Generalization of Heterogeneous Multi-Robot Policies via Awareness and Communication of Capabilities

Anonymous Author(s)
Affiliation
Address
email

Abstract: Recent advances in multi-agent reinforcement learning (MARL) provide promising tools for enabling heterogeneous multi-robot teaming without the need for manually-designed controllers. However, existing approaches often overlook the challenge of generalizing learned policies to teams of new compositions, sizes, and robots. While such generalization might not be important in teams of virtual agents that can retrain policies on-demand, it is pivotal in multi-robot systems that are deployed in the real-world and must adapt to inevitable changes without retraining. As such, multi-robot policies must remain robust to team changes – an ability we call *adaptive teaming*. In this work, we investigate if *awareness and communication of robot capabilities* can improve such generalization. We conduct detailed experiments involving a heterogeneous sensor network task implemented on an established multi-robot test bed. We demonstrate that shared decentralized policies, that enable robots to be both aware of and communicate their capabilities, achieve adaptive teaming by implicitly capturing the fundamental relationship between collective capabilities and effective coordination. Videos of trained policies can be viewed at https://sites.google.com/view/cap-comm.

Keywords: Heterogeneity, Multi-Robot Teaming, Generalization

1 Introduction

Heterogeneous robot teams have the potential to address complex real-world challenges that arise in a wide range of domains, such as precision agriculture, defense, warehouse automation, supply chain optimization, and environmental monitoring. However, a key hurdle in realizing such potential is the challenge of ensuring effective communication, coordination, and control.

Existing approaches to address the challenges of heterogeneous multi-robot systems can be crudely categorized into two groups. First, classical approaches leverage expertly-designed and well-understood controllers that enable simple local interaction rules to give rise to complex global emergent behavior [1]. Indeed, such controllers have proven extraordinarily useful in diverse domains. However, designing such controllers requires both significant technical expertise and considerable domain knowledge. Second, recent learning-based approaches alleviate the need for expertise and domain knowledge by taking advantage of advances in learning frameworks and the availability of computational resources. Such methods are proving to be successful in many domains, such as video games [2], autonomous driving [3], disaster response [4] and manufacturing task scheduling [5].

However, learning approaches are not without their fair share of limitations. First, the majority of existing methods focus on homogeneous teams and as such cannot handle heterogeneous multi-robot teams. Second, and more importantly, even existing methods designed for heterogeneous teams are often solely concerned with the challenge of learning to coordinate a given team, entirely ignoring the challenge of *generalizing* the learned behavior to new teams. Given the potentially-prohibitive cost of retraining coordination policies after deployment, it is imperative that multi-robot policies generalize learned behaviors to inevitable changes to the team.

In this work, we focus on the challenge of generalizing multi-robot policies to team changes. In particular, we focus on zero-shot generalization of trained policies to teams of new compositions,

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

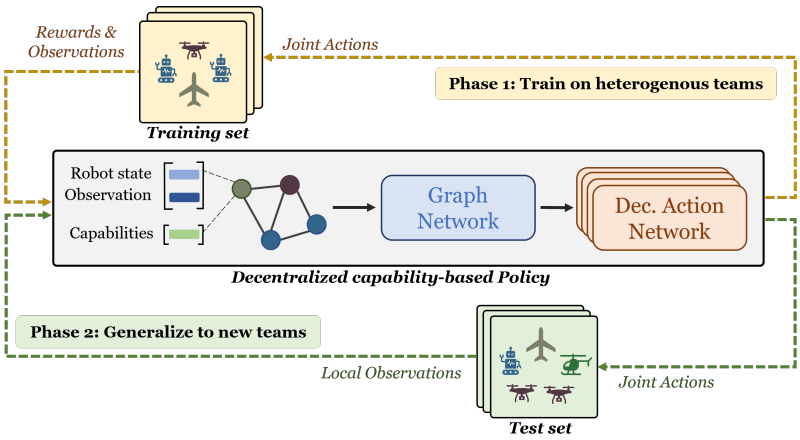


Figure 1: We investigate the role of capability awareness and communication in generalizing heterogeneous multi-robot coordination policies to teams of new composition, size, and robots.

sizes, and robots that are not encountered in training (see Fig. 1). We refer to such generalization as *adaptive teaming*, wherein the learning policy can readily handle changes to the team without additional training. To this end, we need policies that can reason about how a group of diverse robots can collectively achieve a common goal, without assigning rigid specialized roles to individual robots.

