Reinforcement Learning based Multi-Robot Classification via Scalable Communication Structure

Guangyi Liu, Arash Amini, Martin Takáč, Héctor Muñoz-Avila, and Nader Motee

Abstract—In the multi-robot collaboration domain, training with Reinforcement Learning (RL) can become intractable, and performance starts to deteriorate drastically as the number of robots increases. In this work, we proposed a distributed multi-robot learning architecture with a scalable communication structure capable of learning a robust communication policy for time-varying communication topology. We construct the communication structure with Long-Short Term Memory (LSTM) cells and star graphs, in which the computational complexity of the proposed learning algorithm scales linearly with the number of robots and suitable for application with a large number of robots. The proposed methodology is validated with a map classification problem in the simulated environment. It is shown that the proposed architecture achieves a comparable classification accuracy with the centralized methods, maintains high performance with various numbers of robots without additional training cost, and robust to hacking and loss of the robots in the network.

I. INTRODUCTION AND RELATED WORK

Real-time perception and classification with a multi-robot system have been among the outstanding research areas in robotics for the past decades. The significant challenges lie in developing a multi-robot learning and communication structure with guaranteed scalability, high accuracy, and low computational complexities. Recent advances in machine learning and distributed control systems have motivated and opened up new opportunities to revisit some of the classic examples of perception, such as map classification, in the context of networked robotics. In most real-world applications, robots can only sense their operating environment locally due to sensor capabilities and physical constraints. Although robots may communicate and exchange information about their observation with each other, the questions are: what relevant information should they exchange? Should every robot communicate with all other robots or only with its own nearest neighbors? Can robots possibly learn how to communicate with other robots? Finding answers to these and other similar questions will significantly facilitate tackling perception-based problems over a network of robots. As it is common in almost all networked systems, achieving scalability in design will become the key to having a successful story.

In this work, a CNN and LSTM based learning model is applied to robots with partial observations for classification. We propose a scalable communication structure that is constructed with parallel LSTM cells. In this structure, robots learning to communicate with only a finite number of neighbors. This will result in the communication network formed by star graphs and enable scalability to the network’s number of robots. A schematic example is shown in Fig. 1.

Robot’s operation involved with learning to classify an image and accomplish a real-time task has been widely studied by [14, 21]. Especially in the case that robot reacts to the environmental change captured by visual input in a real-time feedback loop [4]. In the real world scenarios, such functionality and performance of robot must be preserved when the imagery inputs are unperturbed or only partially available [23]. Current state-of-the-art approach [7] has widely studied to perceive the underlying state with limited or partial observation by implementation with Reinforcement Learning.

We propose a novel approach for this problem using a multi-robot system that provides observations from different target perspectives, accelerating and improving such classification tasks’ accuracy. Learning the task or skill with a multi-robot system has been proposed by [2, 19]. Communication is a principle feature of the multi-robot system, and learning to communicate effectively by using deep multi-robot reinforcement learning was studied by [5]. In the case when both communication and information input is constrained, which lands in the area of federated learning, robots can still learn a useful model with the approaches proposed by [9, 11, 22], which is essential to our problem since communication is not always available in the real-world scenario.
The CNN-LSTM model has been widely used and studied in recent works. [25] used a combination of regional CNN-LSTM model which divide the text input into several regions and use their weighted contribution for valence-arousal (VA) prediction. [26] uses both 1D and 2D CNN-LSTM model for speech emotion recognition such that the model could learn both local and global emotion-related features. Recent works [3, 24, 12] have used the CNN-LSTM models for the image space related propose with promising results. In our work, the CNN and LSTM based model will be constructed among a group of robots and with a time-varying partial input instead of the stationary and complete inputs.

II. Problem Formulation

The problem is to classify the map of a region, as an image, within a given finite time horizon using a team of robots with localized visual sensing capabilities and a scalable communication network. Applications include identifying search and rescue regions after natural disasters (e.g., earthquake or tsunami), enhancing existing maps of a region in a city for traffic and crowd control, and constructing soil map in geoscience during or after flooding to prevent and control potential subsequent disasters in a short time [1].

