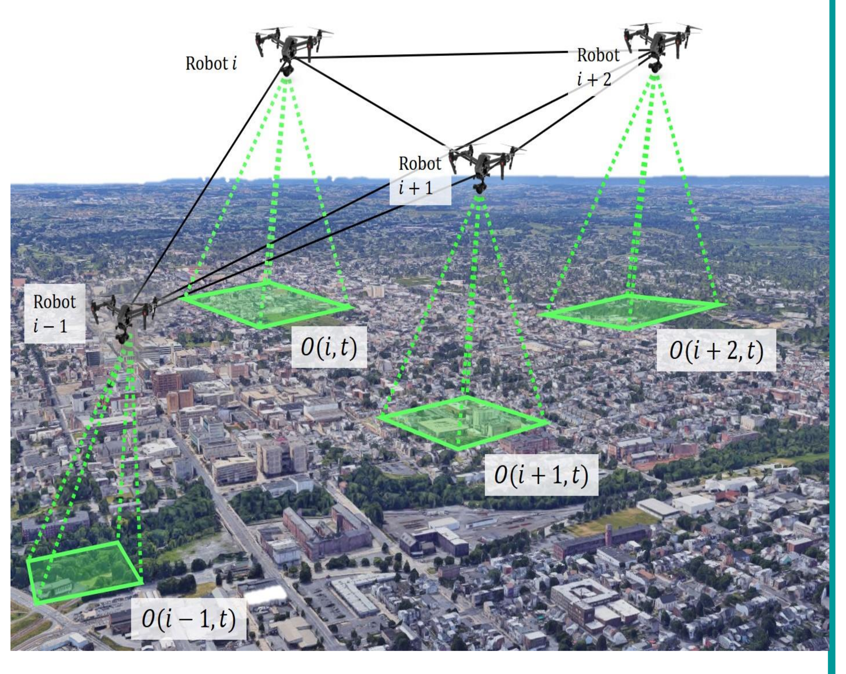
Classification-Aware Path Planning of Network of Robots

Guangyi Liu, Arash Amini, Martin Takáč, and Nader Motee
Lehigh University, Bethlehem, Pennsylvania

