Supplementary Materials

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

Anonymous Author(s)
Affiliation
Address
`email`

1 Training and Evaluation Robot Teams

This section describes the design of the training teams, and the sampling of evaluation teams for the heterogeneous sensor network environment. To learn generalized coordination behavior, the training teams were required to be diverse in terms of composition and capture the underlying distribution of robot capabilities. To design these training teams we first binned robot capabilities into `small`, `medium`, and `large` sensing radii with bin ranges $[0.2m, 0.33m]$, $[0.33m, 0.46m]$, and $[0.46m, 0.60m]$ respectively. We then generated all possible combinations with replacement for teams composed of four robots of `small`, `medium`, and `large` robots for a total of 15 total teams. Each robot assigned to one of the bins `small`, `medium`, and `large` had it's capability (i.e. sensing radius) uniformly sampled within the bin range. This resulted in 15 total teams, for which we hand-selected 5 sufficiently diverse teams to be the training teams. The resulting training teams are given in Table 1.

Training Team Number	Robot Sensing Radii in Meters
1	$(0.2191, 0.2946, 0.2608, 0.3668)$
2	$(0.2746, 0.2746, 0.5824, 0.5756)$
3	$(0.3178, 0.3467, 0.5317, 0.6073)$
4	$(0.2007, 0.5722, 0.5153, 0.4622)$
5	$(0.4487, 0.5526, 0.5826, 0.58343)$

Table 1: Training teams. 5 teams of 4 robots

The evaluation robot teams were sampled differently for the different experimental evaluations performed. In the training evaluation experiment, the teams were the same as the training teams in Table 1. Teams for the generalization experiment to new team compositions, but not new robots, were sampled randomly from the 20 robots from the training teams (with replacement). Each robot from the pool of 20 robots was sampled with equal probability. In contrast, teams for the generalization experiment to new robots were generated by randomly sampling new robots, where each robot's sensing radius was sampled from a uniform distribution independently $U(0.2m, 0.6m)$. For the two generalization experiments, 100 total teams were sampled. Each algorithm was evaluated on the same set of sampled teams by fixing the pseudo random number generator's seed.

2 Graph Neural Networks

We employ a graph convolutional network (GCN) architecture for the decentralized policy π_i, which enables robots to communicate for coordination according to the robot communication graph $\mathcal{G}$.

A GCN is composed of L layers of graph convolutions, followed by non-linearity. In this work, we consider a single graph convolution layer applied to node i is given by

$$h_i^{(l)} = \sigma\Big(\sum_{j \in \mathcal{N}(i) \cup i} \phi_\theta(h_j^{(l-1)})\Big)$$

where $h_j^{(l-1)} \in \mathbb{R}^F$ is the node feature of node $j, \mathcal{N}(i) = \{j|(v_i, v_j) \in \mathcal{E}\}$ are all nodes j connected to i, ϕ_θ is node feature transformation function with parameters θ, σ is a non-linearity (e.g. Relu), and $h_i^l \in \mathbb{R}^G$ is the output node feature.

3 Policy Architectures

Each of the graph neural networks in the GNN-based policy architectures evaluated are composed of an input encoder network, a message passing network, and an action output network. The encoder network is a 2-layer MLP with hidden dimensions of size 64. For the message passing network, a single graph convolution layer composed of 2-layer MLPs with ReLU non-linear activations. The action output network is additionally a 2-layer MLP with hidden dimensions of size 64. The learning rate is 0.005.

`MLP(ID)`/`MLP(CA)`: The MLP architectures compose of a 4-layer multi-layer perceptron with 64 hidden units at each layer and ReLU() non-linearities.

`CA(GNN)`/`CA+CC(GNN)`/`ID(GNN)`: Each of the graph neural networks compose of an input "encoder" network, a message passing network, and an action output network. The encoder network and the action output network are multi-layer perceptrons with hidden layers of size 64, ReLU non-linear activations, and with one and two hidden layers respectively. The message passing network is a graph convolution layer wherein the linear transformation of node features (i.e. observations) is done by a 2-layer MLP with ReLU non-linear activations and 64 dimensional hidden units, followed by a summation of the transformed neighboring node features. The ouptut node features a concatenated with the output feature from the encoder network. This concatenated features is the input to the two out action network. The `CA(GNN)` network doesn't communicate the robot's capabilities with the graph convolution layers. Rather, the capabilities are appended to the output of the encoder network and output of node features of the graph convolution layer just before the the action network. Thus, the the action network is the only part of this model that is conditioned on robot capabilities.

