Privileged Reinforcement and Communication Learning for Distributed, Bandwidth-limited Multi-robot Exploration

Yixiao Ma[1,2], Jingsong Liang[1,2], Yuhong Cao[2], Derek Ming Siang Tan[2], and Guillaume Sartoretti[2]

[1] School of Computing, National University of Singapore, SG 117417.
yixiaoma@u.nus.edu, jingsongliang@u.nus.edu,
[2] Mechanical Engineering Dept., National University of Singapore, SG 117575.
caoyuhong@nus.edu.sg, derektan@u.nus.edu, mpegas@nus.edu.sg,
WWW home page: http://www.marmotlab.org

Abstract. Communication bandwidth is an important consideration in multi-robot exploration, where information exchange among robots is critical. While existing methods typically aim to reduce communication throughput, they either require significant computation or significantly compromise exploration efficiency. In this work, we propose a deep reinforcement learning framework based on communication and privileged reinforcement learning to achieve a significant reduction in bandwidth consumption, while minimally sacrificing exploration efficiency. Specifically, our approach allows robots to learn to embed the most salient information from their individual belief (partial map) over the environment into fixed-sized messages. Robots then reason about their own belief as well as received messages to distributedly explore the environment while avoiding redundant work. In doing so, we employ privileged learning and learned attention mechanisms to endow the critic (i.e., teacher) network with ground truth map knowledge to effectively guide the policy (i.e., student) network during training. Compared to relevant baselines, our model allows the team to reduce communication by up to two orders of magnitude, while only sacrificing a marginal 2.4% in total travel distance, paving the way for efficient, distributed multi-robot exploration in bandwidth-limited scenarios. We open-sourced our full code [3].

Keywords: Deep Reinforcement Learning, Communication Learning, Multi-robot exploration, Distributed Path Planning

1 Introduction

Information sharing in multi-robot exploration is critical to generating high-quality, distributed exploration paths [25]: robots must communicate with each other to obtain more information beyond their own partial knowledge, to make cooperative decisions that can speed up task completion and avoid redundant
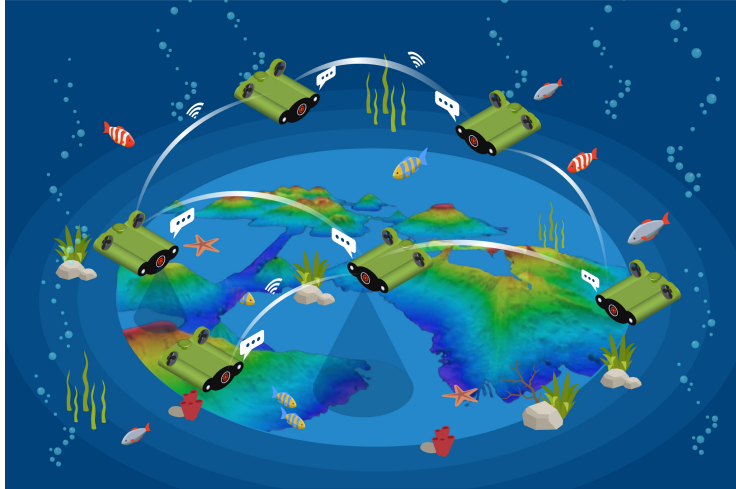
---

[3] https://github.com/marmotlab/Bandwidth-Limited-Multi-Robot-Exploration

Fig. 1. Example application of our approach to a multi-robot exploration task in a bandwidth-constrained environment (here, underwater).

work. Under normal circumstances, it is usually viable for most exploration planners to share vast amounts of important information among team members (such as full partial maps, or robot trajectories) continuously during the mission. However, these approaches become impractical in bandwidth-constrained settings such as underwater and underground environments [11,22], where communication range is often inversely linked to communication bandwidth. Recent research shows that the transmission of an occupancy grid map usually requires a bandwidth of ~ 2 Mbps [23], but that underwater communication bandwidth rarely exceeds 100kbps [11]. This problem of bandwidth constraints is further exacerbated in larger teams, where (even modern) communication channels may not be able to withstand high communication throughput between numerous robots [14].

There are currently two main strategies to reduce communication throughput among robots. The first strategy is to simply decrease the frequency of communications [17]. However, this strategy often leads to poor cooperation, mainly due to the lack of up-to-date information from other robots. The second strategy is to reduce the size of messages, e.g., by only sharing low-dimensional representations of each others' partial map, which recipients can then process to reconstruct/estimate the full map [10,30]. However, these approaches usually come at important computational costs and may result in lower exploration efficiency when essential details from the original map are lost during map exchange.