In such applications, the classification problem should be solved fast enough using a network of robots that can only observe the environment locally.

In this work, robots can only collect local visual observations as a sequence of images from the environment. If robots are within a certain distance from each other, they may establish a communication link between them and exchange their relevant information. Before deploying robots in an environment, they are trained for a given training set of maps. Our objective is to propose a decentralized learning architecture for the classification problem with a scalable communication network. We assume there are \( N \) robots in the network. The time index set is \( \mathbb{T} = \{1, 2, \ldots, T\} \) and \( t \in \mathbb{T} \). The \( i \)'th robot’s observation at time \( t \) is denoted by \( O(i, t) \). We denote the spatial position of robot \( i \) at time \( t \) as \( l(i, t) \in \mathbb{R}^2 \).

III. Learning Architecture

Learning map classification task in the real world by a team of robots requires an architecture that is sample efficient, scalable with the number of robots, fast, highly accurate, and robust. Deep reinforcement learning is a promising tool for the acquisition of complex skills based on existing partial observations to perform the following tasks: extract relevant features, encode and memorize the history of features, communicate with a bounded number of neighbors, plan goals and motions, and classify maps. Fig. 2 illustrates the main components of our proposed architecture.

A. Feature Extraction from Environment

All robots are equipped with identical pre-trained image processors. The incoming visual observation at time \( t \) is fed in parallel to a pre-trained VGG-19 model which is shown by,

\[
v(i, t) = V(O(i, t)),
\]

in which \( v(i, t) \in \mathbb{R}^a \) contains relevant information for classification purposes only. A real constant \( a \) is defined as the size of classification features and \( \theta_1 \) is trainable parameter vector for VGG-19.

Robots are also equipped with a CNN based goal-processor as proposed in [16]. This unit generates relevant features to identify the most informative location on the map so that the robot can be steered towards that location in the next steps. The goal feature vector is shown as,

\[
u(i, t) = G_1(O(i, t), l(i, t)),
\]

in which \( u(i, t) \in \mathbb{R}^b \) contains relevant information for goal-based motion planning. A real constant \( b \) is defined as the size of classification features and \( \theta_2 \) is trainable parameter vector for \( G_1 \).
Remark III.1. The value of the real constant $a$ and $b$ are comparably smaller than $p^2$, the size of raw incoming observations. The outputs of these two processors are concatenated and shown by $x(i, t) = [y(i, t)^T, u(i, t)]^T$.

B. Encoding Feature History with LSTM

Solving the classification problem in real-time becomes challenging as robots can only observe the underlying environment partially and must take actions to explore new regions that will possibly contain some useful information. Keeping track of what features have been observed by a robot in the past can significantly improve efficiency of planning, exploration, and classification.

Rather than requiring robots to exchange the feature vector $x(i, t)$ directly and having them store and fuse their received vectors, robots first encode and memorize history of their own features and then share this encoded information with their neighbors. We propose a pre-trained Long-Short Term Memory (LSTM) [6] cell to efficiently store the history of features in vector $m(i, t) \in \mathbb{R}^c$. We emphasize that all robots are equipped with identical LSTM cells, which drastically reduces training computational complexity. The dynamics of the LSTM cell is governed by

$$\begin{align*}
[m(i, t)^T, w(i, t)^T]^T &= F(m(i, t - 1), w(i, t - 1), x(i, t)),
\end{align*}$$

where $m(i, t) \in \mathbb{R}^c$ contains history of the past features observed by robot $i$ up to time $t$, $w(i, t)$ is the cell state of LSTM, and $\theta_3$ is a trainable parameter vector for the LSTM cell.

C. Scalable Communication Structure

The network of robots is allowed to communicate with its neighbors during the task. The communication topology is represented by a time-varying undirected graph $G(t) = (V, E(t))$, in which $V$ is the set of vertices (robots) with $V = \{1, 2, ..., N\}$.

Remark III.2. If the spatial distance between robots $i$ and $j$ satisfies $\|l(i, t) - l(j, t)\|_2 \leq e$ at time $t$, they may establish a communication link. The positive constant $e \in \mathbb{R}$ denotes the communication range.