4 Policy Training Hyper parameters

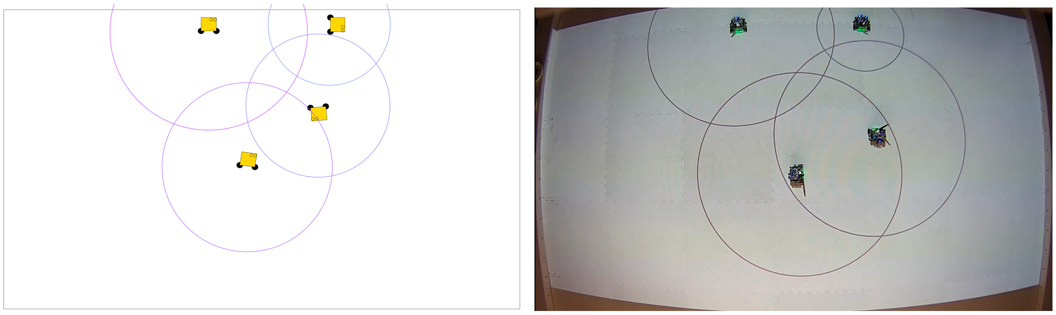
We detail the hyperparameters used to train each of the policies using proximal policy optimization (PPO) [1] in Table 2.

Hyperparameter	Value
Action Selection (Training)	`soft action selection`
Action Selection (Testing)	`hard action selection`
Critic Network Update Interval	200 steps
Learning Rate	0.0005
Entropy Coefficient	0.01
Epochs	4
Clip	0.2
Q Function Steps	5
Buffer Length	64
Number of training steps	20×10^6

Table 2: Training hyperparameters.

5 Environment

Robots have five available actions: they can move left, right, up, down, or stop. After selecting an action, the robots move in their selected direction for slightly less than a second before selecting a new action. The robots start at random locations least 30cm apart from each other, move at $\sim$21cm/second, and utilize barrier certificates [2] that takes effect at 17cm away to ensure they do not collide when running in the physical Robotarium.



(a) Our agents running in Simulation (b) Our agents running in the physical Robotarium

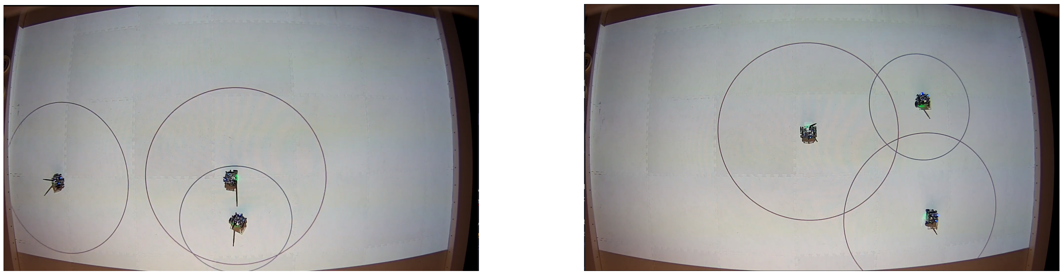
The reward from the heterogeneous sensor network environment is a shared reward. We describe the reward below:

$$D(i,j) = ||p(i) - p(j)|| - (c_i + c_j)$$
$$r(i,j) = \begin{cases} -0.9 * |D(i,j)| + 0.05, & \text{if } D(i,j) < 0 \\ -1.1 * |D(i,j)| - 0.05, & \text{otherwise} \end{cases}$$
$$R = \sum_{i<j}^{N} r(i,j)$$

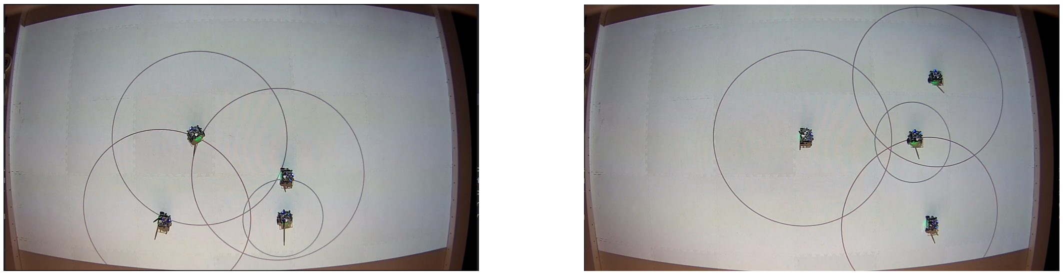
where i and j are robots, $p(i)$ is the position of robot i, c_i is the (capability) sensing radius of robot i, and R is the cumulative team reward shared by all the robots. The above reward is designed to reward the team when robots connected their sensing regions while minimizing overlap so as to maximize the total sensing area.

6 Robotarium Experiment Pictures

We show figures from real robot demonstrations of the trained capability-aware communication policy:



(a) Beginning of episode (3 robots). (b) End of episode (3 robots).



(c) Beginning of episode (4 robots). (d) End of episode (4 robots).

Figure 2: Demonstrations of `CA+CC(GNN)` policy deployed to real robot teams in the Robotarium testbed.

References

[1] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms. 2017. doi:10.48550/ARXIV.1707.06347.

[2] 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.