To address these problems, we propose a novel DRL-based multi-robot exploration framework based on communication learning and privilege learning, tailored for bandwidth-limited scenarios. Our framework relies on learned messages as an alternative to conventional map sharing. Our approach primarily relies on communication learning to allow robots to learn to encode their own belief map into a small, fixed-sized message that is shared with other robots. In doing so, our communication layer allows robots to learn to identify, encode, and

share the most salient portion of their individual belief with each other, within given constraints over message length (i.e., maximum bandwidth within the system). Robots then learn to reason about their own knowledge/state as well as receive messages to form an implicit representation of the overall explored environment. This enables the generation of high-quality, distributed exploration paths. Following our recent work in single robot exploration [5], we rely on privileged learning to boost the performance of our final model. Specifically, we let our critic network access ground truth information during training only, allowing it to provide more accurate action evaluation for the training of the robots' policy network. This training approach significantly enhances our final model's long-term planning capabilities, by allowing robots to reason about their knowledge, as well as received messages, at different spatial and temporal scales.

We compare our model to a conventional multi-robot exploration planner in a set of 100m×100m indoor maps and investigate the impact of using learned messages over traditional partial map sharing. Our results show that our model can reduce the volume of communications by up to **99.2%**, at the cost of a marginal **2.4%** performance loss. These results highlight the capability of our robots to understand and reason about the current global state of the exploration task, without explicitly relying on other robots' detailed map. We finally train a variant of our DRL-based model, where robots are allowed to both share learned messages and explicit partial maps. This model outperforms our map-sharing-free approach and the conventional baseline by 11.4% and 9.2% respectively in terms of exploration distance, highlighting the power of our general framework in high-bandwidth scenarios where full maps may be reliably shared among robots.

2   Prior Works

2.1   Multi-robot Exploration

Approaches to conventional multi-robot exploration are methodologically classified into two main categories: frontier-based and sampling-based. For example, Yu et al. [29] applied artificial potential fields to attract robots towards diverse frontiers while ensuring mutual repulsion to maintain distance between each other. Most recently, Cao et al. [1] proposed an mTSP (Multiple Traveling Salesman Problem) based global planner in conjunction with a sampling-based local planner to explore large-scale environments. The centralized global planner segments exploration areas into several vital nodes and then assigns them to robots. However, these approaches remain greedy and prioritize short-term efficiency, often resulting in shortsighted path planning and suboptimal performances.

Given the rapid advancement of neural networks, many works have looked to deep learning to enhance autonomous exploration. Niroui et al. [19] first pioneered the integration of frontier-based methods with deep reinforcement learning to resolve short-sightedness problems in single-robot exploration. Yu et al. [28] employed asynchronous multi-robot proximal policy optimization as a training method to address multi-robot exploration. Concurrently, Luo et

al. [16] leveraged graph convolutional neural networks to enhance the efficiency of multi-robot exploration. Our most recent work proposed ARiADNE [3,5], an attention-based neural network to achieve state-of-the-art in long-term planning for single-robot exploration. In this work, we extend the ARiADNE framework to multi-robot exploration under bandwidth limitations.

2.2   Communication in Multi-Robot Exploration

Many existing methods [16,28,29] adopt continuous communication to gather up-to-date information that assists robots in making cooperative decisions. However, this often comes at the cost of high communication bandwidth requirements. Some methods reduce communication throughput by decreasing the frequency of communications. For example, Masaba et al [17] proposed a method to utilize a generalized Voronoi graph to segment the environment into sub-regions and allocate robots into distinct regions to explore. Robots only communicate with each others when their own region of responsibility has been fully explored, at which point a new region will be allocated to them. However, this method represents a trade-off between exploration efficiency and communication frequency, because the lack of real-time information from other robots often results in poor cooperation. Alternatively, some methods chose to reduce the size of messages. For instance, Gaussian Mixture Model (GMM) Maps [10] involves exchanging key map parameters and then attempting to reconstruct the map. Nevertheless, these approaches require substantial computational resources, hindering their practical implementation for exploration tasks that demand real-time decision-making. Another example [21,30] extracts semantic information from the map to create a compact topological map that is shared with other robots. However, this approach often results in lower exploration efficiency due to the loss of some details from the original map in the topological representation.

2.3   Communication Learning

Foerster et al. [7] introduced the concept of communication learning in multi-robot reinforcement learning: robots learn what information/message to share within the team, thereby effectively augmenting the knowledge available to individual robots and mitigating the challenges posed by partial observability. Furthermore, Freed et al. [8] introduced a differentiable discrete binary channel for communication learning and demonstrated that this approach surpasses the robustness and performance of the reinforced communication learning [7] proposed by Foerster et al. The key component in communication learning is message generation. Messages can be generated through two methods: by inputting each robot's observation into a message generation module [7,12,13] or by sequentially processing a message that is circulated among the robots [20,26]. Specifically, Kim et al. [13] introduced a novel scheme of intention-sharing within communication learning by swapping intentions that are encoded from partial observations. On the other hand, FCMNet[26] and FCMTran [26] let robots communicate with each other by enabling sequential message processing among robots,

achieving state-of-the-art status in StarCraft II Micromanagement benchmarks. These works demonstrate that communication learning can enable robots to reconstruct and utilize information from high-dimensional compressed messages.

