Multi-Agent Dynamically Networked and Decentralized Pursuit-Evasion

Maryam Kouzehgar[1], Youngbin Song[2], Marcel Bartholomeus Prasetyo[2], Yi Loo[2], Shanhong Liu[2]
Malika Meghjani[2], and Roland Bouffanais[3]

2025 IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS) | 979-8-3315-9359-9/25/$31.00 ©2025 IEEE | DOI: 10.1109/MRS66243.2025.11357257

***Abstract*— Multi-agent pursuit-evasion scenarios in the presence of learning targets pose a significant challenge for robot swarms. In this work, we investigate a cooperative strategy among decentralized unmanned aerial vehicles to keep track of a rapidly evading target afforded with learning capabilities. To deal with the non-stationarity of the environment while solely depending on local observations, we propose a multi-agent reinforcement learning framework enhanced with dynamic intra-swarm communication protocols available during the training process. Specifically, we propose, (a) a coverage-range-based communication network, and (b) a k-nearest neighborhood communication network which are both developed based on reward policies aimed at maintaining network connectivity despite its dynamic nature. Both of these protocols are integrated into the training process as additional reward-shaping terms, which allow the agents to be deployed in a decentralized manner without requiring any information exchange based on a learned joint policy that adapts to the presence of a communication channel. The multi-agent system is trained in a simulated environment using the Multiple Particle Environment and deployed on Crazyflie drones in a real-world controlled environment. Our experimental results in simulation and real-world show that swarms trained without our proposed networking protocols fail to learn any effective policy. In addition, our proposed dynamic communication strategy during training yields two notable benefits to the system's performance during deployment: (i) it reduces the average distance to the target agent, and (ii) it enables the learning of the pursuit task in larger environments.**

I. Introduction

Multi-robot systems (MRS) and robot swarms have recently attracted considerable attention in robotics [1], [2]. It is now firmly established that the key to unlocking effective swarm behaviors and tapping into swarm intelligence lies in the design of behavioral rules that dictate the movements and interactions of individual agents. This design process can be achieved through biological inspiration [3], where typically the swarming rules are devised from the widely used Alignment–Attraction–Avoidance (AAA) model [4]. Alternatively, the behavioral rules can be automatically established though evolutionary stategies [5], or learned [2]. In this work, we develop a strategy based on Multi-Agent Reinforcement Learning (MARL) to address the complex and challenging problem of pursuit-evasion using swarms of robots. More precisely, we focus on scenarios where a collective of drones is assigned the task of chasing a single, faster target drone that also possesses the ability to learn. With real-world applications in mind, we further require that each pursuing drone has only access to information about its immediate surroundings and cannot send any communication signals when deployed. We restrict intra-swarm communication since communication can be noisy and hence unreliable in a realistic context. In addition, there is the potential for signal sabotage or spoofing from an adversary which may compound the reliability issue.

Considering these limitations, we suggest an approach where individual drones in the pursuing swarm are trained to use swarm networking data without the need for active communication during deployment. This approach avoids di-

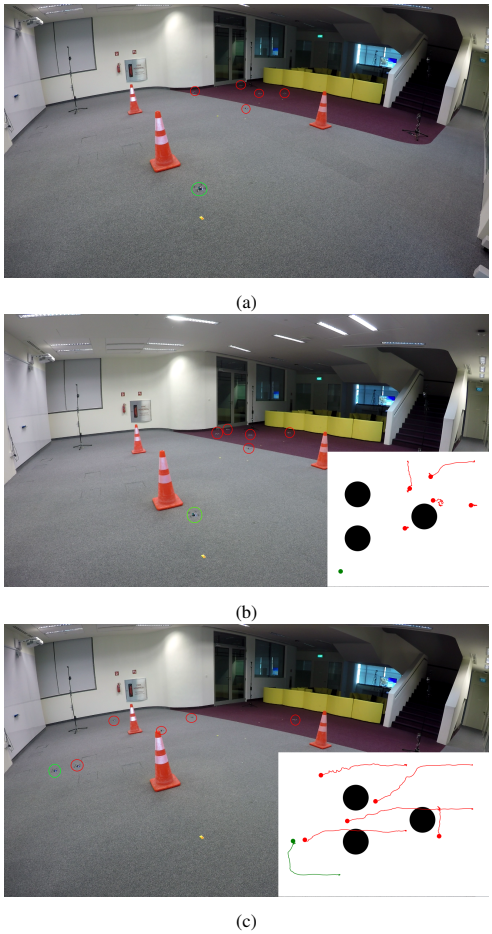


Fig. 1. Multi-player pursuit-evasion drone demo with 5 pursuers (red) and 1 evader (green) (a) initial positions, (b) final positions in the absence of networking, and (c) final positions for kNN-DeNeR