We investigate the role of *robot capabilities* in generalization to new teams. Our key insight is that adaptive teaming requires the understanding of how a team's diverse capabilities combine to influence the behavior of individual robots. For instance, consider an autonomous wildfire response team tasked with multiple concurrent locations. Effective coordination of these robots requires reasoning about the opportunities and constraints introduced by their individual and relative capabilities, such as speed, water capacity, and battery range. In general, robots must learn how their individual capabilities relate to those of others to determine their role in achieving shared objectives.

We develop a policy architecture capable of explicitly reasoning about robot capabilities when determining actions. Our architecture has four key properties: i) *capability awareness*: our design enables actions to be conditioned on continuous capabilities in addition to local observations, and ii) *capability communication*: we leverage graph networks to learn how robots must communicate their capabilities and build effective representations, iii) *robot-agnostic*: we utilize parameter sharing and learn policies that are not tied to individual robots, and iv) *decentralized*: our trained policies can be deployed in a decentralized manner. Together, these four properties provide the potential to generalize to new teams. One can view this design as an extension of agent identification techniques [6] to the metric space of capabilities. As such, capabilities do not merely serve to distinguish between heterogeneous agents during training and enable behavioral heterogeneity [7], but also provide a more general means to encode how individual and relative capabilities influence collective behavior.

We carry out detailed experiments to evaluate the utility of capability awareness and communication in a heterogeneous sensor network task on the Robotarium [8], a well-established multi-robot test bed. Our results reveal that both awareness and communication of capabilities are essential components of generalization, outperforming policies that lack either one or both of these features in terms of average returns and task-specific performance metrics. Further, capability-based policies achieve superior combinatorial generalization than existing agent identification-based baselines, while ensuring comparable performance on the training set.

2 Related Work

Learning for multi-robot teams: Recent advances in deep learning are providing promising approaches that circumvent the challenges associated with classical control of multi-robot systems. Multi-agent reinforcement learning (MARL), in particular, has been shown to be capable of solving a wide variety of tasks, including simple tasks in the multi-agent particle environments (MPE) [9], complex tasks under partial observability [10], coordinating an arbitrary number of agents in video games [2], and effective predictive modeling of multi-agent systems [11]. These approaches

are driven by popular MARL algorithms like Q-mix [12], MADDPG [9], and MAPPO [13] – non-trivial extensions of their single agent counterparts DQN, DDPG, and PPO, respectively. We adopt a PPO-based learning framework given its proven benefits despite its simplicity [13]. Centralized training, decentralized execution (CTDE) is a commonly used framework in which decentralized agents learn to take actions based on local observations while a centralized critic provides feedback based on global information [14, 15]. We use the CTDE paradigm as it lends itself naturally to multi-robot teams since observation and communication are often restricted. However, its important to note that our approach is agnostic to the specific learning algorithm.

Learning for heterogeneous teams: Many MARL algorithms were originally designed for use in homogeneous multi-agent teams. However, truly homogeneous multi-robot teams are rare except for large-scale swarms. Most real-world multi-robot problems such as search & rescue, agriculture, and surveillance require a diverse set of capabilities aggregated from heterogeneous robots [16–18]. While many MARL approaches consider heterogeneity, they either tend to focus on differences in behavior exhibited by physically identical robots [19], or identical behavior exhibited by physically-different robots [20, 21]. A common strategy used to elicit heterogeneous behavior from shared models is referred to as agent identification or behavioral typing, in which the agents' observations are appended with an agent-specific index [22, 23]. While these methods have been shown to be highly effective, recent investigations have revealed issues with scalability [24], and robustness to observation noise [7]. While capability-awareness is similar in spirit to existing identification-based techniques, it does not require assigning individual robots unique indexes and can thus generalize to differently-sized teams. Further, most existing methods do not simultaneously handle teams with physical and behavioral differences. According to a recent heterogeneous multi-robot taxonomy [7], our work falls under the category consisting of physically-different robots that differ in behavior, but share the same objective. Two recent approaches can belong to this same category [4, 7]. However, one is limited to a discrete set of robot types [4] and the other learned decentralized robot-specific policies that cannot handle the addition of new robots and might not generalize to new compositions [7].

