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

Amanda Prorok* Jan Blumenkamp* Qingbiao Li* Ryan Kortvelesy* Zhe Liu*

Department of Computer Science and Technology
University of Cambridge, UK
{asp45,jb2270,q1295,rk627,z1457}@cam.ac.uk

Ethan Stump*

DEVCOM Army Research Laboratory (ARL), 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, yet optimal solvers, to an *offline* learning procedure. Simply put, the idea is to train a policy to copy an optimal pattern generated by a small-scale system, and then transfer that policy to much larger systems, in the hope that the learned strategy scales, 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

1 Introduction

Learning-based methods have proven effective at designing robot control policies for an increasing number of tasks [1, 2]. 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 [3, 4]. We argue that these developments point to a fundamental approach that combines ideas around the application of learning to optimization and produce a flexible framework that could tackle many hard but important problems in robotics, including multi-agent path planning [5], area coverage [6, 7], task allocation [8, 9, 10], formation control [11], and target-tracking [12]. In this paper, we motivate this approach and discuss the crucial challenges and research questions.

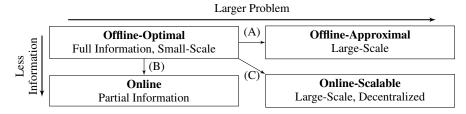


Figure 1: Applications of learning to optimization problems. (A) embodies techniques for learning optimization heuristics; (B) embodies techniques for learning to solve POMDPs; (C) is the emerging topic discussed here, embodying techniques for learning to coordinate large systems in real-world applications

^{*}All authors contributed equally.

The ideas here sit within a larger landscape of the application of learning to the solution of optimization problems. Consider Figure 1, where we consider 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 TSPs [13], VRPs [14], and general MILPs [15]. Along the information axis, which includes dealing with POMDPs, techniques such as RL play a major role, as well as ideas such as tuning Monte-Carlo Tree Search [16], embedding learned components into optimal control frameworks [17], and learning how to bias sampling planners [18].

Practical multi-robot planning and control builds on the progress along both of these axes: the degrees of freedom and environment complexity increase, while the ability to communicate and coordinate at scale decreases. 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 true optimality is NP-hard in many cases [5] and they will struggle when communications are degraded and frequent replanning is required. By contrast, *decentralized* approaches reduce the computational overhead [19] and relax the dependence on centralized units [20, 21] to deal with challenged communications, but account for purely local objectives and cannot explicitly optimize global objectives (e.g., path efficiency).

What the directions of Figure 1 teach us 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*, and 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.

Bridging the gap between the qualities of centralized and decentralized approaches, learning-based methods promise to find solutions that *balance optimality and real-world efficiency*. 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 systems. This agenda means that we need to address (i) how to generate multi-robot training data, (ii) how to generate decentralizable policies, and (iii) how to transfer these policies to real-world systems.

The following section elaborates these three key challenges and indicates promising directions.

2 Learning Decentralized Policies by Copying Centralized Experts

Though planning complexity is reduced with a decentralized approach, use of a learning-based approach requires consideration of state-action space coverage, especially since introducing multiple agents means the size of the joint state-action again grows exponentially. 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 [22, 23]), this position paper focuses on imitation learning strategies. The following paragraphs discuss three key topics that are central to the learning process: data generation, communication strategies, and sim-to-real transfer.

2.1 Experts and Data Generation

How to generate expert data? The work by Li et al. [4] shows 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. The former work considered the specific case-study of multiagent path planning, and used Conflict-Based Search (CBS) [24] to find optimal solutions (i.e., sets of optimal, collision-free paths). Although their results demonstrated unprecedented performance in decentralized systems (i.e., achieving higher than 96% success rates with single-digit flowtime increases, compared to the expert solution), but observed poor generalization. Simply training the models through behavior cloning leads to bias and over-fitting, since the performance of the network is intrinsically constrained by the dataset. Alternative approaches include learning curricula [25] 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 [26]. 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 [27]) is to introduce the expert during training to teach the learner how to recover from past mistakes. In [4], 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 [28]. 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. In addition, data augmentation allows us to understand the ability boundary of the trained model, to analyze the correlation between different factors, and to find identify factors that have the strongest effect on the system performance.