3   Problem Formulation

In this work, we consider multi-robot exploration under constraints over communication bandwidth. We let W be a bounded environment, which can be divided into free area W_f and occupied area W_o, where $W_f \cup W_o = W$ and $W_f \cap W_o = \varnothing$. The belief map M_i is locally maintained by the i-th robot about W, and is categorized into three components: free area M_i^f, occupied area M_i^o, and unknown area M_i^u. Initially, each robot starts at p_i^0 with an empty individual belief map, such that $M_i = M_i^u$. During the exploration task, each robot makes individual decisions and generates/follows a trajectory, i.e., a sequence of waypoints $\psi_i = \{p_i^0, p_i^1, ...\}$, each located in the free area W_f. Each robot updates its belief map according to regular measurements along its path, by (re-)classifying formerly-unknown area into either M_f or M_o.

The objective of multi-robot exploration can be formulated as a MinMax function, where we aim to minimize the maximum trajectory length among all robots (i.e., makespan of the task):

$$\min \max_{i \in 1,...,n} L(\psi_i), \tag{1}$$

where $L(\psi_i) = \sum_{t=1}^m D(p_i^{t-1}, p_i^t)$ is the trajectory length to complete exploration, with $D(p_i^{t-1}, p_i^t)$ the distance between p_i^{t-1} and p_i^t. We consider the multi-robot exploration completes when the exploration rate (i.e., the proportion of explored free area $|\bigcup_{i=1}^n M_i^f|$ compared to the true free area in the environment W_f) exceeds a given threshold θ.

In this work, we further consider a bandwidth constraint for robots' communication during exploration, preventing robots from sharing their local belief map M_i^t with each others (in practice, we impose a maximum of $\tilde{2}56$ bytes per message). Instead, robots can only exchanges compact messages, containing small amounts of information such as their current position. Note that, without accurate map sharing, robots cannot fully confirm whether an area in the environment has been explored by others or not. Therefore, in such a problem setup, it is more reasonable to set the exploration objective as efficiently exploring most of the environment rather than the whole environment. Consequently, we set the exploration terminating threshold θ between $[0.9, 1]$ in practice.

4   Distributed Bandwidth-Limited Exploration

4.1   Exploration As An RL Problem

We cast the exploration problem as a partially observable Markov Decision Process, formulated using the tuple $(S, A, \Gamma, R, \Omega, O)$, where S denotes the state
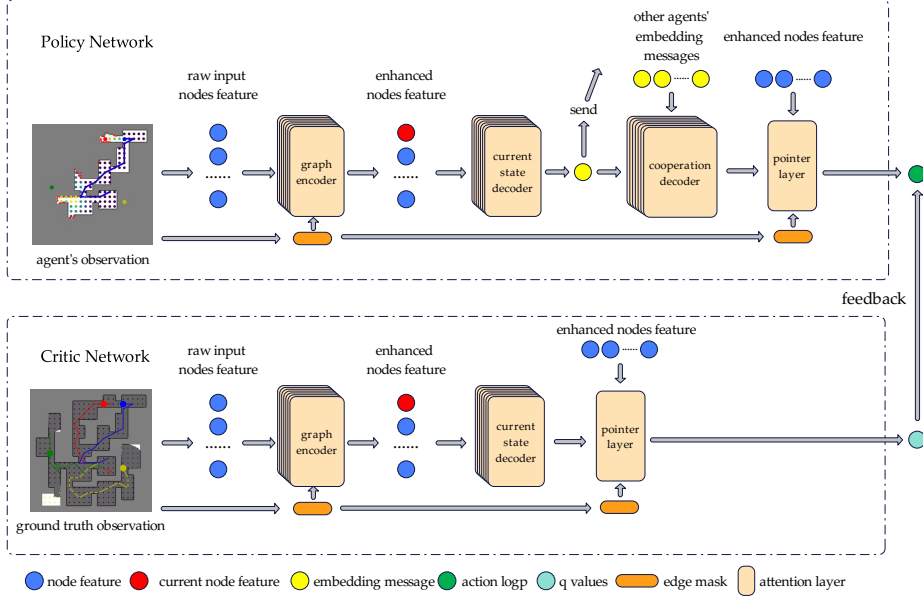
Fig. 2. Architecture of our policy and critic networks.