Generalization: For applications to real-world multi-robot systems, it is essential to consider the generalization capabilities of learned control policies. In our formulation, there are two axes of generalization in heterogeneous multi-robot teams: combinatorial generalization (new team sizes and new compositions of the same robots) and individual capability generalization (new robots). Prior works reliant on feed forward or recurrent networks tend to be limited to teams of static size [9, 25]. Combinatorial generalization for homogeneous teams can be achieved with graph network-based policies [25, 26]. However, existing methods tend to struggle with generalization in the presence of heterogeneity [27]. While methods that employ agent identification [22, 23] might be able to achieve combinatorial generalization by reusing the IDs from training, it is unclear how they can handle new robots. In start contrast, capability-based policies are robot-agnostic and can take the capabilities of the new robot as an input feature to determine its actions.

3 Capability Awareness and Communication for Adaptive Teaming

In this section, we first model heterogeneous teams and then introduce policy architectures that enable capability awareness and communication, along with the associated training pipeline.

3.1 Modeling Heterogeneous Multi-Robot Teams

We model teams of N heterogeneous robots as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $v_i \in \mathcal{V}$ is a robot, and each edge $e_{ij} = (v_i, v_j) \in \mathcal{E}$ is a communication link. We use z_i to denote the observations of the ith robot, including information about the ego robot and the environment. We assume that the robots' heterogeneity stems from their capabilities. Formally, we represent the capabilities of ith robot by a real-valued vector $c_i \in \mathcal{C} \subset \mathbb{R}_+^C$, where $\mathcal{C}$ is the C-dimensional space of all capabilities considered across robots. Examples of multi-dimensional capabilities could include payload, speed, sensing radius, etc. When robot i does not posses the kth capability, we set the kth element of c_i to zero.

3.2 Problem Description

We are interested in learning a decentralized control policy that can i) effectively coordinate a team of heterogeneous robots to achieve the task objectives, and ii) generalize readily to teams of both novel compositions and novel robots that are not encountered during training. Our problem can be viewed as a Multi-Agent Reinforcement Learning (MARL) problem that can be formalized as a decentralized partially-observable Markov Decision Process (Dec-POMDP). We expand on the Dec-POMDP formulation to incorporate the *capabilities* of heterogeneous robots and arrive at the tuple $\langle \mathcal{D}, \mathcal{S}, \{\mathcal{A}_i\}, \{\mathcal{Z}_i\}, \mathcal{C}, T, R, O \rangle$ where $\mathcal{D}$ is the set of N robots, $\mathcal{S}$ is the set of global states, $\{\mathcal{A}_i\}$ is a set of actions spaces across all robots, $\{\mathcal{Z}_i\}$ is a set of joint observations across all robots, $\mathcal{C}$ is the multi-dimensional space of capabilities, R is the global reward function, and T and O are the joint state transition and observation models, respectively. Our objective is to learn decentralized action policies that can control each robot so as to maximize the expected return $\mathbb{E}[\sum_{t=0}^{T_h} r_t]$ over the task horizon T_h. The decentralized policy of the ith robot $\pi(a_i|\widetilde{o}_i)$ defines the probability that Robot i takes Action a_i given its effective observation $\widetilde{o}_i$. The effective observation $\widetilde{o}_i$ of the ith robot is a function of both its individual observation and that of others in its neighborhood.

3.3 Policy Architecture

To enable capability awareness and communication in multi-robot coordination policies, we designed a policy architecture that leverages graph convolutional network (GCNs) since a plethora of recent approaches attest to their ability to learn effective communication protocols and enable decentralized decision-making in multi-agent teams [28]. Further, since operations are local to nodes and can therefore generalize to graphs of any topology (i.e., permutation invariance [29]). We illustrate our architecture in Fig. 2 and explain its components below.

Capability Awareness: We argue that the heterogeneity of robots can have a significant impact on how the team must coordinate to achieve a task. Specifically, individual and collective capabilities can affect the roles the robots play within the team. For instance, consider a heterogeneous mobile robot team responding to wildfire incidents at multiple locations. To effectively respond, a robot within the team must account for its speed and water capacity. Therefore, it is necessary that robots are aware of their capabilities. To enable such awareness, we append each robot's capability vector c_i to its observations before passing them along as node features to the graph network. This information will help each robot condition its actions not just on observations, but also on its capabilities.

Capability Communication: In addition to awareness, communicating capability information can enable a team to reason about how its collective capabilities impact task performance. Revisiting our wildfire example, robots in the response team can effectively coordinate their efforts by implicitly and dynamically taking on roles based on their relative speed and water capacity. But such complex decision making is only possible if robots communicate with each other about their capabilities. Each node in our GCN-based policy receives capability information along with the corresponding robot's local observations so the learned communication protocol can help the team communicate and effectively build representations of the collective capabilities.