[1]Maryam Kouzeghar is with Engineering Institute of Technology, Melbourne Campus, Australia. maryam.kouzehgar@eit.edu.au

[2]Youngbin Song, Marcel Bartholomeus Prasetyo, Yi Loo, Shanhong Liu, and Malika Meghjani are with the Singapore University of Technology and Design, Singapore. {youngbin_song, marcel_prasetyo, malika_meghjani}@sutd.edu.sg, {yi_loo, shanhong_liu}@mymail.sutd.edu.sg

[3]Roland Bouffanais is with the University of Geneva, Switzerland. roland.bouffanais@unige.ch

rect intra-swarm communication, thereby allowing drones to operate more covertly. Specifically, we utilize the Centralized Training with Decentralized Execution (CTDE) framework, as outlined by Lowe et al. [6] (2017), for the development of MARL algorithms. Following the CTDE approach, agents share a centralized value network during the (centralized) training phase, while solely relying on their own separate policy networks during deployment (i.e., decentralized execution). The centralized nature of the value network enables the integration of network communication data into the training reward, thereby enriching the overall value attributed to the multi-agent swarm. As a direct result, the joint policy learned by the multi-agent system (MAS) also benefits from the network communication information. Specifically, we propose two distinct yet complementary network communication protocols as reward-shaping terms at the training stage. These communication protocols continuously rewire the communication networks that affect the locally shaped reward of each agent at each time step.

Time-varying network topologies are often encountered with rule-based robot swarms, albeit more as an additional constraint to be managed rather than a feature to be exploited [7]. In contrast, we introduce the Decentralized Networking Reward (DeNeR) algorithm, designed to take as an input the topology of the network interconnecting agents, which produces individually tailored rewards for each one. The efficiency of the DeNeR algorithm is qualitatively apparent in Fig. 1 (see insets therein), where we compare the final positions of the pursuer agents trained with DeNeR to those trained without any networking strategy.

The main contributions of this work are threefold: (a) two dynamic network communication protocols facilitating the propagation of the target's location in the presence of partial observability with an approach to compute the local reward for individual agents according to the selected network protocol, (b) we empirically validate that our proposed approach results in the successful pursuit of the target in large environments, in contrast to the original CTDE polices learned without networking protocols that systematically fail at capturing the target, and (c) we experimentally implement our proposed approach in a real-world environment with a swarm of Crazyflie micro-drones.

II. Related Work

In this paper, our main focus is on a specific application, namely multi-player pursuit-evasion, which has broad applications [8]. Given the unavoidable uncertainties and complexities inherent in real-world environments, analytical approaches, as discussed in [9], [10], fall short in effectively handling real-life applications of pursuit-evasion scenarios. As a result, numerous reinforcement learning (RL) strategies, particularly those based on MARL, have been developed to tackle the uncertainties and various challenges encountered in pursuit-evasion scenarios within unfamiliar environments [11]. Most MARL studies fall into three categories: (a) fully centralized RL, which can be considered with all MAS [12], (b) centralized learning and decentralized execution (CTDE style) [6], [13]–[15] applicable to swarm systems that require distributed control in practice, and (c) fully decentralized approaches in some rare exceptions [16]. We consider the CTDE setup for MARL algorithms and develop two variants incorporating our proposed dynamic network communication protocols.

Our previous work focused on single-robot operations to speed up RL strategies by using experience replay [17] and self-organizing maps [18]. In a set of our other studies, a heterogeneous swarm for dynamic monitoring operations was deployed [19] and a swarm of identical autonomous buoys were investigated together with the process of their design and manufacturing [3], [7]. A range of classical swarming behaviors, such as, aggregation, geofencing and heading consensus, are presented in [7] whereas [3] highlights the adaptive environmental monitoring in dynamic regions, but in contrast to the current work, the mentioned previous work are not supported with swarm learning to provide self-programming capabilities.

A key contribution of this study is developing a strategy based on MARL that is enhanced with dynamic networking abilities for MAS, utilizing the characteristics of distributed control. By tuning the communication channel and/or filtering neighboring information, a MRS can control the information flow within the system to maximize the effectiveness of the collective response [20]–[23]. In another recent work [24], we developed a variant of the Multi-Agent Deep Deterministic Policy Gradient (MADDPG) algorithm [6] for ocean monitoring applications using a swarm of buoys. Our most recent work [2] proposed a role-based MADDPG in the frame of multi-target pursuit-evasion problem. Based on this wide range of previous work dealing with networked swarms and MARL [2], [24], we have developed dynamic communication protocols within MARL to leverage the rewiring of networks on multi-agent learning within a cluttered environment with partial observability of the intelligent and evasive target in pursuit mission.

III. Problem formulation

In this paper, the cooperative pursuit-evasion problem is considered for N robots pursuing a single evader in an environment with M randomly scattered obstacles. The evader is considered to be faster and also is taking part in its own individual learning process to fulfill a successful escape.