space, A the action space, Γ the state transition function, R the reward set, Ω the observation space of the robot, and O the observation function. During exploration, the robot does not have access to the true state $s^t, \forall s^t \in S$; instead, it operates based on its observation $o^t, \forall o^t \in \Omega$. According to its observation, each robot selects and executes action $a^t \sim \pi(\cdot \mid o^t)$, where it selects and travels to the next way-point p^t. The next state s^{t+1} is determined by the current state s^t and the action a^t, through the state transition function $\Gamma(s^{t+1} \mid s^t, a^t)$ (unknown to the). Each robot$_i$'s objective is to learn an optimal policy $\pi^*(a_i^t \mid o_i^t)$ that maximizes the long-term expected return $\mathbb{E}\left[\sum_{t=1}^{m} r^t\right]$, where $r^t \in R$. In the following sections, we detail the observation, action space, and reward structure of our RL formulation.

4.2   Policy Network

Policy Network Observation Similar to our prior works [3,5], the observation for each robot is an informative graph $G^t = (V^t, E_i^t)$ defined over the free explored space in each robot's partial map. E_i^t is defined as the collision-free edges connecting each vertex in V^t with its k nearest neighbors. Each vertex $v_i = (c_x^t, c_y^t, u^t, g^t, p^t) \in V^t$ has four components, namely coordinates, utility, guidepost, and position indicator, where coordinates (c_x^t, c_y^t) represent the 2D position of the graph vertex, utility u^t measures the number of observable frontiers from that vertex, and the guidepost g^t component represents whether this vertex has been visited previously. Extending [3] to a multi-robot setting, we further augment the observation with an additional component called occupancy, which indicates whether each vertex is currently occupied by the current robot,

by another robot, or is unoccupied, with values -1, +1, and 0 respectively. This component enables each robot to understand the geometric relationships between their position and those of other robots, thereby improving their cooperative capabilities. To represent another robot located outside the area explored by the robot so far (i.e., outside the area covered by its graph G^t), we manually insert a temporary vertex v to the graph G^t.

Graph Encoder For each robot, we use a graph encoder to extract high-dimensional and informative node features from its informative graph. The graph encoder contains multiple consecutive attention layers [24], each learning to intelligently fuse neighboring node features. We first project every node from V^t in the informative graph G^t into a high-dimensional (dimension $d = 64$ in practice) node feature. We denote those raw node features as $F^{raw} \in \mathbb{R}^{d \times N}$, where N is the number of nodes in the graph. Thereafter, we input F^{raw} and the edge mask $\hat{M}$ into attention layers. The edge mask $\hat{M} \in \mathbb{R}^{N \times N}$ is calculated from E_i^t, where $\hat{M}_{ij} = \begin{cases} 0, (v_i, v_j) \in E_i^t \\ 1, (v_i, v_j) \notin E_i^t. \end{cases}$. It is applied in the attention layers to only allow the merging of neighboring node features. Each attention layer reads:

$$
\begin{aligned}
q_i &= W^Q h_i^q, k_i = W^K h_i^{k,v}, \\
v_i &= W^V h_i^{k,v}, u_{ij} = \frac{q_i^T \cdot k_j}{\sqrt{d}}, \\
w_{ij} &= \begin{cases} \frac{e^{u_{ij}}}{\sum_{j=1}^n e^{u_{ij}}}, \hat{M}_{ij} = 0 \\ 0, \hat{M}_{ij} = 1 \end{cases}, h_i' = \sum_{j=1}^n w_{ij} v_j,
\end{aligned}
\tag{2}
$$

where $W^Q, W^K, W^V \in \mathbb{R}^{d \times d}$ denote learnable weight matrices, h^q is the query source, and $h^{k,v}$ is the key-and-value source. Note that in the graph encoder, $h^q = h^{k,v}$ (known as self-attention). For the first attention layer, $h^q = h^{k,v} = F^{raw}$. The following layers sequentially take the previous layer's output as input. We obtain the output of the final attention layer as enhanced node features $\hat{F} \in \mathbb{R}^{d \times N}$.

Current State Decoder The current state decoder distills the relationships between the current robot position and all other graph vertices. We first input the enhanced node features of the current node $\hat{F}_c$ as the query source and all enhanced node features $\hat{F}$ as the key and value source into an attention layer. Thereafter, we concatenate the output features together with $\hat{F}_c$ and re-project the output to a d-dimensional feature μ. Here, μ represents the knowledge about current accessible state (not global state, because our robot's observation is partial), capturing multi-scale dependencies among previously traversed areas. Finally, μ is broadcast to all other robots as this robot's message for this decision step, as we discussed in section 3.