Note that our policy is robot-agnostic and learns the implicit and interconnected relationships between the observations, capabilities, and actions of all robots in the team. Further, as we demonstrate in our experiments, capability awareness and communication enables generalization to teams with new robot compositions, sizes, and even to entirely new robots as long as their capabilities belong to the same space of capabilities $\mathcal{C}$.

3.4 Training Procedure

We utilize parameter sharing to train a single action policy that is shared by all of the robots. Parameter sharing is known to improve learning efficiency by limiting the number of parameters. More importantly, parameter sharing is required for our problem so that action policies can transfer to new robots without the need for training new policies. Additionally, we believe that parameter sharing serves a secondary role in learning generalizable strategies for efficient generalization. Sharing parameters enables the policy to learn generalized coordination strategies that, when conditioned on robot capabilities and local observations, can be adapted to specific robots and contexts.

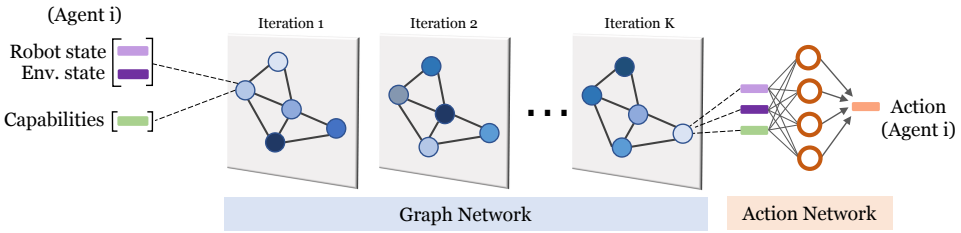


Figure 2: Our policy architecture enables robots to explicitly account for its capabilities and that of other robots in the team. We leverage a graph convolution network to facilitate communication of state and capability information among neighbors, and a parameter-shared action network to allow each agent take actions conditioned on individual but accumulated information.

We employ a centralized training, decentralized execution (CTDE) paradigm to train the action policy. We apply an actor-critic model, and train using proximal policy optimization (PPO). The actor-critic model is composed of a decentralized actor network (i.e., shared action policy) that maps robots observations to control actions, and a centralized critic network [30], which estimates the value of the team's current state based on centralized information about the environment and robots aggregated from individual observations. Finally, we trained our policies on multiple teams until convergence with the teams changing every 10 episodes to stabilize training.

4 Experimental Evaluation

We conducted three sets of experiments to evaluate how capability awareness and communication impact zero-shot combinatorial generalization in heterogeneous multi-robot policies as detailed below. First, we evaluate how the introduction of capabilities into the policy architecture impacts training. Second, we investigate generalization to new team compositions of robots encountered during training. Third, we investigate generalization to teams made of unseen robots.

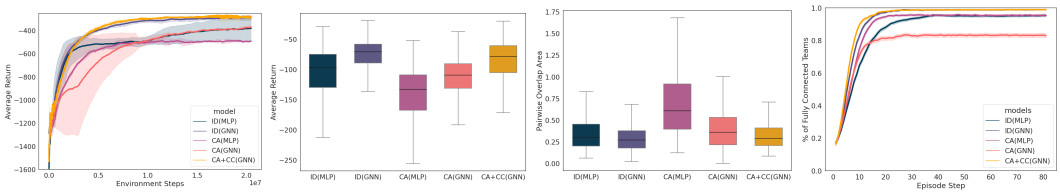


Figure 3: Evaluation of learned policies on the training set.

4.1 Experimental Design

Environment: We designed a multi-robot heterogeneous sensor network task in the Robotarium [8], a remotely accessible test bed to conduct real multi-robot coordination experiments. The heterogeneous sensor network task requires a robot team to form a single fully connected sensor network, while minimizing sensing area overlap to maximize area coverage. Each Robot i in the sensor network has a circular sensing-communication area with radius c_i centered at its position. Robots i and j are connected if their sensing areas overlap. We created the heterogeneous sensor network environment on top of the Robotarium's simulator, that simulates realistic unicycle robot dynamics, for training and evaluating the policies. This enables the policies to be directly deployed on the real robots in the physical Robotarium.