We make the following assumptions in our work: (a) Pursuers form a homogeneous swarm and they can sense (observe) neighbours in their networked region (section IV-A). (b) The pursuers and evader respectively move at constant speed of v_p and v_e, and the evader is 30% faster than pursuers. (c) At each timestep t, the environment state (denoted by S) consists of agents' velocity in the x- and y-directions, and agents' position in two dimensions (x, y). In addition, the action space is the continuous movement commands also in 2D [25]. (d) Lastly, during training, agents rely on a central critic and upon execution, the i-th pursuer has access to a local observation O_i relying on S, and executes an action a_i. A brief review of the reward

policies for bounding to space, collision avoidance, learning pursuers and learning evaders are thoroughly illustrated in our recent work [2]. In this paper, pursuers will benefit from an additional piece of reward that is governed by dynamic networking protocols.

IV. Proposed Approach

Our proposed network protocol reward focuses on propagating information about the location of the target agent to other agents in the swarm. Each agent determines whether the target agent is in the vicinity of its neighbourhood of agents and propagates that information to other agents in its neighborhood. That information is subsequently propagated to the neighbourhood of each neighbour agent, resulting in a 2-hop propagation network. Each hop decreases the reliability of the propagated information. Each pursuer agent is then informed if its neighbor has the target among its neighbors and also if the neighbor of its neighbor is observing the target. In the following subsections, we will outline the two approaches to compute the set of neighborhood agents which results in the communication protocols for information propagation. This is followed by the approach to compute the overall reward terms for each agent.

A. Communication Protocols

1) Coverage Range (CR) Based Communication Protocol: Inspired by [24], where the concept of robot sensory range (coverage range) was utilized in MARL reward shaping schemes to fulfill area coverage, we apply the same concept to compute the neighborhood set of agents for information propagation. Specifically for each agent i in the pursuer swarm and the set of other agents excluding i, A_{-i}, we define N_i as the set of agents within the neighborhood of i. Agent $j \in A_{-i}$ is said to be in N_i if inter-agent distance, $Dist(i,j) <= d_{cr}$ where d_{cr} is a preset threshold distance. An illustration of our proposed CR network protocol depicting the concept of decreasing trust values is presented in Fig. 2.

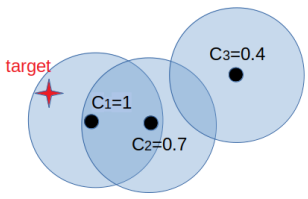


Fig. 2. CR Networking concept with decreasing trust values $c_1 > c_2 > c_3$

2) K Nearest-Neighbours (KNN) Based Communication Protocol: For this protocol, we define N_i as the K-nearest neighbor agents as sorted by $Dist(i,j)$. Specifically, $j \in A_{-i}$ is said to be in N_i if $j \in kNN_i$ where kNN_i is the top k nearest agents by distance to i. A schematic of the concept is illustrated in Fig. 3

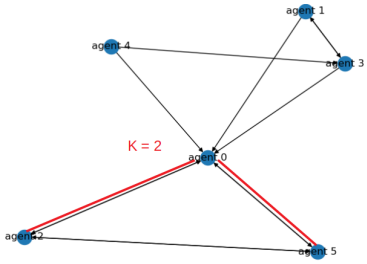


Fig. 3. Concept of networking based on Knn communication protocol

B. Rewarding Policy for Networked Pursuit-Evasion

We can now define the overall network protocol reward for each agent i, R_i^{NP} :

$$R_i^{NP} = c_1 \sum_{j \in N_i} \mathbb{1}_j(target) + c_2 \sum_{j \in N_i} \sum_{k \neq i, k \in N_j} \mathbb{1}_k(target) + c_3 \sum_{j \in N_i} \sum_{k \neq i, k \in N_j} \sum_{l \neq i, l \in N_k} \mathbb{1}_l(target) \quad (1)$$

Where $\mathbb{1}_j(target)$ returns 1 if agent $j \in N_i$ is the target agent. c_1, c_2, and c_3 are used to denote the weighing terms associated with the distance the information has traveled through the network, with c_3 having the smallest value as it denotes agents that are furthest away hence the least reliable. In our implementations we set $c_1 = 1$, $c_2 = 0.7$, and $c_3 = 0.4$ respectively which were chosen empirically.

Apart from networking reward, R_i^{NP}, added upon pursuers' reward, the rest of the rewarding structure [2] is summarized in Table I where B_i is meant for *Bounding to Space* and here is defined to bound the agents in a 10m $\times$ 10m square over $[-5, 5]$ in $x - y$ span. C_i represents *Collision Avoidance* reward in which $Dist(i,j)$ is the distance between two agents and $R(i)$ is the radius of agent i. Within evader's reward (R_{E_i}), CR_{E_i} represents evader's collision (to pursuers) reward, and finally pursuer's reward is computed using B_i, C_i, pursuer's collision (to target) reward CR_{P_i}, pursuer's distance (to target) reward DR_{P_i} and networking reward R_i^{NP}.