In this manner, $E(t)$ denotes the set of all communication links at time $t$, i.e., $E(t) \subseteq \{i, j\} \mid i, j \in V$ and $\|l(i, t) - l(j, t)\|_2 \leq e$. The neighborhood of robot $i$ in $G(t)$ is denoted by $N_G(i)(t)$. At every time step, the number of possible communication graphs grows exponentially with the number of robots. For a team of robots to solve the classification problem, one needs to train them, as a team, with a relatively large number of possible communication graphs to ensure the convergence on various possible communication topologies [17]. This will naturally increase the computational complexity of training exponentially and makes the training process intractable since robots may encounter almost infinite possible communication networks [20] [15]. To address this challenge,

we propose a scalable communication structure that allows robots to learn how to communicate with only a bounded number of neighboring robots.

Let us consider a star graph $S_k$, which has $k + 1$ vertices (robots) and $k$ edges, which is shown in Fig. 3 (b). By taking into account constraints of battery life, we suppose each robot is capable of processing incoming information from only $\delta > 0$ neighbors in the range. This implies the graph neighborhood of robot $i$ is a star graph $S_k$ with $k \leq \delta$, and

$$\sup_{t \in \mathbb{N}} \Delta(G(t)) \leq \delta,$$

in which $\Delta(G(t))$ denotes the maximum degree of graph $G(t)$. This would form the communication network as a $(\delta/2, 0)$-sparse graph [10], an example is shown in Fig. 3 (a).

Remark III.3. With this structure, training a team of robots $w.r.t$ all possible communication graphs boils down to training with up to $\delta$ neighbors, i.e., the communication graphs for the $i$'th robot reduces to $S_1$,...,$S_{\delta}$.

In the next step, we design a learning mechanism for robots to acquire desired communication skills with the following key properties. First, robots will learn how to communicate and fuse messages received from up to $\delta$ neighbors with their own past history. This implies robots are equipped with a message processor to handle all possible localized communication topologies $\{S_1, ..., S_{\delta}\}$. Second, parallel implementation of the learning mechanism using LSTM cells make the message processing permutation invariant, i.e., the output of the message processor $(\hat{m}(i, t) \in \mathbb{R}^c)$ does not depend on the labeling of robots.

The scalable communication with star topology $S_k$ requires $k$ parallel identical LSTM cells $H_k$ and a fully connected linear layer $H_k$, we refer to Fig. 3 (c). All robots run Algorithm 1 synchronously as the communication structure:

- (Line 1-2) Robot send its message to all robots within $e$ distance and receive up to $\delta$ messages from neighbors at each time step. If robot $i$ receives $k$ messages, then

![Fig. 3: (a) Complete graph versus sparse graph; (b) Topology of star graphs $S_1$ and $S_\delta$; (c) Scalable Communication Structure constructed with $S_k$.](image)
After performing necessary computations, robot separates
contains updated features for classification and goal-based mo-
Remark III.4. The output $H$ planner. We denote the trainable parameters for
is particularly suitable to classify maps over large spatial
needed.
robots are contained in the network, robots can communicate
in the star graph $S_k$ is fed into $H_k$ and generates $\hat{m}(i,t)$
as output.
Remark III.4. The output $\hat{m}(i,t) = [\hat{\psi}(i,t)^T, \hat{\phi}(i,t)^T]^T$ contains updated features for classification and goal-based motion planning.
After performing necessary computations, robot separates
Robots take localized observations, process it
The computational complexity of the proposed communication
In summary, our proposed communication structure and
D. Goal-based Motion Planning and Map Classification
In order to close the loop for motion planning, the goal-
based motion planner $G_1$ used. A goal position $g(i,t) \in \mathbb{R}^2$ denotes the most informative region, it will be sampled as,
\begin{equation}
g(i,t) = G_2(\hat{u}(i,t)), \quad \text{(5)}
\end{equation}
in which $\theta_5$ denotes the trainable parameter for goal-based motion planner. In order to encourage the robot to follow the
goal and collect more informative observations, $g_{i,t}$ is kept
fixed over time interval $[t,t+T_g]$.
At the same time, robot $i$ attempts to classify the target
environment. The classifier consists of a fully connected
layers with bias and followed by a SoftMax function. The classifier utilizes the updated feature history vector $\psi(i,t)$ as its
input. A prediction vector will be generated as,
\begin{equation}
q(i,t) = C(\psi(i,t)), \quad \text{(6)}
\end{equation}
in which the trainable parameter is defined as $\theta_6$. The true label for $k$'th class $Q_k$ is defined as the $k$'th Euclidean basis in $\mathbb{R}^M$, in which $M$ denotes the total number of labels. The reward for classifying $k$'th label on robot $i$ is evaluated by a log-sum-exp (LSE) loss $[13]$ as following,
\begin{equation}
r(i,T) = -\text{LSE}(Q_k, q(i,T)). \quad \text{(7)}
\end{equation}