Policy architectures: In order to systematically examine the impact of capability awareness and communication, we consider the following robot action policy architectures:

- `ID(MLP)`: Robot ID-based MLP
- `ID(GNN)`: Robot ID-based GNN
- `CA(MLP)`: Capability-aware MLP
- `CA(GNN)`: Capability-aware GNN without communication of capabilities,

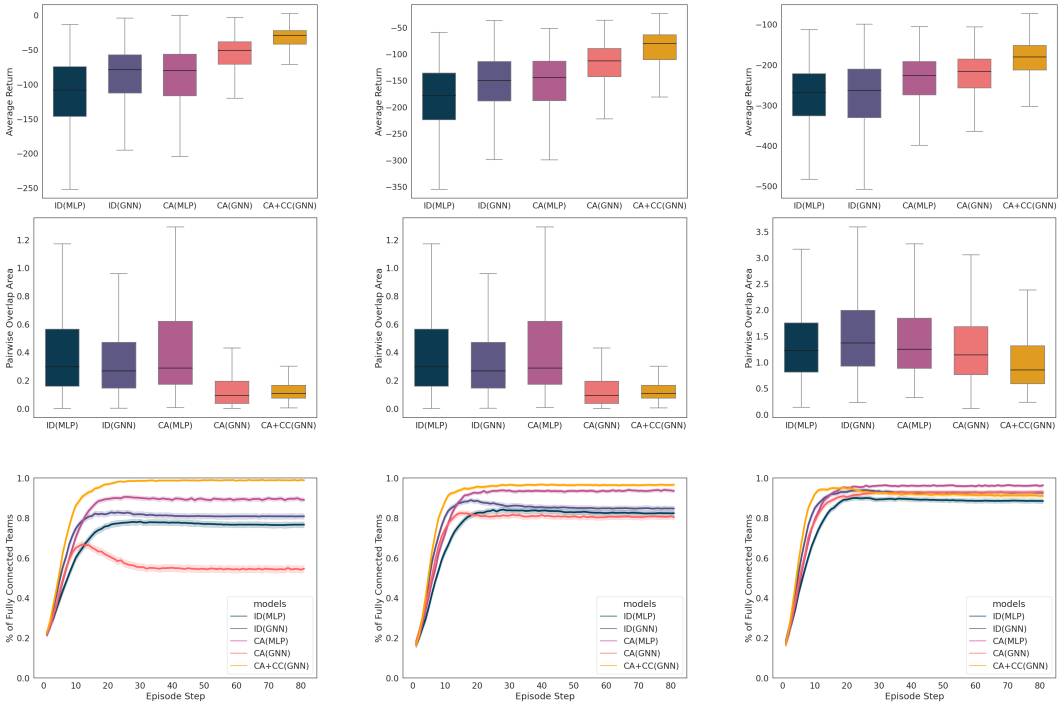
- `CA+CC(GNN)`: Both capability awareness and communication.

The ID-based baselines stand in for SOTA approaches that employ behavioral typing to handle heterogeneous teams [22, 23], and as such question the need for capabilities. The MLP based baselines help us investigate the need for communication. Finally, the `CA(GNN)` enables communication of observations but does not does not communicate capability information.

Metrics: We evaluated all policies based on three metrics: i) *Average Returns*: average joint reward received over the task horizon (higher is better), ii) *Pairwise Overlap*: sum of pairwise overlap in coverage area among robots (lower is better), iii) *% Fully Connected*: Percentage of teams (in train or test set) that managed to form a sensor network that connects all of the robots (higher is better).

Training: The training set composed of five robot teams with four robots each. The training teams were hand picked such that the team compositions and degree of robot capability heterogeneity were diverse. Robots' sensing capabilities were randomly sampled from a uniform distribution $c_i \sim U(0.2, 0.6)$. Each robot was also assigned a one-hot ID for the robot ID-based policies. We trained each policy with 3 random seeds for 20 million time steps. We resampled robot teams every 10 episodes, as we found that to be necessary to stabilize learning across multiple teams.

4.2 Training Performance



(a) 3 Robots (b) 4 Robots (c) 5 Robots

Figure 4: Comparison of policy architectures generalizing to new robot teams composed of robots from training teams.

We first focus on the training curves and subsequent evaluations conducted on the training set, without considering generalization. The goal of this experiment is to ensure introducing capabilities does not negatively impact training. The learning curves in terms of average return are shown in Fig. 3. All models achieved convergence within 20 million environment steps, with `ID(GNN)` and `CA+CC(GNN)` exhibited both the fastest convergence and the highest returns. These results suggest that communication of individual robot features, whether based on IDs or capabilities, improves learning efficiency and performance for heterogeneous coordination.