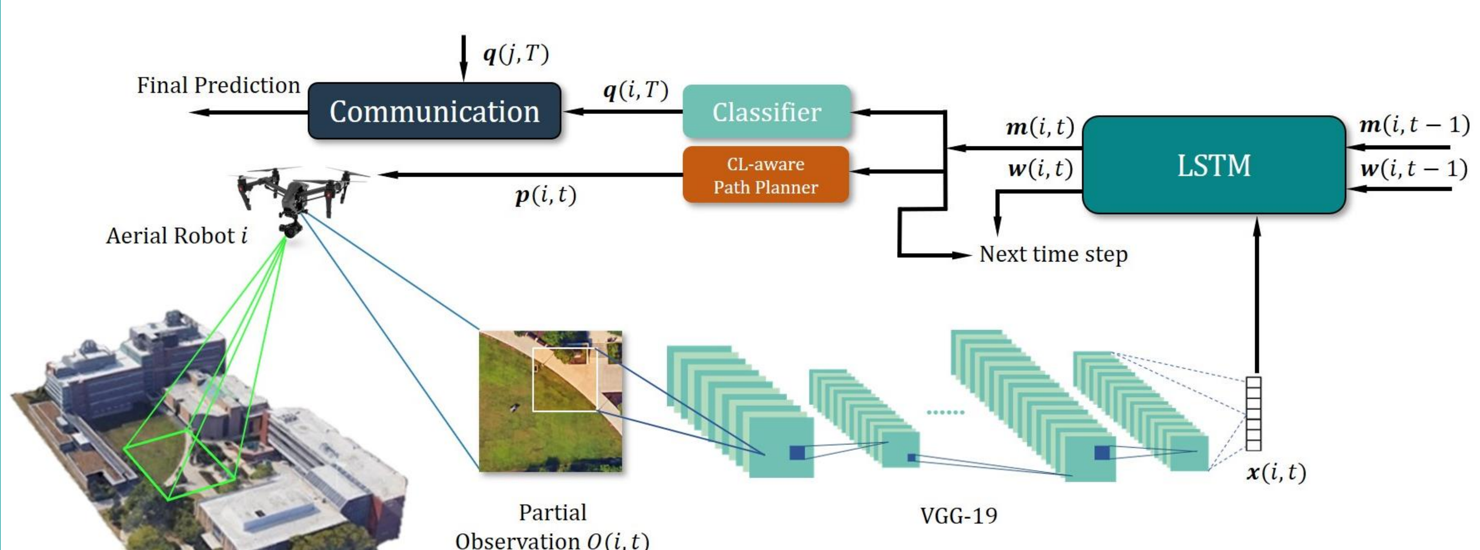
Problem Statement

- N identical robots: parameter sharing.
- Partial observations (pose-dependent) per agent.
- Capable of taking actions (i.e., navigation).
- Capable of communication over a complete network.
- Desired task: *Multi-Robot Path Planning for Classification*



A snapshot of UAVs classifying the target environment.

Distributed Classification Architecture



Localized Feature Extraction

- For the i'th robot, the localized visual input $O(i,t)$ is fed into a pre-trained VGG-19 model V_{θ_1}.
- The output of the VGG-19 model presented by a feature vector $x(i,t)$, such that

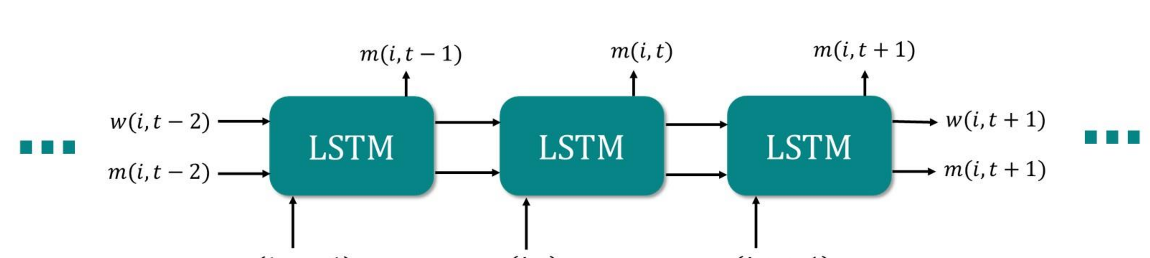
$$x(i,t) = V_{\theta_1}\big(O(i,t)\big)$$

in which θ_1 is the trainable parameter vector of the VGG-19 model.

- The output feature vector $x(i,t)$ contains the extracted classification features from the i'th robot at time t.

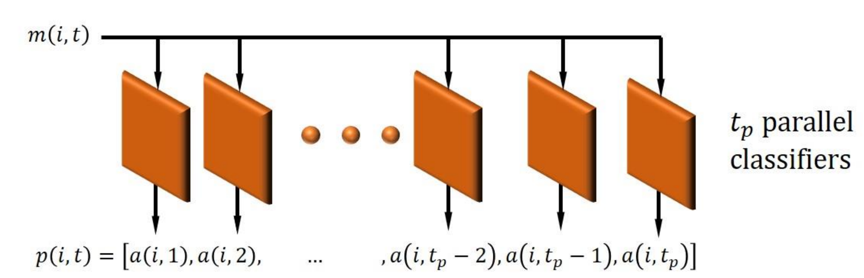
LSTM Feature Encoding

We use a Long-Short Term Memory (LSTM) cell to store the feature vectors of the robot. The feature vector $x(i,t)$ is treated as the input into the LSTM cell, while the hidden states $m(i,t)$ of the LSTM is utilized as the output for both classification and path planning purposes.