2.2 Communication Strategies 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 [29, 30, 3, 4]. 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. The 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 [29]. This produces a non-injective mapping from global states to latent states, where similar global states 'overlap', further improving generalization.

Are all messages equally important? Unfortunately, if communication happens concurrently and equivalently among many neighboring robots, it is likely to cause redundant information, burden the 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 [31], 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 [32]. Learning attention over static graphs has shown to be efficient. Liu et al. [33] developed a learning-based communication model that constructs the communication group on a static graph to address what to transmit and which agent to communicate to for collaborative perception. However, its permutation equivariance, time invariance and its practical effectiveness in dynamic multi-agent communication graphs have not yet been verified. Recently, Li et al. [34] 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. Their results show that it is possible to achieve performance close to that of a coupled centralized expert algorithm, while scaling to problem instances that are $\times 100$ larger than the training instances.

2.3 Sim-to-Real Transfer

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* [35].

Why is sim-to-real transfer difficult? Even though simulations have become more realistic and easily accessible over recent years [36, 37], it is 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 the physical interaction with the real world and physical constraints such as inertia, for example in robotic grasping [38, 39], drone flight [40, 41] or robotic locomotion [42, 43].

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. We already have established that communication is key to efficient multi-robot interaction, but it is not obvious 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. Furthermore, randomness such as message dropouts and delays are typically not considered during synchronous training. To the best of our knowledge, no research has been conducted that evaluates those factors and the impact 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 [23, 44].

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 [23, 44], yet leading to sub-optimal policies. More realistic (network) simulations [45] are always helpful, but also costly alternatives. Methods such as sim-to-real via real-to-sim [46] or training agents in the real-world in a mixed reality setting [47] 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 [48, 49].

3 Future Avenues

The sections above lay out the challenges entailed by the described approach. Yet, this begs the following two questions:

Is imitation learning the right paradigm? There are two main approaches to training a controller for a multi-robot system: imitation learning (e.g., [50]) and reinforcement learning (e.g., [51]). The most obvious benefit to RL is that it does not require an expert algorithm, as it simply optimizes a reward. However, 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. Finally, IL is inherently limited by the expert algorithm. As such, possible future directions should explore the combination of both IL and RL (e.g., [22]) in the context of decentralized multi-robot systems.

Is it possible to learn small-scale coordination patterns for large-scale systems? Ideally, we hope that controllers trained on only a few robots (which not only facilitates data generation, but also accelerates the training process), can then be deployed on large-scale systems with hundreds and even thousands of robots. Achieving this expectation may be within our reach. A recent example can be found in [52], 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 [34], 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 $\frac{1}{30}$ 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] A. Rajeswaran, K. Lowrey, E. V. Todorov, and S. M. Kakade. Towards generalization and simplicity in continuous control. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems* (NIPS), volume 30, pages 6550–6561. Curran Associates, Inc., 2017.
- [2] 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), pages 23–30, Sept. 2017. doi:10.1109/IROS.2017.8202133.
- [3] E. Tolstaya, F. Gama, J. Paulos, G. Pappas, V. Kumar, and A. Ribeiro. Learning Decentralized Controllers for Robot Swarms with Graph Neural Networks. *arXiv:1903.10527 [cs]*, Mar. 2019. URL http://arxiv.org/abs/1903.10527. arXiv: 1903.10527.
- [4] Q. Li, F. Gama, A. Ribeiro, and A. Prorok. Graph neural networks for decentralized multirobot path planning. In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 11785–11792. IEEE.
- [5] J. Yu and S. M. LaValle. Structure and intractability of optimal multi-robot path planning on graphs. In *AAAI*, 2013.
- [6] M. Schwager, D. Rus, and J.-J. Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.
- [7] I. E. Rabban and P. Tokekar. Improved resilient coverage maximization with multiple robots. *arXiv preprint arXiv:2007.02204*, 2020.
- [8] S. S. Ponda, L. B. Johnson, and J. P. How. Distributed chance-constrained task allocation for autonomous multi-agent teams. In *American Control Conference (ACC)*, pages 4528–4533, 2012.
- [9] A. Prorok. Redundant Robot Assignment on Graphs with Uncertain Edge Costs. In N. Correll, M. Schwager, and M. Otte, editors, *Distributed Autonomous Robotic Systems*, Springer Proceedings in Advanced Robotics, pages 313–327, 2019. ISBN 978-3-030-05816-6.
- [10] M. M. Zavlanos, L. Spesivtsev, and G. J. Pappas. A distributed auction algorithm for the assignment problem. In 2008 47th IEEE Conference on Decision and Control, pages 1212– 1217, Dec. 2008. doi:10.1109/CDC.2008.4739098. ISSN: 0191-2216.
- [11] N. Michael, M. M. Zavlanos, V. Kumar, and G. J. Pappas. Distributed multi-robot task assignment and formation control. In *IEEE International Conference Robotics and Automation*, pages 128–133, 2008.
- [12] B. Jung and G. S. Sukhatme. Cooperative multi-robot target tracking. In *Distributed autonomous robotic systems* 7, pages 81–90. Springer, 2006.
- [13] M. Deudon, P. Cournut, A. Lacoste, Y. Adulyasak, and L.-M. Rousseau. Learning heuristics for the tsp by policy gradient. In *International conference on the integration of constraint programming, artificial intelligence, and operations research*, pages 170–181. Springer, 2018.