Results reported in Fig. 3 compare the trained decentralized policies for the training set of teams for all three metrics. All policies except `CA(MLP)` performed comparably and were able to effec-

tively learn to maximize expected returns, achieve a fully connected sensor network, and minimize the pairwise overlap in coverage area. Further, `CA+CC(GNN)` and `ID(GNN)` perform similarly but marginally better than the other baselines. `CA(MLP)`'s suboptimal performance indicates that capability awareness in isolation without any communication hurts performance.

Taken together, these results suggest that capability awareness and communication can lead to effective training in heterogeneous teams. ID-based methods are able to perform at a similar level – but this is to be expected given that we conducted these evaluations on the training set and IDs are sufficient to implicitly assign roles and coordinate heterogeneous robots within known teams.

4.3 Zero-Shot Generalization to New Team Compositions

In this experiment, we evaluate how the trained policies generalize to team compositions and sizes not encountered during training. Fig. 4 shows plots quantifying each policy's ability to generalization to new compositions with team sizes of 3, 4, and 5 robots. To ensure a fair comparison, we evaluated all policies on the same set of 100 teams by randomly sampling novel compositions of the robots from the training set. We evaluated each policy on each test team across 10 episodes per seed. Given that this evaluation involved no new individual robots, we reused each robot's ID from the training set to facilitate the evaluation of ID-based policies.

We found that `CA+CC(GNN)` achieved the best average returns (highest mean and lowest variance) across all team sizes. Both ID-based methods (`ID(MLP)` and `ID(GNN)`) resulted in lower returns compared to all three capability-awareness baselines. This is in stark contrast to the results for the training set in Sec. 4.2, demonstrating that IDs alone do not help generalize learned policies to new team compositions. The poor generalization performance of ID-based methods is likely because they fail to implicitly reason about robot heterogeneity, and instead overfit the relationships between robot IDs and behavior in the training set. Further, `CA+CC(GNN)` in particular consistently outperforms all other policies across metrics and variations. This supports our hypothesis that both capability-awareness and communication are necessary to improve generalization.

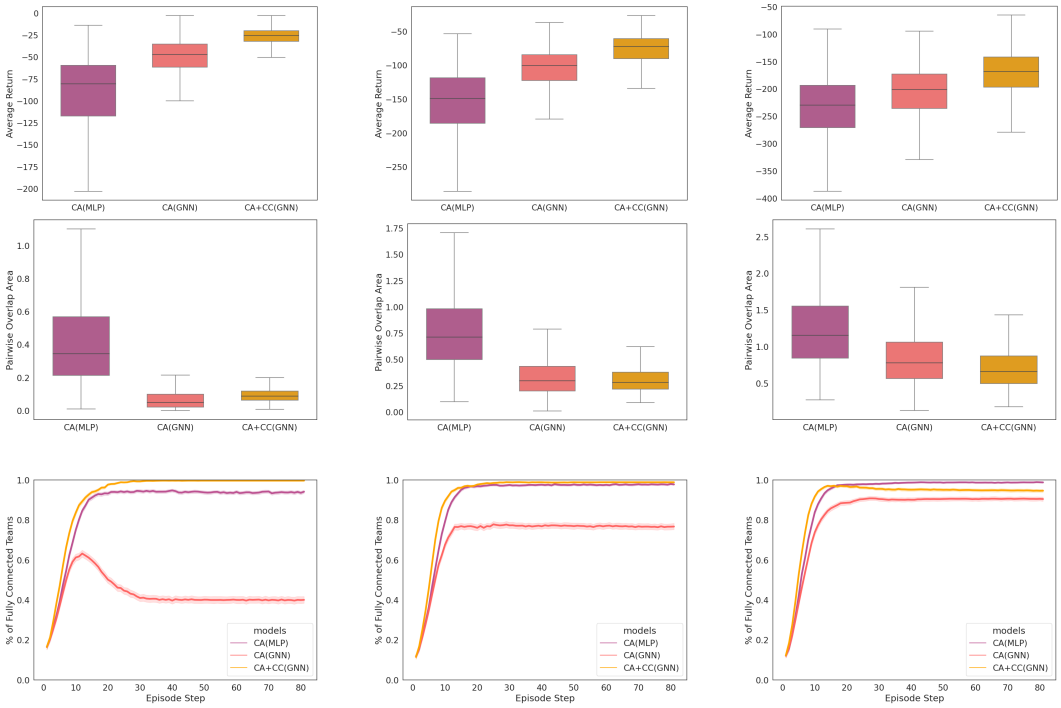
4.4 Zero-Shot Generalization to New Robots

Finally, we evaluate the trained polices' ability to generalization to teams composed of entirely new robots whose sensing radii are different from those encountered in training. Similar to the previous experiment, we also evaluate the generalization to new team sizes. Note that it is not possible to generalized ID-based policies to teams with new robots, since there is no obvious way to assign IDs to the the new robots. As such, we compare the performance of all three capability-based policies.