Classification-Aware Path Planning

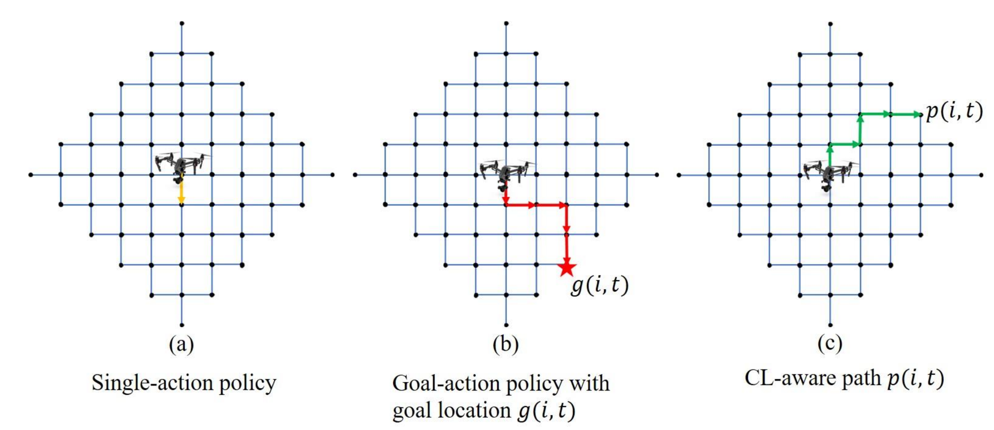
- Instead of using the single action motion planning or the goal-based motion planning, we introduce a **classification-aware path planner** that utilizes t_p parallel NNs to generate the actions for the next t_p steps in a classification manner.



- The CL-aware path planner utilizes the feature memory $m(i,t)$ to sample the actions for $\{t, t+1, \ldots, t+t_p\}$ time steps, which is shown by

$$p(i,t) = P_{\theta_3}\big(m(i,t)\big)$$

- This manipulation highlights the dependencies on **both long-term and short-term** classification rewards.



Communication and Map Classification

- **Classification:** Robots use the feature memory $m(i,t)$ to classify the map. The classifier is constructed with fully connected layers,

$$q(i,T) = C_{\theta_4}\big(m(i,T)\big)$$

- **Communication:** We assume all robots are connected via a complete graph. They exchange and fuse their prediction vector via a consensus type communication module. The global prediction is presented by

$$q = 1/N \sum_{i=1}^{N} q(i,T).$$

- The predicted label is presented by $\arg\max q$.