IV. Training Preliminaries
We denote all trainable parameters in the proposed architectural as $\Theta = \{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6\}$. The estimated cost function for robot $i$ is defined as $J_i(\Theta) = r(i,T)$. Implementation with RL refers to solve the optimization problem as maximizing $J_i(\Theta)$ subject to constructing $G(t)$ with star graph $S(i,t)$, finite number of robots $N$ and finite time horizon $T$. In order to ensure the optimal performance among all robots, we replace $J_i(\Theta)$ with a global average $J(\Theta) = (\sum_{i=1}^N r(i,T))/N$.

The training procedure is shown in Algorithm 2. Trainable parameters $\Theta$ will be arbitrarily initialized at the beginning of each epoch. For each training epoch, when $t = 0$, we arbitrary reset the location of robots $l(i,0)$, feature history $m(i,0)$ and cell state $w(i,0)$.

- (Line 5-6) Robots take localized observations, process it with image and goal processors and encode the features with the LSTM unit.
- (Line 10-12) Once communication and goal-based motion planning is available, robots fuse their memory with its neighbors and samples the goal location.
- (Line 17-18) Robots classify the target environment and generate the reward for updating parameters.

The training is separated into 3 stages, each stage will terminate when further does not show any improvement. In the first stage, communication and goal-based motion planning are disabled, robots will solely learn classification and memory encoding with random actions. Parameters $\{\theta_2, \theta_3, \theta_4, \theta_5\}$ are detached from the gradient. In the second stage, goal-based planning is enabled while communication is still disabled. Robots will learn to explore the most informative region. Parameters $\{\theta_1, \theta_3, \theta_4, \theta_5, \theta_6\}$ are detached from the gradient. In the last stage, communication is enabled, robots learn to update their state vectors with the scalable communication structure. Parameters $\{\theta_1, \theta_2, \theta_3, \theta_5, \theta_6\}$ are detached from the gradient.

Remark IV.1. In the first and second stage, robot will use it’s own state vector instead when communication is not available (e.g. line 10); In the first stage, a random action is sampled when goal location is not available (e.g. line 11).

Algorithm 1: Scalable Communication Structure

Data: $m(i,t), w(i,t)$, and $m(j,t)$ from neighbors
Result: $\hat{m}(i,t)$
1 Initialize $N_{G(t)}(i)$ with $e$ and compute $k \leq \delta$;
2 Categorize $N_{G(t)}(i) = S_k, H_k$, and $\hat{H}_k$;
3 while $j \in N_{G(t)}(i)$ do
4 $\hat{m}(j,t) \leftarrow H_k(m_{i,t}, w_{i,t}, m(j,t))$;
5 end
6 $\hat{m}(i,t) \leftarrow [\hat{m}(1,t)^T, \ldots, \hat{m}(k,t)^T]^T$;
7 $\hat{m}(j,t) \leftarrow \hat{H}_k(\hat{m}(i,t))$;