In Fig. 5, we report the performance of all three capability-based policies in terms of all three metrics. Both GNN based policies (`CA(GNN)` and `CA+CC(GNN)`) considerably outperform the `CA(MLP)` in terms of both average return and pairwise overlap, underscoring the importance of communication in generalization to teams with new robots. However, we also see that communication of observations alone is insufficient, as evidenced by the fact that `CA+CC(GNN)`, which communicates both observations and capabilities, consistently outperforms `CA(GNN)`, which only communicates observations, in terms of all three metrics. A curious observation from this experiment is the anomalous suboptimal performance of `CA(GNN)` in terms of % fully connected. We suspect this is because `CA(GNN)` fails to maintain a fully-connected sensor network in a subset of teams compared to the other two baselines.

4.5 Real-Robot Demonstrations

We also deployed the trained policies of `CA+CC(GNN)`, `CA(MLP)`, and `CA(GNN)` on the physical Robotarium (see Supplementary Matieral for visuals). Overall, we find that the benefits reported above extend to physical robot teams. We find that `CA+CC(GNN)` and `CA(GNN)` policies generalize to physical robots and successfully build a sensor network while minimizing sensing overlap for teams of 3 and 4 robots. The `CA(MLP)` policy resulted in significantly worse performance, wherein robots' executed paths provoked significant engagement of the Robotarium's barrier certificates due to potential collisions.



(a) 3 Robots (b) 4 Robots (c) 5 Robots

Figure 5: Comparison of policy architectures generalizing to robot teams comprised of new robots.

5 Limitations

While our framework allows for multi-dimensional capability vectors, our experiments only involved variations in a single capability. However, we expect capability-based policies to generalize in a multi-dimensional capability space since recent evidence suggests that high-dimensional one-hot encoding work well for learning control policies [27]. Further, we do not consider high-level planning and task-allocation and rely solely on the learning framework to perform implicit assignments to sub-tasks within the macro task. There is a rich body of literature that addresses hand-crafted methods for efficient allocations for trait-based multi-robot teams that outperform learning-based methods. Lastly, in this work we discuss decentralized control but employ a graph neural network that uses a fully connected communication graph, for simplicity. We believe this is a restrictive, yet sound, assumption based on the proven success of graph networks in decentralized control [25, 31].

6 Conclusion

This paper investigated the awareness and communication of robot capabilities for generalization of heterogeneous multi-robot policies. We developed a graph network-based decentralized policy architecture based on parameter sharing that enables robots to communicate their observations and capabilities to achieve effective coordination.

We conducted detailed experiments on a robot heterogeneous sensor network task implemented on the established Robotarium multi-robot test bed. Our results in evaluation testing generalization to new robot team compositions, sizes, and robots show that robots which are both aware of and communicate their capabilities achieve improved ability to adapt in new teams.

References

[1] J. Cortés and M. Egerstedt. Coordinated control of multi-robot systems: A survey. *SICE Journal of Control, Measurement, and System Integration*, 10(6):495–503, 2017.

[2] B. Ellis, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning, 2022. URL https://arxiv.org/abs/2212.07489.

[3] S. Shalev-Shwartz, S. Shammah, and A. Shashua. Safe, multi-agent, reinforcement learning for autonomous driving, 2016.

[4] E. Seraj, Z. Wang, R. Paleja, D. Martin, M. Sklar, A. Patel, and M. Gombolay. Learning efficient diverse communication for cooperative heterogeneous teaming. In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '22, page 1173–1182, Richland, SC, 2022. International Foundation for Autonomous Agents and Multiagent Systems. ISBN 9781450392136.

[5] Z. Wang, C. Liu, and M. Gombolay. Heterogeneous graph attention networks for scalable multi-robot scheduling with temporospatial constraints. *Autonomous Robots*, 46(1):249–268, Jan. 2022. ISSN 1573-7527. doi:10.1007/s10514-021-09997-2.

[6] C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang. Celebrating Diversity in Shared Multi-Agent Reinforcement Learning, Nov. 2021.

[7] M. Bettini, A. Shankar, and A. Prorok. Heterogeneous multi-robot reinforcement learning, 2023.