Cooperation Decoder The cooperation decoder aggregates relationships between an agent's own learned message (output of the current state decoder) and
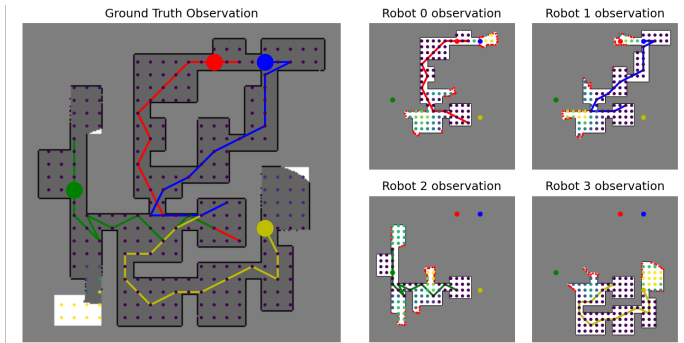
Fig. 3. Observation structure for our policy and critic networks. The left part is the ground truth observation used in our critic network, while the four plots on the right respectively represent the partial observations used as input to the robots' policy networks. The large dots in red, blue, yellow, and green represent the current position of the robots, while the smaller nodes' varying colors denote varying exploration utility. We omit the depiction of the collision-free graph edges for visualization purposes.

the received learned messages from other robots to form an implicit representation of the overall explored environment. We first take robot$_i$'s learned message μ_i^t as the query source of a cross-attention layer. Concurrently, we concatenate all other robots' learned messages, $\mu_{-i}^t = \mu_{1:n}^t \setminus \mu_i^t$, and pass it into the same attention layer as the key-and-value source. From here, we derive the cooperative feature $F^{co} \in \mathbb{R}^{d \times 1}$, which represents the implicit context of the overall explored environment reasoned from all robots' learned messages. Notably, the cooperation decoder can accept messages from an arbitrary number of robots, highlighting the scalability of our approach to different team sizes.

Action Space The action space comprises all neighboring vertices to the current vertex the robot is occupying. We pass the cooperative feature F^{co} and the neighboring nodes enhanced feature $\hat{F}_{nbr}$ into the pointer layer [3,4,15]. Its output takes the form of a logarithmic probability distribution, from which we sample an action during training, and select an action greedily during testing/deployment. Overall, the attention-based neural network outputs a stochastic policy described by $\pi(a_i^t \mid o_i^t) = p_i^{t+1}, (p_i^{t+1}, p_i^t) \in E_i^t$.

4.3   Training

We employ parameter sharing to train a homogeneous policy for all agents. Specifically, we rely on the soft actor-critic (SAC) algorithm [6,9], which trains a policy networks and double critic networks. Compared to standard SAC, following our previous work [5], we further propose a privileged critic network which significantly boosts the performance of the trained model.

Privileged critic network We employ privileged learning to endow the critic (i.e. teacher) model with ground truth map knowledge to effectively guide the

policy (i.e. student) model during training. It aims to increase the accuracy of the state-action values (the predicted long-term return) predicted by the critic network, which is then used to guide the policy network training. In particular, compared to our policy network discussed in the previous section, we modify the observation and network structure of the critic network. First, our critic network utilizes a ground truth graph $\hat{G}^t = (\hat{V}^t, \hat{E}^t)$ (derived from the ground truth state s^t) as the observation instead of the partial informative graph G^t (derived from belief map M_i^t), as illustrated in Figure 3. There, utilities of nodes in $\hat{V}^t$ are determined by the number of neighboring unexplored free area.

Second, we eliminate the communication layer from the critic network. As the model has access to the global ground truth already, messages shared by other agents are not helpful for return estimation. As seen from figure 4, the introduction of privileged map information significantly enhanced training performance by 23.8%. Moreover, we notice improved training stability as seen from lower variance compared to the standard SAC. This demonstrates that the critic network can better guide the policy network with more accurate long-term return estimation derived from privileged information.
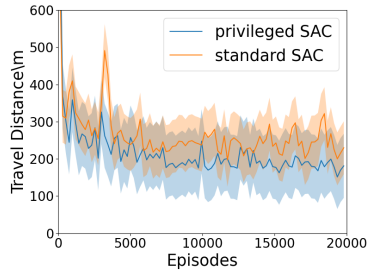

Fig. 4. The ablation of Privileged Learning. Both models have been trained for 20,000 episodes to converge. A lower travel distance signifies better model performance.

