Adversarial Reinforcement Learning for Deceptive Network-Fragmentation Attacks on Cooperative Unmanned Vehicle Swarms

Erick J. Rodríguez-Seda and Joel M. Esposito

***Abstract*— Cooperative unmanned vehicles rely on connectivity to coordinate their motion and accomplish common tasks. This paper investigates whether a small group of adversarial vehicles can safely fragment a cooperative swarm's communication graph through deceptive motion, without physically removing vehicles or manipulating communication links. We propose a decentralized multi-agent reinforcement learning framework in which adversarial vehicles use local observations to disrupt and fragment a network of cooperative vehicles. The agents are trained with rewards based on connected components, a weighted Laplacian connectivity index, and proximity to influential cooperative vehicles. We evaluate the resulting policies against two heuristic attack strategies across five cooperative connectivity-control laws and four adversarial group sizes. In simulations with 16 cooperative vehicles, the learned strategy achieves the lowest mean algebraic connectivity in 19 of 20 settings and the lowest mean Laplacian first spectral moment in 18 of 20 settings. Its advantage in actual fragmentation is strongest with larger adversarial groups: with 12 adversaries, the cooperative graph is disconnected for approximately 48%–79% of the simulated time, depending on the cooperative controller. These findings demonstrate that local, motion-based adversarial policies can exploit cooperative control behavior to disrupt swarm connectivity.**

I. Introduction

Autonomous swarms are nowadays used for a wide range of applications, including surveillance, environmental monitoring, search and rescue, communication relaying, and force protection [1]. A swarm of autonomous unmanned vehicles (AUV) can replace a single complex machine and complete multiple tasks in a shorter time, cover larger areas, and adapt more easily to single-point failures [2]. These advantages, however, depend strongly on continued cooperation among vehicles, which in turn relies on how well connected the group is [3]. Therefore, an adversarial agent capable of disrupting the behavior or connectivity structure of the swarm can negatively affect the group's objectives and mission.

Recent studies show that AUV swarms are vulnerable to denial-of-service, jamming, spoofing, routing, and false-data injection attacks [4]. Consequently, most existing research focuses on securing communication, identifying malicious nodes [5], [6], or protecting critical AUVs [7]. Alternatively, an adversary may exploit the swarm's own cooperative behavior. Instead of jamming communication or directly compromising a legitimate AUV, an attacker can act like another swarm member and disrupt the group's collective behavior [8], as shown in Fig. 1. For example, [9] uses positional and false-target deception to increase the number of agents moving in different directions, thereby reducing swarm cohesion. Similarly, the meta-deceptive policy-learning framework in [10] trains a deceptive agent to exploit weaknesses in cooperative policies while avoiding detection. Other approaches exploit swarm connectivity to identify and kinetically remove critical nodes that affect their collective behavior [11]–[13]. Yan et al., in particular, explore dismantling networks by exploiting the connectivity of hypergraphs rather than simple graphs [14]. Yet these approaches rely on node elimination rather than deception.

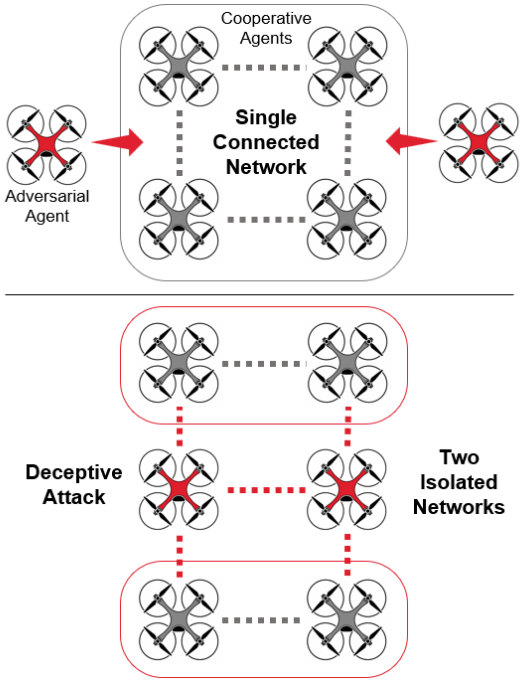


Fig. 1: Motion-based Deceptive Network-Fragmentation Attack. Dashed lines represent information links. Adversarial agents move toward the cooperative swarm and position themselves to effectively eliminate the information path between two groups of vehicles.