- [14] M. Nazari, A. Oroojlooy, L. V. Snyder, and M. Takáč. Reinforcement learning for solving the vehicle routing problem. arXiv preprint arXiv:1802.04240, 2018.
- [15] E. B. Khalil, B. Dilkina, G. L. Nemhauser, S. Ahmed, and Y. Shao. Learning to run heuristics in tree search. In *IJCAI*, pages 659–666, 2017.
- [16] S. Katt, F. A. Oliehoek, and C. Amato. Learning in pomdps with monte carlo tree search. In *International Conference on Machine Learning*, pages 1819–1827. PMLR, 2017.
- [17] C. Richter, J. Ware, and N. Roy. High-speed autonomous navigation of unknown environments using learned probabilities of collision. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6114–6121. IEEE, 2014.
- [18] K. Liu, M. Stadler, and N. Roy. Learned sampling distributions for efficient planning in hybrid geometric and object-level representations. In 2020 IEEE International Conference on Robotics and Automation (ICRA), pages 9555–9562. IEEE, 2020.
- [19] V. R. Desaraju and J. P. How. Decentralized path planning for multi-agent teams with complex constraints. *Autonomous Robots*, 32(4):385–403, 2012. Publisher: Springer.
- [20] J. Van den Berg, M. Lin, and D. Manocha. Reciprocal velocity obstacles for real-time multiagent navigation. In *IEEE international conference on robotics and automation (ICRA)*, pages 1928–1935, 2008. tex.organization: IEEE.
- [21] B. Wang, Z. Liu, Q. Li, and A. Prorok. Mobile robot path planning in dynamic environments through globally guided reinforcement learning. *IEEE Robotics and Automation Letters*, 5(4): 6932–6939, 2020.
- [22] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, J. Oh, D. Horgan, M. Kroiss, I. Danihelka, A. Huang, L. Sifre, T. Cai, J. P. Agapiou, M. Jaderberg, A. S. Vezhnevets, R. Leblond, T. Pohlen, V. Dalibard, D. Budden, Y. Sulsky, J. Molloy, T. L. Paine, C. Gulcehre, Z. Wang, T. Pfaff, Y. Wu, R. Ring, D. Yogatama, D. Wünsch, K. McKinney, O. Smith, T. Schaul, T. Lillicrap, K. Kavukcuoglu, D. Hassabis, C. Apps, and D. Silver. Grandmaster level in starcraft ii using multi-agent reinforcement learning. In *Nature*, 2019. URL https://www.nature.com/articles/s41586-019-1724-z.pdf.
- [23] J. Blumenkamp and A. Prorok. The emergence of adversarial communication in multi-agent reinforcement learning. *Conference on Robot Learning (CoRL)*, 2020.
- [24] G. Sharon, R. Stern, A. Felner, and N. R. Sturtevant. Conflict-based search for optimal multiagent pathfinding. *Artificial Intelligence*, 219:40–66, 2015.
- [25] Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 41–48, 2009.
- [26] A. Attia and S. Dayan. Global overview of imitation learning. *arXiv preprint* arXiv:1801.06503, 2018.
- [27] S. Ross, G. Gordon, and D. Bagnell. 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*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.
- [28] J. Ho and S. Ermon. Generative adversarial imitation learning. *Advances in neural information processing systems*, 29:4565–4573, 2016.
- [29] R. Kortvelesy and A. Prorok. Modgnn: Expert policy approximation in multi-agent systems with a modular graph neural network architecture. *International Conference on Robotics and Automation (ICRA)*, 2021.
- [30] A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar. Graph policy gradients for large scale robot control. In *Conference on Robot Learning*, pages 823–834, 2020.