Reward Structure We modify our training reward structure based on our previous single robot exploration work [3] to encourage robots to collaboratively explore the environment. The original reward structure in [3] consists of three parts: (1) The first part r_o, is the is the number of observed frontiers at the new viewpoint. (2) The second part r_c is a punishment on the distance between the previous and new viewpoints. (3) The last part r_f is a finishing reward given after the environment has been fully explored. In our approach, we perform three modifications to this original reward function. First, we change r_o to the unexplored area around the new viewpoint using the ground truth map. This aims to provide a more accurate assessment of the exploration utility given the selected position. Second, we introduce a team reward ξ, which is related to the newly explored area discovered between the previous and current viewpoints (i.e., $\xi^t = \rho(\bigcup_{i=1}^n M_i^{t+1} \setminus \bigcup_{i=1}^n M_i^t)$, where ρ is a normalize factor). This reward is designed to encourage robots to explore cooperatively instead of greedily pursuing individual exploration reward r_o . Third, we introduce a momentum reward λ_i^t to encourage robots to move in any general direction with minimal backtracking. Its purpose is to shorten the highly random processes at the beginning of training and optimize robot's trajectory. The calculation of λ_i^t is detailed in algorithm 1. Thus, the overall reward at decision step t is

$$r^t(a^t, o^t) = r_o + \lambda_i^t + \xi^t + r_f - r_c, \tag{3}$$

Algorithm 1 momentum reward calculation

Require: current position $p^t = (x_1, y_1)$, next way-point $p^{t+1} = (x_2, y_2)$, previous step
movement direction $\vec{\alpha}$;

1: current step movement direction $\vec{\beta} \leftarrow p^t \cdot p^{t+1} = (x_1 x_2, y_1 y_2)$;
2: momentum reward $\lambda^t \leftarrow 0.1 \times (\vec{\alpha} \cdot \vec{\beta})$;
3: $\vec{\alpha} \leftarrow \vec{\beta}$;
4: **Return:** λ^t

Training Details Our model is trained on a set of 4000 dungeon maps like
figure 3, each $100m \times 100m$. The sensor range of our robots is 20 meters. Nodes
are uniformly distributed 4 meters apart in the free area. Each node connect to
its neighboring nodes up to 12 meters away (i.e., max 24 neighboring nodes).
During training, we set the number of robots $N = 4$, the max episode length is
set to 128 steps, the batch size to 256, the learning rate to $1e - 5$, and discount
factor $\gamma = 1$. We set the episode buffer size to 20000, and commence training
when at least 5000-step data have been collected. We utilize Ray [18] for parallel
data collection. The model requires around 20,000 episodes to converge, taking
around 108 hours on a desktop with an Intel i9-9900k CPU and two NVIDIA
RTX 2080 SUPER GPUs.

5   Experiments

5.1   Experimental Setup

We extend TARE [2]'s TSP-based local planner to multi-robot settings, and refer
to it as **mtsp_based**. Specifically, the resulting **mtsp_based** planner samples
the frontiers into viewpoints and connects them through solving a mTSP (in
practice we use ORtools). Note that this exploration planner is distinct from the
multi-robot planner used in MUI-TARE [27] and MTARE [1]: MUI-TARE [27]
specifically aims at enhancing the quality of map merging during the explo-
ration process, while MTARE [1] focuses on exploration under communication
constraints in large-scale environments. Therefore, we adopted *mtsp_based* as
our baseline, as a better alternative to fairly represent conventional algorithms.

In **mtsp_based**, each robot shares its own belief map with other robots,
forming a global belief map with positions $(\bigcup_{i=1}^{n} M_i^t, p_{1:n}^t)$. On the other hand,
for a fair comparison of the long-term exploration capabilities, we proposed a
variant of our model: *global_map*, which also has access to shared belief maps.
That is, its observation is $(\bigcup_{i=1}^{n} M_i^t, \psi_i^{1:t}, p_{1:n}^t, \mu_{1:n}^t)$ instead of the original ob-
servation $(M_i^t, \psi_i^{1:t}, p_{1:n}^t, \mu_{1:n}^t)$.

In our experiments, robots communicate through learned messages at every
time step t, where each message packet is a 1x64-dimensional float32 tensor (i.e.,
64×4 bytes). Our competing baselines communicate using belief map messages,
where each message packet is a $H \times W$ int8 tensor (i.e., $H^t \times W^t \times 1$ bytes), and
H and W is the height and width of the map message respectively.

Table 1. Comparisons with baselines ($\theta = 0.95$). The criteria for evaluation includes the average and standard deviation of travel distance (abbreviated as D), upload data volume (abbreviated as UV), download data volume (abbreviated as DV) and total planning time (abbreviated as T). Lower values for these metrics are preferable ($\downarrow$). When collecting statistics, we report the performance of the worst-performing robot within each scenario.