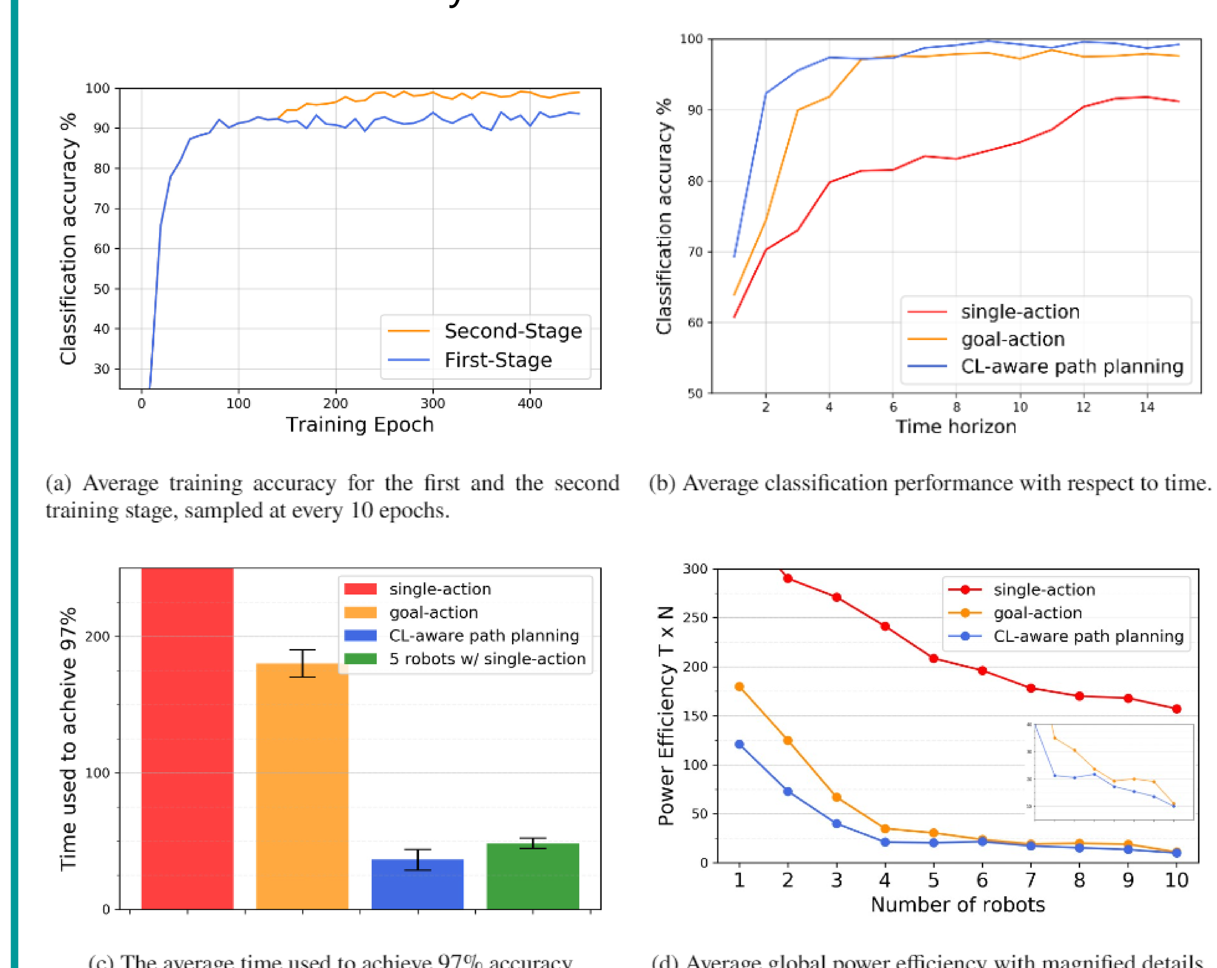
Satellite Map Dataset

- We created a campus map dataset from Google Earth to serve as the simulation dataset.

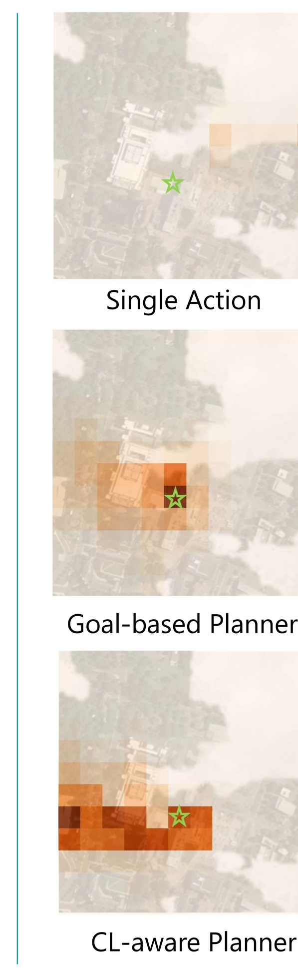
- The examples above shows the changes in both seasons and years for maps with the same label.

Simulation and Testing Results

- We validate the usefulness of our method in both Satellite Map Dataset and the MNIST dataset in PyTorch.