[8] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt. The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems. *IEEE Control Systems Magazine*, 40(1): 26–44, 2020.

[9] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Neural Information Processing Systems (NIPS)*, 2017.

[10] S. Omidshafiei, J. Pazis, C. Amato, J. P. How, and J. Vian. Deep decentralized multi-task multi-agent reinforcement learning under partial observability. In *International Conference on Machine Learning*, pages 2681–2690, 2017.

[11] Y. Hoshen. Vain: Attentional multi-agent predictive modeling. In *Advances in Neural Information Processing Systems*, pages 2701–2711, 2017.

[12] T. Rashid, M. Samvelyan, C. S. de Witt, G. Farquhar, J. Foerster, and S. Whiteson. Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning, 2018.

[13] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu. The surprising effectiveness of ppo in cooperative, multi-agent games, 2022.

[14] V. Egorov and A. Shpilman. Scalable multi-agent model-based reinforcement learning, 2022.

[15] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson. Counterfactual multi-agent policy gradients, 2017.

[16] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter. Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 10 2020. doi:10.1002/rob.21993.

[17] T. Dang, M. Tranzatto, S. Khattak, F. Mascarich, K. Alexis, and M. Hutter. Graph-based subterranean exploration path planning using aerial and legged robots. *Journal of Field Robotics*, 37(8):1363–1388, 2020. doi:https://doi.org/10.1002/rob.21993. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21993.

[18] G. Gil, D. E. Casagrande, L. P. Cortés, and R. Verschae. Why the low adoption of robotics in the farms? challenges for the establishment of commercial agricultural robots. *Smart Agricultural Technology*, 3:100069, 2023. ISSN 2772-3755. doi:https://doi.org/10.1016/j.atech.2022.100069. URL https://www.sciencedirect.com/science/article/pii/S277237552200034X.

[19] C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang. Celebrating diversity in shared multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 3991–4002, 2021.

[20] J. K. Terry, N. Grammel, S. Son, and B. Black. Parameter sharing for heterogeneous agents in multi-agent reinforcement learning, 2022.

[21] C. Wakilpoor, P. J. Martin, C. Rebhuhn, and A. Vu. Heterogeneous multi-agent reinforcement learning for unknown environment mapping. *arXiv preprint arXiv:2010.02663*, 2020.

[22] J. Foerster, I. A. Assael, N. De Freitas, and S. Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.

[23] J. K. Gupta, M. Egorov, and M. Kochenderfer. Cooperative multi-agent control using deep reinforcement learning. In *Autonomous Agents and Multiagent Systems: AAMAS 2017 Workshops, Best Papers, São Paulo, Brazil, May 8-12, 2017, Revised Selected Papers 16*, pages 66–83. Springer, 2017.

[24] F. Christianos, G. Papoudakis, M. A. Rahman, and S. V. Albrecht. Scaling multi-agent reinforcement learning with selective parameter sharing. In *International Conference on Machine Learning*, pages 1989–1998. PMLR, 2021.

[25] A. Agarwal, S. Kumar, and K. Sycara. Learning transferable cooperative behavior in multi-agent teams, 2019.

[26] Q. Li, W. Lin, Z. Liu, and A. Prorok. Message-aware graph attention networks for large-scale multi-robot path planning, 2021.

[27] A. Mahajan, M. Samvelyan, T. Gupta, B. Ellis, M. Sun, T. Rocktäschel, and S. Whiteson. Generalization in cooperative multi-agent systems, 2022.

[28] A. Agarwal, S. Kumar, and K. Sycara. Learning transferable cooperative behavior in multi-agent teams, 2019. URL https://arxiv.org/abs/1906.01202.

[29] A. Khan, E. Tolstaya, A. Ribeiro, and V. Kumar. Graph Policy Gradients for Large Scale Robot Control. In *Proceedings of the Conference on Robot Learning*, pages 823–834. PMLR, May 2020.

[30] C. Yu, A. Velu, E. Vinitsky, Y. Wang, A. M. Bayen, and Y. Wu. The surprising effectiveness of MAPPO in cooperative, multi-agent games. *CoRR*, abs/2103.01955, 2021. URL https://arxiv.org/abs/2103.01955.

[31] Q. Li, F. Gama, A. Ribeiro, and A. Prorok. Graph neural networks for decentralized multi-robot path planning. *CoRR*, abs/1912.06095, 2019. URL http://arxiv.org/abs/1912.06095.