Nums of robot	mtsp_based		global_map		ours	
	4	8	4	8	4	8
D(m)↓	200.8(±44.0)	182.2(±41.7)	**182.2**(±35.6)	**150.1**(±47.5)	205.7(±64.6)	168.9(±84.45)
UV(MB)↓	656.2(±206.8)	554.7(±191.4)	679.2(±174.1)	516.6(±230.9)	**5.7**(±1.8)	**4.5**(±2.2)
DV(MB)↓	1791.1(±566.0)	3303.1(±1146.9)	2037.7(±522.5)	3616.4(±1616.8)	**17.2**(±5.5)	**32.0**(±15.7)

5.2 Comparisons Analysis

As shown in Figure 5, we construct 100 dungeon maps (100m × 100m) for benchmark testing which have never been seen by the agents during training. We first investigate the performance of the agents' learned messages μ_i^t for information sharing. As illustrated in Table 1, when the team size $N = 4$, our model saves **99.2%** com-

Fig. 5. examples maps from our testing set.

munication volume and surpasses **mtsp_based** by **2.4%** in terms of travel distance. These results indicate that learned messages can help robots reason about the global knowledge/state to generate high-quality, distributed exploration paths, without explicitly sharing detailed maps.

We further train a version of our DRL-based model, where robots are allowed to both share learned messages and explicit partial map. Even when sharing full belief maps, our variant **global_map** still surpasses **mtsp_based** by 9.2% in terms of travel distance. This demonstrates that our approach also exhibits exploration efficiency in high-bandwidth scenarios, where sharing maps among robots is possible.

Finally, we note that the scalability of our approach to larger team sizes is an important criterion for a distributed multi-robot system. Without retraining, we double the team size N from 4 to 8. As demonstrated in Table 1, our approach once again outperforms **mtsp_based** by 7.2% in terms of travel distance. Furthermore, the travel distance margin between **global_map** and our approach still remains similar, even with this larger team size (only 11.1% higher than **global_map** when $N = 8$). We believe that these result demonstrate the robust scalability of our approach.

6 Conclusion

In this work, we propose a DRL-based multi-robot exploration framework that achieves a significant reduction in bandwidth consumption, while maintaining

high exploration efficiency. Our approach leverages communication learning to embed the most salient information from individual beliefs (partial map) of the environment into fixed-sized messages that are shared with other robots. Robots then reason about their own belief as well as received messages to form an *implicit*, global context of the environment (and overall state of the exploration task) and generate distributed exploration paths. To these ends, our approach leverages privileged learning based on learned attention mechanisms, to endow the critic network with ground truth map knowledge and better guide the policy network during training. Compared to our baselines, we demonstrate 99.2% communication bandwidth savings while only sacrificing a marginal 2.4% in travel distance. These results highlight the capability of our robots to understand and reason about the current global state of the exploration task, without explicitly relying on other robots' detailed belief map.

Future works will focus on further reducing the volume of communication within the team, to minimize bandwidth by reducing the frequency of communication autonomously or shrinking message sizes. We will also further enhance the performance of our model with more efficient communication, and consider cases with intermittent communications. Finally, we hope to conduct real-world experiments to validate the feasibility of our method in the future.

7 Acknowledgements

References

1. C Cao, H Zhu, Z Ren, H Choset, and J Zhang. Representation granularity enables time-efficient autonomous exploration in large, complex worlds. *Science Robotics*, 8(80):eadf0970, 2023.
2. Chao Cao, Hongbiao Zhu, Howie Choset, and Ji Zhang. Tare: A hierarchical framework for efficiently exploring complex 3d environments. In *Robotics: Science and Systems*, volume 5, page 2, 2021.
3. Yuhong Cao, Tianxiang Hou, Yizhuo Wang, Xian Yi, and Guillaume Sartoretti. Ariadne: A reinforcement learning approach using attention-based deep networks for exploration. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10219–10225. IEEE, 2023.
4. Yuhong Cao, Zhanhong Sun, and Guillaume Sartoretti. Dan: Decentralized attention-based neural network for the minmax multiple traveling salesman problem. In *International Symposium on Distributed Autonomous Robotic Systems*, pages 202–215. Springer, 2022.
5. Yuhong Cao, Rui Zhao, Yizhuo Wang, Bairan Xiang, and Guillaume Sartoretti. Deep reinforcement learning-based large-scale robot exploration. *IEEE Robotics and Automation Letters*, 2024.
6. Petros Christodoulou. Soft actor-critic for discrete action settings. *arXiv preprint arXiv:1910.07207*, 2019.

