Sharon, Roni Stern, Ariel Felner, and Nathan R Sturtevant. 2015. Conflict-based search for optimal multi-agent pathfinding. *Artificial Intelligence* 219 (2015), 40–66.
- [48] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. 2018. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. In *Robotics: Science and Systems*. <https://arxiv.org/pdf/1804.10332.pdf>
- [49] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel. 2017. 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)*, 23–30. <https://doi.org/10.1109/IROS.2017.8202133>
- [50] Ekaterina Tolstaya, Fernando Gama, James Paulos, George Pappas, Vijay Kumar, and Alejandro Ribeiro. 2019. Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks. *arXiv:1903.10527 [cs]* (March 2019). <http://arxiv.org/abs/1903.10527> arXiv: 1903.10527.
- [51] Jur Van den Berg, Ming Lin, and Dinesh Manocha. 2008. Reciprocal velocity obstacles for real-time multi-agent navigation. In *IEEE international conference on robotics and automation (ICRA)*. 1928–1935. [text.org/organization](http://www.ros.org/organization): IEEE.
- [52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. 5998–6008.
- [53] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. *International Conference on Learning Representations* (2018).
- [54] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, Junhyuk Oh, Dan Horgan, Manuel Kroiss, Ivo Danihelka, Aja Huang, Laurent Sifre, Trevor Cai, John P Agapiou, Max Jaderberg, Alexander S Vezhnevets, Rémi Leblond, Tobias Pohlen, Valentin Dalibard, David Budden, Yury Sulsky, James Molloy, Tom L Paine, Caglar Gulcehre, Ziyu Wang, Tobias Pfaff, Yuhuai Wu, Roman Ring, Dani Yogatama, Dario Wünsch, Katrina McKinney, Oliver Smith, Tom Schaul, Timothy Lillicrap, Koray Kavukcuoglu, Demis Hassabis, Chris Apps, and David Silver. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. In *Nature*. <https://www.nature.com/articles/s41586-019-1724-z.pdf>
- [55] Binyu Wang, Zhe Liu, Qingbiao Li, and Amanda Prorok. 2020. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robotics and Automation Letters* 5, 4 (2020), 6932–6939.
- [56] Baoqian Wang, Junfei Xie, and Nikolay Atanasov. 2021. Coding for Distributed Multi-Agent Reinforcement Learning. *arXiv preprint arXiv:2101.02308* (2021).
- [57] Jingjin Yu and Steven M LaValle. 2013. Structure and intractability of optimal multi-robot path planning on graphs. In *AAAI*.
- [58] Jingwei Zhang, Lei Tai, Peng Yun, Yufeng Xiong, Ming Liu, Joschka Boedecker, and Wolfram Burgard. 2019. Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *IEEE Robotics and Automation Letters* 4, 2 (2019), 1148–1155.