In this paper, we investigate using Multi-Agent Reinforcement Learning (MARL) for *deceptive network-fragmentation*

E. J. Rodríguez-Seda and J. M. Esposito are with the Department of Weapons, Robotics, and Control Engineering at the United States Naval Academy, Annapolis, MD. rodrigue@usna.edu, esposito@usna.edu

The views expressed in this document are those of the author(s) and do not reflect the official policy or position of the U.S. Naval Academy, Department of the Navy, the Department of War, or the U.S. Government.

attacks. A deceptive network-fragmentation attack is an adversarial policy that exploits the legitimate behavior of a cooperative multi-agent system to induce the formation of multiple disconnected cooperative subgroups without physically or electronically removing cooperative agents or manipulating their communication channels. We propose a decentralized framework in which a small group of adversarial agents is trained with deep reinforcement learning to minimize the connectivity and induce fragmentation of a cooperative swarm's communication graph, using only local observations and motion-based deception. We define the adversarial reward in terms of algebraic connectivity and the number of connected components, similar to [14]; the proximity or influence exerted on critical nodes (typically those with higher degree or influence on the network [15]); and the Laplacian's first spectral moment. Simulation-based experiments are conducted with different groups of adversarial agents trained against a larger group of vehicles under five different connectivity-maintenance control laws. Results show that our proposed approach outperforms two heuristic baseline methods in most scenarios, particularly for large adversarial groups.

II. Problem Formulation

We consider a swarm of N cooperative (i.e., *friendly*) vehicles and a smaller group of $M < N$ adversarial (i.e., *malicious*) agents, also called deceivers, with equations of motion given by

$$\dot{\mathbf{x}}_i(t) = \mathbf{u}_i(t), \qquad \forall\, i \in \mathbb{N} \cup \mathbb{M},\ t \geq 0 \tag{1}$$

where $\mathbf{x}_i(t) \in \mathbb{R}^2$ are the position coordinates, $\mathbf{u}_i(t) \in \{\mathbf{u} \in \mathbb{R}^2 \mid \|\mathbf{u}\| \leq \bar{u}\}$ are the bounded control inputs, $\bar{u} > 0$ is the vehicles' maximum velocity, and

$$\mathbb{N} = \{1, \ldots, N\}, \qquad \mathbb{M} = \{N+1, \cdots, N+M\}$$

are the sets of cooperative and malicious vehicles, respectively. We assume that all vehicles can receive or detect the position of other vehicles within a bounded range $R > 0$. In what follows, we also assume that all vehicles have the same capabilities and physical properties, and that adversaries' malicious nature is revealed only to other malicious agents and remains hidden from the cooperative swarm.

A. Swarm Connectivity

We describe the connectivity of a set of n vehicles, $\mathcal{V}$, as an undirected R-disk graph $\mathcal{G}(t) = (\mathcal{V}, \mathcal{E}(t))$, where $\mathcal{V}$ are the vertices and $\mathcal{E}(t) = \{(i,j) \in \mathcal{V} \times \mathcal{V} \mid i \neq j, \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| < R\}$ is the set of edges. If two vehicles share an edge, they are said to be adjacent or neighbors. The ith vehicle's neighborhood is its set of neighbors, denoted as $\mathcal{N}_i(t)$. The structural and relationship dynamics of $\mathcal{G}(t)$ can be mathematically represented by the graph's Laplacian, $L(t) = D(t) - A(t)$, where $A(t) = [a_{ij}(t)]_{i,j \in \mathcal{V}}$ is the $n \times n$ distance-based weighted adjacency matrix with ijth entries given by

$$a_{ij}(t) = \begin{cases} \dfrac{(R - \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|)^2}{R^2}, & \text{if } \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| < R \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

and $D(t) = \operatorname{diag}\left(\left[\sum_j^n a_{1j}(t), \cdots, \sum_j^n a_{nj}(t)\right]\right)$ is the weighted degree matrix. Let $\lambda_j(L(t))$ represent the jth smallest eigenvalue of $L(t)$. We say that the graph is connected, meaning that there is a path between any two vehicles, if the second smallest eigenvalue of the Laplacian, $\lambda_2(L(t))$, is greater than zero, and disconnected if $\lambda_2(L(t)) = 0$. In addition, the number of zero eigenvalues is equal to the number of disjoint groups or components [16] and, in general, the larger $\lambda_2(L(t))$ is, the harder it is to split or disrupt the connectivity graph, $\mathcal{G}(t)$.