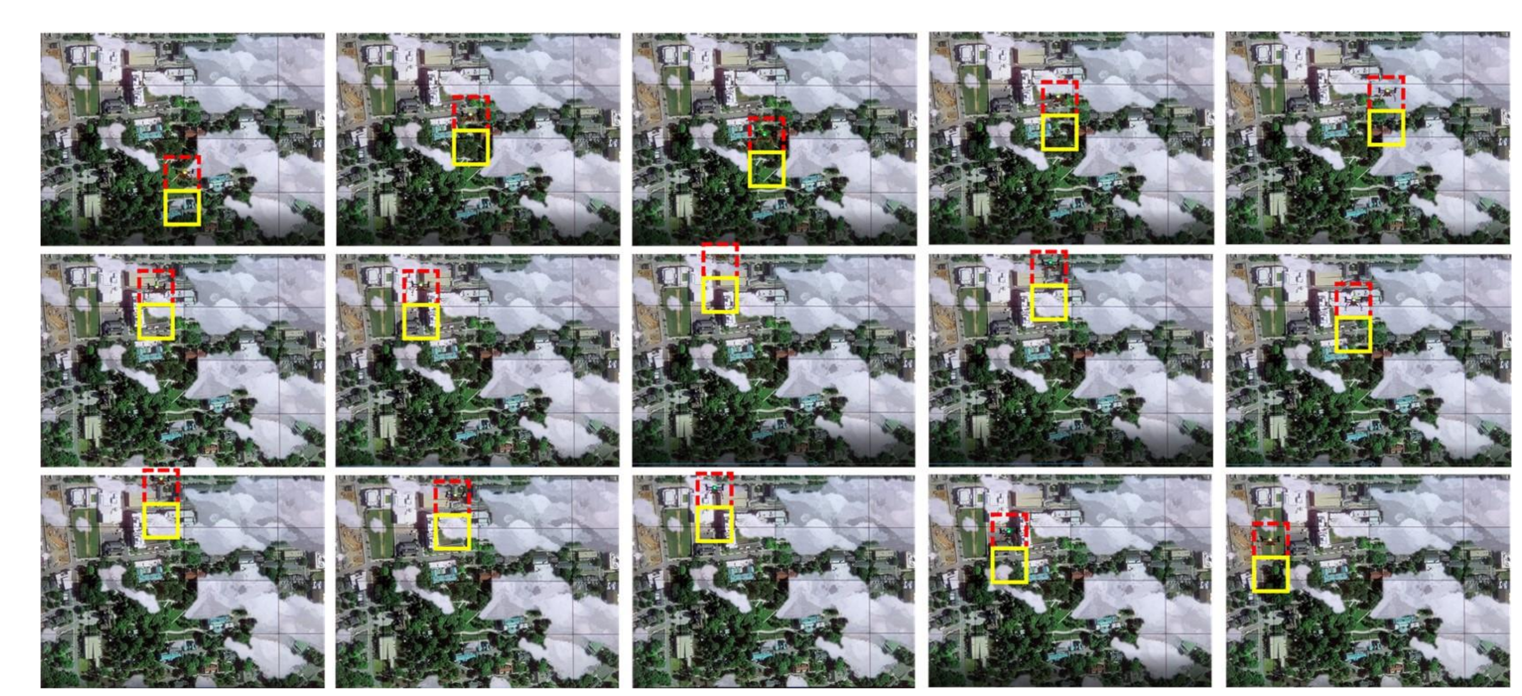
(a) Average training accuracy for the first and the second training stage, sampled at every 10 epochs.
(b) Average classification performance with respect to time.
(c) The average time used to achieve 97% accuracy.
(d) Average global power efficiency with magnified details.



- Our proposed method shows a significant improvement from other methods and has a comparable performance w.r.t. the centralized approach.

Table 1: Average optimal performance with $T = 15$ (%).

Action Policy	1 robot	5 robots	10 robots	20 robots	VGG-19 w/ full map	Average optimality gap (%)
Single-action	67.59	91.68	93.34	95.77		12.33
Goal-action	72.42	97.30	97.56	98.14	99.43	8.08
CL-aware	81.95	98.38	98.68	99.21		4.88



- Snapshot taken for a single quadcopter trying to classify the map of Lehigh University.

Conclusion

- We use parallel NNs to solve the path planning problem in a classification manner. Our proposed method shows significant improvement from the state-of-the-art methods.
- Future Work: Enabling the communication of path information and optimizing the CL-aware path planner to remove some redundant paths.