In general the CTDE algorithms used for collaborative scenarios, all agents use a shared collective reward. In contrast, in this paper, the reward for each agent includes some aspects of individual reward along with some aspect of collective reward stemming from networking concepts. Accordingly, with partial network-based observations, some levels of decentralization is automatically integrated in the training phase at the level of local rewards. In all of our implementations, we use MADDPG as the underlying MARL algorithm.

V. Simulation Results

A. Effect of Networking protocols on pursuit-evasion in small-scale environment

In this section, the simulation results for three different communication protocols are summarized for comparison in small-scale environment. In all simulations, we have 3 obstacles and 10 agents vs. 1 moving target that is 30%

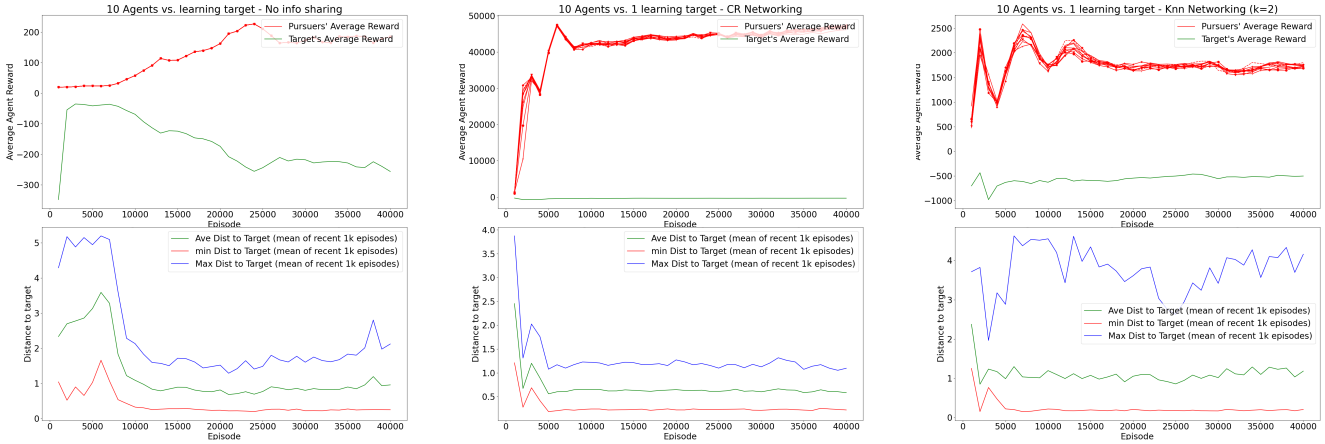


Fig. 4. Quantitative Results for No Networking , CR-DeNeR and kNN-DeNeR in 2 m$\times$ 2 m space representing learning performance (top) and distance to target (bottom)

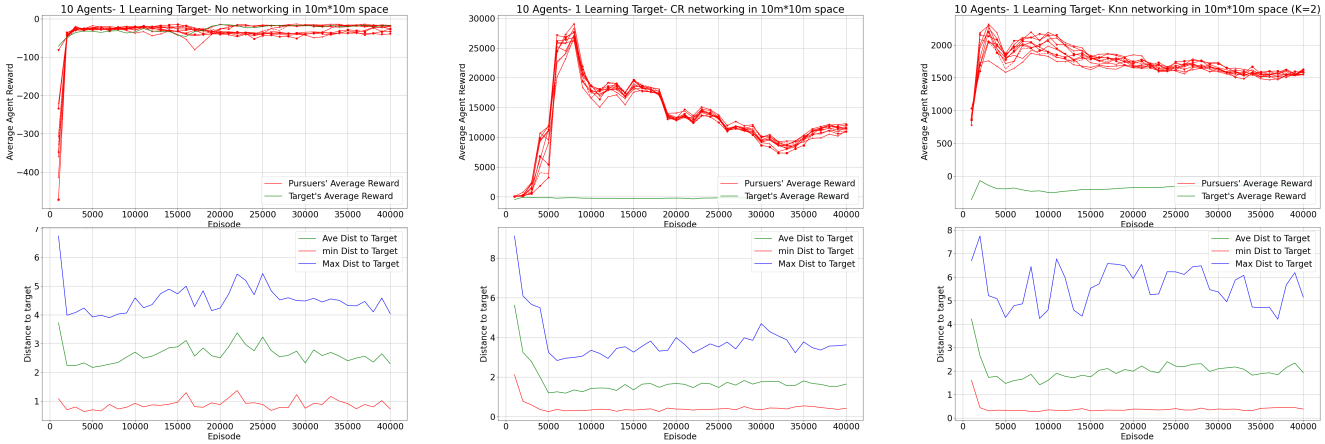


Fig. 5. Quantitative results for No Networking, CR-DeNeR and kNN-DeNeR within 10 m$\times$ 10 m space representing learning performance (top) and distance to target (bottom)