$N_{G(t)}(i) = S_k$. $H_k$ and $\hat{H}_k$ will be selected as its communication structure.
- (Line 3-5) Each parallel LSTM cell $H_k$ processes incoming message $m(j,t)$ from the $j$'th neighbor with robot's own states $m(i,t)$ and $w(i,t)$.
- (Line 6-7) The collection of $\hat{m}(j,t)$ from $k$ neighbors in the star graph $S_k$ is fed into $\hat{H}_k$ and generates $\hat{m}(i,t)$ as output.
Algorithm 2: Multi-Robot Classification Training

Procedure

Data: Trainable parameters $\Theta$ and learning rate $l_r$
Result: Updated parameters $\Theta$

for each training epoch do
  Initialize $l(i,0)$, $O(i,0)$, $m(i,0)$ and $w(i,0)$;
  while $t \in \mathbb{T}$ do
    while $i \in \mathcal{V}$ do
      $x(i,t) \leftarrow [V(O(i,t)^T, G_1(O(i,t), l(i,t))^T];$
      $[m(i,t)^T, w(i,t)^T] \leftarrow F(m(i,t-1), w(i,t-1), x(i,t));$
      $i \leftarrow i + 1;$
    end
    while $i \in \mathcal{V}$ do
      $[\hat{v}(i,t)^T, \hat{u}(i,t)^T]^T \leftarrow \text{Algorithm 1}$
      $g(i,t) \leftarrow G_2(\hat{u}(i,t));$
      update $l(i,t)$ w.r.t. goal location;
      $i \leftarrow i + 1;$
    end
    $t \leftarrow t + 1;$
  end
  $q(i,T) \leftarrow C(\hat{v}(i,T));$
  $r(i,T) \leftarrow -\text{LSE}(Q_k, q(i,T));$
  if train w/ random action then
    $(\theta_1, \theta_3, \theta_6) \leftarrow (\theta_1, \theta_3, \theta_6) - l_r \nabla_{\theta_1, \theta_3, \theta_6} J;$
  else if train w/ goal-based action then
    $(\theta_2, \theta_5) \leftarrow (\theta_2, \theta_5) - l_r \nabla_{\theta_2, \theta_5} J;$
  else
    $(\theta_4, \theta_4) \leftarrow (\theta_4, \theta_4) - l_r \nabla_{\theta_4, \theta_4} J;
  end
end

V. CASE STUDIES

A. Satellite Map Dataset

A map dataset is created by using satellite maps exported from Google Earth of 10 university campuses over past 40 years. In Fig. 4(a) and 4(b), changes of landmarks and features are shown in both long and short time intervals. This dataset would require robots to filter out minor changes and focus only on the major features of the map.

In order to simulate the real-world environment, we added 80 random generated clouds on each samples of the original dataset, such that each cloud will cover 40% of the map (Fig. 4(c)). Such manipulation will increase the difficulty of the classification for 1) the total available area of the map is constrained; 2) different maps share more similar features by covered with same cloud. The clouded map dataset consists of 20,000 training maps and 2,000 unseen testing maps, the dataset can be found in the supplementary materials. The size of the map is 1024x768 pixels and the aerial robot is allowed to take square partial observations with a frame size of $p^2 = 64 \times 64$ pixels (relative observation size 0.52%). Range of communication $\epsilon$ is set as 480 pixels to encourage robots using various star graph when training with communication.

B. Simulation on MNIST and Map Dataset

We use ADAM [8] with a learning rate $l_r = 0.0001$ to train the model in PyTorch [18]. The testing accuracy is presented as a global average among all robots and 5 random seeds. Fig. 5 shows the learning curve of map dataset with clouds for $N = 5$ and $T = 15$. The curve represents 3 training stage which starts at 0, 100 and 200 epochs. The drop in orange curve around 200 epoch indicates the learning for communication module. Table. 1 also implies the proposed method achieves a comparable performance w.r.t. the model that utilizes the full image as input. Snapshots of experiments with $N = 5$ and $T = 12$ is shown in Fig. 4. We refer to the supplemental video for a detailed illustration and real-world experiments with UAV.