7. Jakob Foerster, Ioannis Alexandros Assael, Nando De Freitas, and Shimon Whiteson. Learning to communicate with deep multi-agent reinforcement learning. *Advances in neural information processing systems*, 29, 2016.
8. Benjamin Freed, Guillaume Sartoretti, Jiaheng Hu, and Howie Choset. Communication learning via backpropagation in discrete channels with unknown noise. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7160–7168, 2020.
9. Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018.
10. Ehsan Javanmardi, Mahdi Javanmardi, Yanlei Gu, and Shunsuke Kamijo. Autonomous vehicle self-localization based on probabilistic planar surface map and multi-channel lidar in urban area. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–8. IEEE, 2017.
11. Yukihiro Kida, Mitsuyasu Deguchi, and Takuya Shimura. Experimental result for a high-rate underwater acoustic communication in deep sea for a manned submersible shinkai6500. *The Journal of the Marine Acoustics Society of Japan*, 45(4):197–203, 2018.
12. Daewoo Kim, Sangwoo Moon, David Hostallero, Wan Ju Kang, Taeyoung Lee, Kyunghwan Son, and Yung Yi. Learning to schedule communication in multi-agent reinforcement learning. *arXiv preprint arXiv:1902.01554*, 2019.
13. Woojun Kim, Jongeui Park, and Youngchul Sung. Communication in multi-agent reinforcement learning: Intention sharing. In *International Conference on Learning Representations*, 2020.
14. Eric Klavins. Communication complexity of multi-robot systems. In *Algorithmic Foundations of Robotics V*, pages 275–291. Springer, 2004.
15. Jingsong Liang, Zhichen Wang, Yuhong Cao, Jimmy Chiun, Mengqi Zhang, and Guillaume Adrien Sartoretti. Context-aware deep reinforcement learning for autonomous robotic navigation in unknown area. In *Conference on Robot Learning*, pages 1425–1436. PMLR, 2023.
16. Tianze Luo, Budhitama Subagdja, Di Wang, and Ah-Hwee Tan. Multi-agent collaborative exploration through graph-based deep reinforcement learning. In *2019 IEEE International Conference on Agents (ICA)*, pages 2–7. IEEE, 2019.
17. Kizito Masaba and Alberto Quattrini Li. Gvgexp: Communication-constrained multi-robot exploration system based on generalized voronoi graphs. In *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 146–154. IEEE, 2021.
18. Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I Jordan, et al. Ray: A distributed framework for emerging {AI} applications. In *13th USENIX symposium on operating systems design and implementation (OSDI 18)*, pages 561–577, 2018.
19. Farzad Niroui, Kaicheng Zhang, Zendai Kashino, and Goldie Nejat. Deep reinforcement learning robot for search and rescue applications: Exploration in unknown cluttered environments. *IEEE Robotics and Automation Letters*, 4(2):610–617, 2019.
20. Peng Peng, Ying Wen, Yaodong Yang, Quan Yuan, Zhenkun Tang, Haitao Long, and Jun Wang. Multiagent bidirectionally-coordinated nets: Emergence of human-level coordination in learning to play starcraft combat games. *arXiv preprint arXiv:1703.10069*, 2017.

21. Ioannis M Rekleitis, Vida Dujmovic, and Gregory Dudek. Efficient topological exploration. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 676–681. IEEE, 1999.
22. Abdul Salam, Usman Raza, Abdul Salam, and Usman Raza. Underground wireless channel bandwidth and capacity. *Signals in the Soil: Developments in Internet of Underground Things*, pages 167–188, 2020.
23. Wennie Tabib, Kshitij Goel, John W Yao, Mosam Dabhi, Curtis Boirum, and Nathan Michael. Real-time information-theoretic exploration with gaussian mixture model maps. In *Robotics: Science and Systems*, volume 2, 2019.
24. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
25. Yutong Wang, Mehul Damani, Pamela Wang, Yuhong Cao, and Guillaume Sartoretti. Distributed reinforcement learning for robot teams: A review. *Current Robotics Reports*, 3(4):239–257, 2022.
26. Yutong Wang, Yizhuo Wang, and Guillaume Sartoretti. Full communication memory networks for team-level cooperation learning. *Autonomous Agents and Multi-Agent Systems*, 37(2):33, 2023.
27. Jingtian Yan, Xingqiao Lin, Zhongqiang Ren, Shiqi Zhao, Jieqiong Yu, Chao Cao, Peng Yin, Ji Zhang, and Sebastian Scherer. Mui-tare: Cooperative multi-agent exploration with unknown initial position. *IEEE Robotics and Automation Letters*, 2023.
28. Chao Yu, Xinyi Yang, Jiaxuan Gao, Jiayu Chen, Yunfei Li, Jijia Liu, Yunfei Xiang, Ruixin Huang, Huazhong Yang, Yi Wu, et al. Asynchronous multi-agent reinforcement learning for efficient real-time multi-robot cooperative exploration. *arXiv preprint arXiv:2301.03398*, 2023.
29. Jincheng Yu, Jianming Tong, Yuanfan Xu, Zhilin Xu, Haolin Dong, Tianxiang Yang, and Yu Wang. Smmr-explore: Submap-based multi-robot exploration system with multi-robot multi-target potential field exploration method. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8779–8785. IEEE, 2021.
30. Zhaoliang Zhang, Jincheng Yu, Jiahao Tang, Yuanfan Xu, and Yu Wang. Mr-topomap: Multi-robot exploration based on topological map in communication restricted environment. *IEEE Robotics and Automation Letters*, 7(4):10794–10801, 2022.