TABLE I: REWARDING POLICIES FOR DYNAMICALLY NETWORKED PURSUIT-EVASION

Rewarding policy	**Reward Function**
Bounding to Space	$B_i = -\sum_{\substack{k_i=x_i,y_i \\ c_i=c_1,c_2}} \min(\exp(5\,\lvert k_i\rvert - 25), c_i)$
Collision Avoidance	$C_i = \begin{cases} -c_3 & \text{if } Dist(i,j) <= R(i) + R(j) \\ 0 & \text{else} \end{cases}$
Learning Evaders	$R_{E_i} = B_i + C_i - CR_{E_i}$
Learning Pursuers	$R_{P_i} = B_i + C_i + CR_{P_i} - DR_{P_i} + R_i^{NP}$

faster than the agents. All entities are randomly scattered in a 2 m $\times$ 2 m space.

The analysis for distance to target is summarized in Table II. Considering the dimensions of agents (diameter= 0.15 m) and the target (diameter= 0.1 m), it is observed that with 0.379 m distance-to-target, the swarm has been successful to closely track the target applying baseline strategy in small-scale environment. In the following, the impact of networking on pursuit mission will be elaborated.

TABLE II: DISTANCE TO TARGET FOR NETWORKING PROTOCOLS IN 2 M $\times$ 2 M SPACE (FIRST 2K EPISODES TRUNCATED)

Ave Distance to target	No Network	CR Network	Knn Network (k=2)
Min	0.37916	0.23986	0.20017
Ave	1.23283	0.63992	1.05621
Max	2.30937	1.20323	3.71482

Fig. 4 illustrates the set of learning performance graphs and the Minimum, Maximum and Average distance to target vs. training episodes for comparison of "no-networking" MADDPG, CR-DeNeR and kNN-DeNeR. In addition, qual-

itative system performance is highlighted in Fig. 6.

Fig. 6. Qualitative performance results at the final time step for (a) No Networking (no-networking), (b) CR-DeNeR, (c) kNN-DeNeR within 2 m$\times$ 2 m space

As illustrated in IV-A.1, for CR-DeNeR, a 2-hop network with corresponding trust values is utilized. With regards to Table II and considering the above-mentioned dimensions for agents and the target, the swarm has not only been successful to closely track the target (min distance=0.239 m), but also the overall average distance to target has been significantly decreased (reduced by 50%) by applying CR-DeNeR.

Identically, for kNN-DeNeR, a 2-nn network with similar trust values to CR protocol is implemented. From Table II, kNN-DeNeR has significantly decreased minimum distance to target and in this protocol the swarm was most closely tracking the target. In addition, the average distance to target is also decreased compared to baseline strategy. Moreover, the increased maximum distance to target shows the enhanced swarm exploratory behavior when applying kNN-DeNeR.

In general, adding networking protocols has enhanced the tracking performance by decreasing the overall distance to target. According to average minimum distances, kNN-DeNeR has been tracking the target most closely with an insignificant difference compared to CR-DeNeR. According to average maximum and mean distances, CR-DeNeR distinctly stands out as it cuts down the distances to almost half compared to no-networking scenario i.e. in CR protocol the swarm leader can closely track the target while strongly dragging the rest of the swarm in the target direction resulting in lower maximum distance to target. The higher average maximum in kNN-DeNeR shows that the swarm has been more scattered around the target and the agents' exploration (vs. exploitation) was favorably addressed in this communication protocol. In terms of learning criteria, CR-DeNeR can hit much higher reward values (Fig. 4, second column) in comparison to kNN-DeNeR (Fig. 4, third column), that is true while apart from networking strategy, they benefit from pretty similar rewarding strategy (Eq. 1) dealing with directly-seen/neighbor-seen targets, and this is another proof to CR-DeNeR's strength to navigate the whole swarm more closely to target. In terms of the speed of convergence, both networking protocols seem to behave almost identically in 2 m $\times$ 2 m space by converging to a smooth reward level around $15k$ episodes.