B. Objectives

The friendly swarm's objective is: 1) to stay near the center of their domain, 2) to remain connected at all times, and 3) to keep a safe relative distance from each other while avoiding dense and rigid connectivity graphs. The latter objective means that the agents should be safely spread around the domain's center and allowed to change neighbors. The objective of the smaller group of adversaries is to split the communication graph safely (i.e., without collisions) by exploiting the cooperative swarm behavior. In short, the attackers want to maximize the fraction of time that $\lambda_2(L(t)) = 0$ as well as the multiplicity of zero eigenvalues.

III. Cooperative Swarm Control

The cooperative swarm has three main objectives: stay close to the origin, remain connected, and avoid collisions. To achieve these objectives, we start by considering the following decentralized, unsaturated control framework for all $i \in \mathbb{N}$

$$\mathbf{u}_i^*(t) = \mathbf{u}_i^P(t) + \mathbf{u}_i^C(t) + \mathbf{u}_i^A(t) \tag{3}$$

where $\mathbf{u}_i^P(t) = -K_P \mathbf{x}_i(t)$ for $K_P > 0$ is a proportional control and $\mathbf{u}_i^C(t)$ is the connectivity maintenance control, to be defined later. The collision avoidance control is given by

$$\mathbf{u}_i^A(t) = \sum_{j \in \mathcal{N}_i(t)} \mathbf{u}_{ij}^A(t) \tag{4}$$

where $\mathcal{N}_i(t) = \{j \in \mathbb{M} \cup \mathbb{N} \setminus i \mid \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| < R\}$ and

$$\mathbf{u}_{ij}^A(t) = \frac{4(R_A^2 - r_A^2)(R_A^2 - \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2)}{(\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|^2 - r_A^2)^3}(\mathbf{x}_i(t) - \mathbf{x}_j(t))$$

if $r_A < \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| < R_A$ and $\mathbf{u}_{ij}^A(t) = \mathbf{0}$ otherwise. The radii $r_A > 0$ and $R_A \in (r_A, R)$ denote the minimum relative safe distance and avoidance control activation range, respectively. It can be shown that, if the $\mathbf{u}_i^C(t)$ is bounded, the unsaturated control (3) will safely drive the swarm to a neighborhood of the origin [17]. Now, since the control input

and velocities are assumed to be bounded, we consider the following saturated control law

$$\mathbf{u}_i(t) = \mathrm{sat}_{\bar{u}}(\mathbf{u}_i^*(t)) = \begin{cases} \mathbf{u}_i^*(t), & \text{if } \|\mathbf{u}_i^*(t)\| \leq \bar{u} \\ \bar{u}\dfrac{\mathbf{u}_i^*(t)}{\|\mathbf{u}_i^*(t)\|}, & \text{otherwise.} \end{cases} \tag{5}$$

Note that, in contrast to $\mathbf{u}_i^*(t)$, (5) cannot guarantee collision avoidance at all times for $N > 2$, a known issue of saturated control systems [18].

As for $\mathbf{u}_i^C(t)$, we consider five different decentralized protocols. Since the adversaries aim to disrupt the swarm through deceptive behavior rather than communication, we emphasize frameworks that rely solely on neighbors' position information, except for one approach in which agents also need to share the states of their local estimators. Similarly, since the swarm is expected to discourage highly dense connectivity graphs and to allow the addition and attrition of edges, we consider only methods in which the number of neighbors remains relatively small. Finally, given the saturation of $\mathbf{u}_i(t)$, we emphasize bounded connectivity laws.

A. Simple Distance-based Unweighted Consensus (SDUC)

A common decentralized yet simple approach to achieve cohesion is the distance-based unweighted consensus protocol (SDUC) [19], [20]:

$$\mathbf{u}_i^C(t) = K_C \sum_{j \in \mathcal{N}_i(t)} (\mathbf{x}_j(t) - \mathbf{x}_i(t)) \tag{6}$$

where $K_C > 0$. Under no disturbances or perturbations and with an initially connected graph, the agents are guaranteed to approach each other.

B. Common-Neighbor-based Weighted Consensus (CNWC)

The SDUC approach treats each neighbor equally regardless of the structural dynamics of the $\mathcal{G}(t)$. Alternatively, one can assign weights to the consensus protocol based on the number of common neighbors, i.e., apply a common-neighbor-based consensus protocol [21]:

$$\mathbf{u}_i^C(t) = K_C \sum_{j \in \mathcal{N}_i(t)} \frac{(\mathbf{x}_j(t) - \mathbf{x}_i(t))}{1 + |\mathcal{N}_i(t) \cap \mathcal{N}_j(t)|}. \tag{7}$$

The CNWC prioritizes (our outweighs) connections with agents with whom it shares the least number of neighbors. Note that the ith vehicle does not need to know $\mathcal{N}_j(t)$ to compute the number of common neighbors [21].