C. Scalability with Sparse Network

We validate the scalability of sparse communication network by using both method trained for $N = 5$ and $T = 15$. The result of map dataset with clouds is shown in Table. 1. In the case using sparse communication network (Sparse-<5>), robots preserve performance when new robots join the network without additional training. However, performance will decrease for robots trained with complete communication network (Complete-<5>). In addition, we test the scalability for complete communication network trained with $N = 80$ and $T = 15$ (Complete-<80>) by reducing number of robots in the network.

Our method also reduces computational cost for training with a large number of robots. We evaluate the extra training time consumption for complete communication network to reach the same level of Sparse-<5>. We evaluated the additional time used for training a model from Complete-<5>, which is faster than training from scratch. The result
Fig. 6: Snapshots of experiments with $N = 5$ and $T = 12$. Stars denote goal locations for each robot and dashed lines stand for the communication links.

| Dataset   | Observation size (%) | 1 robot | 5 robots | 10 robots | VGG-19 w/ full image |
|-----------|----------------------|---------|----------|-----------|---------------------|
| MNIST     |                      | 91.27   | 94.98    | 98.31     | 99.33               |
| Map w/o clouds | 0.52               | 85.19   | 98.94    | 99.71     | 99.84               |
| Map w/ clouds | 2.08              | 88.59   | 99.27    | 99.84     | 99.84               |

**TABLE I:** Average classification accuracy (%) over MNIST and map dataset.

| Method & Number of robots | 5   | 10  | 20  | 40  | 80  |
|---------------------------|-----|-----|-----|-----|-----|
| Sparse-<5>                | 97.30 | 97.56 | 96.90 | 97.20 | 96.72 |
| Complete-<5>              | 95.60 | 93.21 | 86.55 | 85.46 | 63.36 |
| Complete-<80>             | 73.54 | 76.81 | 78.55 | 90.59 | 98.24 |

**TABLE II:** Average classification accuracy (%) with various number of robots.

is shown in Table. This indicates training with complete communication network would take a significant amount of time compared to Sparse-<5> to reach the same performance level.

**D. Network Efficiency and Robustness**

We also evaluate time used for various number of robots to reach a classification accuracy threshold of 97% in Fig. (a). The network of robots shows a relatively intuitive result that to achieve a certain accuracy the time cost $T$ is inversely proportional to the number of robots $N$. Since all robots are identical, the total power consumption $T \times N$ for various number of robots in the network is also shown in Fig. (a). This result indicates the total energy cost for a network of robots would be dramatically reduced as the number of robots increases.

The sparsely connected network also grants the robustness to the architecture, since robots can reform to a new communication network $G(t)$ with a previously learned star graph $S_k$. We validate the robustness by letting 20 robots trained with Sparse-<5> to classify the map with $T = 10$. We randomly remove some robots from network during the task, and evaluate the testing results of remaining robots. The result is shown in Fig. (b) as the proposed scalable communication structure preserve a good performance until losing 80% of the robots.

| Number of Robots | 5   | 10  | 20  | 40  | 80  |
|------------------|-----|-----|-----|-----|-----|
| Additional training time (min) | $\infty$ | 320 | 1132 | 1248 | 2945 |

**TABLE III:** Additional training time used for complete communication network to reach the same performance w.r.t sparse communication network, tested on a NVIDIA Tesla K80 GPU.

VI. Conclusion

We present a learning and scalable communication structure that allows a large network of robots with localized sensing capabilities to classify the target environment. We demonstrate the usefulness of the proposed methodology by using image and map classification as examples in simulation and real-world experiments. The learning model utilizes CNN and LSTM cells for feature extraction and memory encoding, while the star graph-based sparse topologies are used to construct the scalable communication structure. It is shown that using a multi-robot system with our proposed architecture could achieve a comparable level of performance w.r.t. the conventional centralized methods. Besides, the multi-robot system reduces energy costs significantly since it only covers the most informative regions. Using a sparse communication topology has practical implications, including minimizing the number of message transmissions over the network, improving the energy usage of robots, and enhancing network privacy and security due to short-range communication. The proposed architecture’s scalability will save a significant amount of training time and energy for multi-robot systems, especially with a large number of robots.