B. Effect of Networking protocols on pursuit-evasion in larger environments

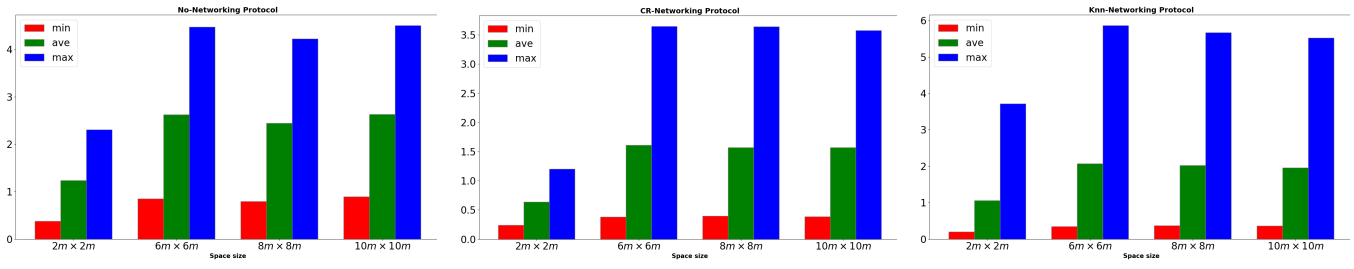
In this section, we aim to check the swarm learning performance in larger environments. In Fig. 5, the simulation results for 10 m$\times 10$ m space are illustrated. Considering the no-networking protocol learning performance graph in larger spaces (first column), it is apparent that without networking protocols, no meaningful learning has occurred for the swarm in larger spaces, and even the target and pursuers have not been able to recognize themselves as distinguished entities to be successful in following their individual rewarding policies resulting in distinct reward levels. Thus, in practice no pursuit-evasion happens based on the learned model, and the agents are lost in the space. This failure of no-networking protocol in larger spaces is identical to naive swarm behavior where swarming rules fail to operate in expanded space due to decreased swarm density [26]. In addition, the distance to target analysis provided for 10 m$\times 10$ m space (Table. III), is reflecting the poor target tracking in no-networking baseline strategy (min dist= 0.89 m) compared to improvements utilizing networking protocols (min dist $\simeq$ 0.3 m). With regards to learning performance graphs for CR and Knn protocols in 10 m$\times$ 10 m space (second and third column), it is observed that firstly, networking protocols can help the swarm to maintain their desired pursuing behaviors amidst the space expansion, and secondly, kNN-DeNeR benefits from a faster convergence in comparison to CR-DeNeR i.e. we have convergence around $20k$ episodes for kNN-DeNeR, while CR-DeNeR shows stronger fluctuations until partial convergence around $40k$ episodes and still needs more episodes to reach a smoother reward level. Fig. 8 highlights the qualitative performance of networking protocols in 10 m$\times$ 10 m space. Furthermore, an illustration of system performance amid incremental space expansion is given in Fig. 7 which shows that while increasing the space (from 2 m$\times$ 2 m to 10 m$\times$ 10 m), adding networking protocols has helped to maintain the minimal distance to target while the baseline no-networking strategy fails in larger environments.

TABLE III: DISTANCE TO TARGET FOR NETWORKING PROTOCOLS IN 10 M$\times$ 10 M SPACE (FIRST 2K EPISODES TRUNCATED)

Ave Distance to target	No Network	CR Network	Knn Network(k=2)
Min	0.89074	0.38519	0.35910
Ave	2.62945	1.56938	1.95634
Max	4.50057	3.57748	5.52517

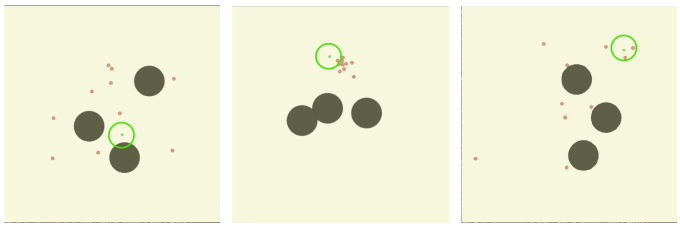
VI. REAL-WORLD RESULTS

For the real-world experiments, we use *Crazyflie 2.1* which is developed by Bitcraze and benefits from a stackable structure of several decks for specific use cases: (a) we use *Loco-Positioning Decks* on each drone as a part of Ultra Wide Band (UWB) *Loco-Positioning System*, (b) a *Flowdeck v2* is attached to the bottom of each drone to sense the height of flight, (c) Using *Crazyradio PA*, position-based movement commands are sent from the computer through the radio communication channels that are configured using *Bitcraze Crazyflie Client*. As depicted in Fig. 9, the experimental environment of 6 m $\times 6$ m is setup with 4 stands on the corners to hold UWB anchors (two anchors attached to each



(a) (b) (c)

Fig. 7. Distance to target bar graphs amidst space expansion for (a) MADDPG (no networking baseline), (b) CR-DeNeR, (c) kNN-DeNeR



(a) (b) (c)

Fig. 8. Qualitative Results for performance of (a) MADDPG (no-networking), (b) CR-DeNeR, (c) kNN-DeNeR in pursuit-evasion within 10 m× 10 m . We highlight the final position of the target agent in the green circle.