- [31] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.
- [32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. *International Conference on Learning Representations*, 2018.
- [33] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira. When2com: multi-agent perception via communication graph grouping. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4106–4115, 2020.
- [34] Q. Li, W. Lin, Z. Liu, and A. Prorok. Message-aware graph attention networks for large-scale multi-robot path planning. *IEEE Robotics and Automation Letters*, 6(3):5533–5540, 2021.
- [35] N. Jakobi, P. Husbands, and I. Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In *Advances in Artificial Life*, volume 929. Springer, 1995.
- [36] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16, 2017.
- [37] E. Coumans and Y. Bai. Pybullet, a python module for physics simulation for games, robotics and machine learning. http://pybullet.org, 2016–2021.
- [38] 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 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12619–12629, Los Alamitos, CA, USA, jun 2019. IEEE Computer Society. doi:10.1109/CVPR.2019.01291. URL https://doi.ieeecomputersociety.org/10.1109/CVPR.2019.01291.
- [39] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige, et al. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. In 2018 IEEE international conference on robotics and automation (ICRA), pages 4243–4250. IEEE, 2018.
- [40] A. Loquercio, E. Kaufmann, R. Ranftl, A. Dosovitskiy, V. Koltun, and D. Scaramuzza. Deep drone racing: From simulation to reality with domain randomization. *IEEE Transactions on Robotics*, 36(1):1–14, 2019.
- [41] E. Kaufmann, A. Loquercio, R. Ranftl, M. Müller, V. Koltun, and D. Scaramuzza. Deep drone acrobatics. In *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, July 2020. doi:10.15607/RSS.2020.XVI.040.
- [42] A. Nagabandi, I. Clavera, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *ICLR*, 2018.
- [43] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Simto-real: Learning agile locomotion for quadruped robots. In *Robotics: Science and Systems*, 2018. URL https://arxiv.org/pdf/1804.10332.pdf.
- [44] R. Mitchell, J. Blumenkamp, and A. Prorok. Gaussian process based message filtering for robust multi-agent cooperation in the presence of adversarial communication. *CoRR*, abs/2012.00508, 2020. URL https://arxiv.org/abs/2012.00508.
- [45] M. Calvo-Fullana, D. Mox, A. Pyattaev, J. Fink, V. Kumar, and A. Ribeiro. Ros-netsim: A framework for the integration of robotic and network simulators. *IEEE Robotics and Automation Letters*, 6(2):1120–1127, 2021.
- [46] J. Zhang, L. Tai, P. Yun, Y. Xiong, M. Liu, J. Boedecker, and W. Burgard. Vr-goggles for robots: Real-to-sim domain adaptation for visual control. *IEEE Robotics and Automation Letters*, 4(2):1148–1155, 2019.

- [47] R. Mitchell, J. Fletcher, J. Panerati, and A. Prorok. Multi-vehicle mixed reality reinforcement learning for autonomous multi-lane driving. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '20, page 1928–1930, Richland, SC, 2020. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450375184.
- [48] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.
- [49] B. Wang, J. Xie, and N. Atanasov. Coding for distributed multi-agent reinforcement learning. arXiv preprint arXiv:2101.02308, 2021.
- [50] H. M. Le, Y. Yue, P. Carr, and P. Lucey. Coordinated multi-agent imitation learning. In *International Conference on Machine Learning*, 2017. URL http://proceedings.mlr.press/v70/le17a/le17a.pdf.
- [51] 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*, NIPS'17, page 6382–6393, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.
- [52] M. Damani, Z. Luo, E. Wenzel, and G. Sartoretti. Primal₂: Pathfinding via reinforcement and imitation multi-agent learning lifelong. *IEEE Robotics and Automation Letters*, 6(2): 2666–2673, 2021.