stand), a USB radio antenna, and a computer as control station. As static obstacles, 3 traffic safety cones are used. We implemented our proposed communication protocols integrated with MADDPG algorithm with 5 agents against a single faster evasive target. The experimental results are compared for no-networking protocol vs. Knn networking. Following the color code for simulation results, in experimental results the target is annotated in green while the pursuers are circled in red. Fig. 1a illustrates the initial positions of agents for both experiments (considered identical for the sake of comparison). Fig. 1b and Fig. 1c show live trajectories (drones' UWB readings) and final snapshots of the experiment videos respectively for no-networking and Knn networking. Based on Fig. 1c, the pursuit is assessed to be successful for kNN-DeNeR alongside fulfilment of exploration. Based on Fig. 1b, we show that no-networking baseline strategy provides no success in terms of target tracking, and the agents are lost in space and cannot even distinguish their identity as pursuer or evader. We include a full video of our real-world experiments at this link. [1]

VII. Conclusion and Discussion

In this paper, a collaborative framework is considered for a swarm of agents which are tasked to pursue a single evasive target. Both pursuers and evaders are in a multi-agent reinforcement learning (MARL) setup interacting in the same environment, and thus have the opportunity to enhance the expected behaviors via being challenged by the opponent. To further enhance the swarm performance in this setup, we have proposed two networking protocols, namely networking based on coverage range (CR-DeNeR) and networking based

[1]https://youtu.be/zGT0A-szSQU

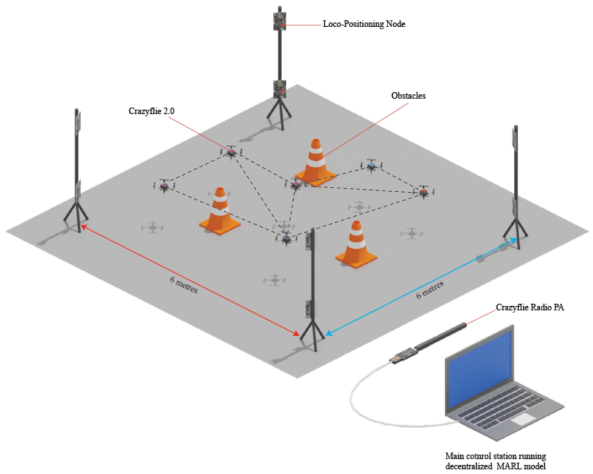


Fig. 9. An illustration of the experimental test-bed

on K nearest neighbors concept (kNN-DeNeR) for swarms of agents to learn joint policies that leverage information from these protocols while being deployed in radio silence. Simulation and experimental results show that relying on the proposed dynamically rewiring networks has substantially benefited the whole swarm specifically in terms of maintaining the expected pursuit-evasion capability amidst significant space expansions. Specifically, in larger spaces, the baseline no-networking protocol fails to learn any meaningful pursuit behavior, leaving the whole swarm uncertain about their identity as a pursuer or evader, let alone fulfilling a successful pursuit-evasion.

The simulation and real-world results show that both networking protocols i.e. CR-DeNeR and kNN-DeNeR have significantly improved the swarm's overall performance. Both the protocols can cut off the minimum distance to target to less than half, however the difference of the proposed approaches lies in their impact on swarm exploring vs. exploiting the whole swarm for target tracking. In this regard, kNN-DeNeR, with higher maximum distance to target, shows a stronger exploration, while CR-DeNeR, results in a more condensed navigation of swarm.

Acknowledgement

This project is supported by the Ministry of Education, Singapore, under its SUTD Kick-starter Initiative (Project number: SKI 2021_05_07) and Google South & Southeast Asia Research Awards (2024).

REFERENCES

[1] H. L. Kwa, J. L. Kit, and R. Bouffanais, "Balancing collective exploration and exploitation in multi-agent and multi-robot systems: A review," *Frontiers in Robotics and AI*, vol. 8, pp. 771520, 2022.

[2] M. Kouzeghar, Y. Song, M. Meghjani, and R. Bouffanais, "Multi-target pursuit by a decentralized heterogeneous uav swarm using deep multi-agent reinforcement learning," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3289–3295.

[3] B. M. Zoss, D. Mateo, Y. K. Kuan, G. Tokić, M. Chamanbaz, L. Goh, F. Vallegra, R. Bouffanais, and D. K. Yue, "Distributed system of autonomous buoys for scalable deployment and monitoring of large waterbodies," *Auton. Robots*, vol. 42, no. 8, pp. 1669–1689, Dec. 2018.

[4] R. Bouffanais, *Design and Control of Swarm Dynamics*, Springer, 2016.

[5] K. Hasselmann, A. Ligot, J. Ruddick, and M. Birattari, "Empirical assessment and comparison of neuro-evolutionary methods for the automatic off-line design of robot swarms," *Nature Communications*, vol. 12, no. 1, pp. 4345, 2021.

[6] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, USA, 2017, NIPS'17, pp. 6382–6393, Curran Associates Inc.

[7] M. Chamanbaz, D. Mateo, B. M. Zoss, G. Tokić, E. Wilhelm, R. Bouffanais, and D. K. P. Yue, "Swarm-enabling technology for multi-robot systems," *Frontiers in Robotics and AI*, vol. 4, pp. 12, 2017.

[8] J. M. Eklund, J. Sprinkle, and S. S. Sastry, "Switched and symmetric pursuit/evasion games using online model predictive control with application to autonomous aircraft," *IEEE Transactions on Control Systems Technology*, vol. 20, no. 3, pp. 604–620, May 2012.

[9] M. Pachter, A. Von Moll, E. Garcia, D. Casbeer, and D. Milutinović, "Cooperative pursuit by multiple pursuers of a single evader," *Journal of Aerospace Information Systems*, vol. 17, no. 8, pp. 371–389, Aug. 2020.

[10] A. Von Moll, D. W. Casbeer, E. Garcia, and D. Milutinović, "Pursuit-evasion of an evader by multiple pursuers," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. June 2018, IEEE.

[11] Y. Wang, L. Dong, and C. Sun, "Cooperative control for multi-player pursuit-evasion games with reinforcement learning," *Neurocomputing*, vol. 412, pp. 101–114, 2020.

[12] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Autonomous Agents and Multiagent Systems*, G. Sukthankar and J. A. Rodriguez-Aguilar, Eds., Cham, 2017, pp. 66–83, Springer International Publishing.

[13] M. Hüttenrauch, A. Šošić, and G. Neumann, "Local communication protocols for learning complex swarm behaviors with deep reinforcement learning," in *International Conference on Swarm Intelligence (ANTS)*. 2018, Springer International Publishing.

[14] M. Hüttenrauch, A. Šošić, and G. Neumann, "Deep reinforcement learning for swarm systems," *Journal of Machine Learning Research*, vol. 20, no. 54, pp. 1–31, 2019.

[15] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *AAAI Conference on Artificial Intelligence*, 2018.

[16] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in *Proceedings of the 35th International Conference on Machine Learning*, Jennifer Dy and Andreas Krause, Eds., Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018, vol. 80 of *Proceedings of Machine Learning Research*, pp. 5872–5881, PMLR.

[17] T. G. Karimpanal and R. Bouffanais, "Experience replay using transition sequences," *Frontiers in Neurorobotics*, vol. 12, pp. 32, 2018.

[18] T. G. Karimpanal and R. Bouffanais, "Self-organizing maps for storage and transfer of knowledge in reinforcement learning," *Adaptive Behavior*, vol. 27, no. 2, pp. 111–126, 2019.

[19] F. Vallegra, D. Mateo, G Tokic, R. Bouffanais, and D. K. P. Yue, "Gradual collective upgrade of a swarm of autonomous buoys for dynamic ocean monitoring," *OCEANS 2018 MTS/IEEE Charleston*, pp. 1–7, 2018.

[20] H. L. Kwa, G Tokić, R. Bouffanais, and D. K. P. Yue, "Heterogeneous swarms for maritime dynamic target search and tracking," *CoRR*, vol. abs/2008.00696, 2020.

[21] H. L. Kwa, J. L. Kit, and R. Bouffanais, "Tailoring exploration and exploitation in multi-agent systems with short-term memory and limited social interaction," 2021 International Workshop on Agent-Based Modelling of Human Behaviour (ABMHuB).

[22] Hian Lee Kwa, Jabez Leong Kit, and Roland Bouffanais, "Tracking multiple fast targets with swarms: Interplay between social interaction and agent memory," *CoRR*, vol. abs/2108.07122, 2021.

[23] D. Mateo, N. Horsevad, V. Hassani, M. Chamanbaz, and R. Bouffanais, "Optimal network topology for responsive collective behavior," *Science Advances*, vol. 5, no. 4, pp. eaau0999, 2019.

[24] M. Kouzehgar, M. Meghjani, and R. Bouffanais, "Multi-agent reinforcement learning for dynamic ocean monitoring by a swarm of buoys," in *Global Oceans 2020: Singapore – U.S. Gulf Coast*, 2020, pp. 1–8.

[25] R. et. al. Lowe, "Multi-agent particle environment," `https://github.com/openai/multiagent-particle-envs`.

[26] H. L. Kwa, J. Philippot, and R. Bouffanais, "Effect of swarm density on collective tracking performance," *Swarm Intelligence*, vol. 17, no. 3, pp. 253–281, 2023.