C. Relative-Neighbor-based Unweighted Consensus (RNUC)

Similar to the CNWC framework, one can assign weights to the consensus protocol using the concept of relative neighborhood graphs (RNG) [22], [23], which prioritize agents that are closer to one. Accordingly, one can define the connectivity control as

$$\mathbf{u}_i^C(t) = K_C \sum_{j \in \mathcal{N}_i(t)} (b_{ij}(t) - \varepsilon(1 - b_{ij}(t)))(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \tag{8}$$

where

$$b_{ij}(t) = \begin{cases} 0, & \text{if } \exists k \in \mathcal{N}_i(t) \mid \\ & \max\limits_{\ell \in \{i,j\}} \|\mathbf{x}_\ell(t) - \mathbf{x}_k(t)\| < \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| \} \\ 1, & \text{otherwise} \end{cases} \tag{9}$$

The RNUC framework strengthens connections with neighbors in its RNG and relaxes connections with those outside it.

D. Relative-Neighbor-based Weighted Consensus (RNWC)

The RNUC framework prioritizes those agents that are closer to each other. One can increase their relative strength by assigning distance-based weights

$$\mathbf{u}_i^C(t) = K_C \sum_{j \in \mathcal{N}_i(t)} w_{ij}(b_{ij}(t) - \varepsilon(1 - b_{ij}(t)))(\mathbf{x}_j(t) - \mathbf{x}_i(t)) \tag{10}$$

where

$$w_{ij}(t) = \begin{cases} \dfrac{1}{(R - \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|)^2}, & \text{if } b_{ij} = 1 \\ R^{-2}, & \text{otherwise.} \end{cases} \tag{11}$$

Note that, in the RNWC approach, the control can be unbounded. Yet, compared to other unbounded control methods like [20], RNWC allows edge deletion.

E. Decentralized Maintenance and Connectivity Estimation Consensus (MCEC)

A main challenge in decentralized connectivity maintenance is that no vehicle has complete knowledge of $\mathcal{G}(t)$; therefore, the agents do not know if the group is connected. One control approach to address this challenge is to estimate $\lambda_2(\mathcal{L}(t))$ and its corresponding eigenvector [24], [25], where $\mathcal{L}(t)$ represents the unweighted Laplacian matrix. The control law requires agents to share, in addition to their position, their estimator states, $[\eta_i(t), z_{i,1}(t), \omega_{i,1}(t), z_{i,2}(t), \omega_{i,2}(t)]^\top$, which are updated according to

$$\begin{aligned} \dot{\eta}_i(t) =& -K_1 z_{i,1}(t) - K_2 \sum_{j \in \mathcal{N}_i(t)} (\eta_i(t) - \eta_j(t)) \\ & - K_3(z_{i,2}(t) - 1)\eta_i(t), \\ \dot{z}_{i,k}(t) =& K_4(\eta_i^k(t) - z_{i,k}(t)) - K_5 \sum_{j \in \mathcal{N}_i(t)} (z_{i,k}(t) - z_{j,k}(t)) \\ & + K_6 \sum_{j \in \mathcal{N}_i(t)} (\omega_{i,k}(t) - \omega_{j,k}(t)), \qquad \forall k \in \{1,2\} \\ \dot{\omega}_{i,k}(t) =& -K_6 \sum_{j \in \mathcal{N}_i(t)} (z_{i,k}(t) - z_{j,k}(t)), \qquad \forall k \in \{1,2\} \end{aligned}$$

where $K_\ell > 0$ for $\ell \in \{2,4,5,6\}$ and $K_\ell > K_2\lambda_2(\mathcal{L}(t))$ for $\ell \in \{1,3\}$. The local estimate of $\lambda_2(\mathcal{L}(t))$ is then given by $\hat{\lambda}_{i,2}(t) = K_3(1 - z_{i,2}(t))/K_2$ and the connectivity maintenance control law is

$$\begin{aligned} \mathbf{u}_i^C(t) =& \sum_{j \in \mathcal{N}_i(t)} \frac{(R - \|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|)(\eta_i(t) - \eta_j(t))^2(\mathbf{x}_j(t) - \mathbf{x}_i(t))}{\|\mathbf{x}_j(t) - \mathbf{x}_i(t)\|} \\ & \times \frac{K_C}{\max\{\hat{\lambda}_{i,2}(t) - \mu, \mu\}} \end{aligned} \tag{12}$$

where $\mu > 0$ represents the desired lower bound on $\lambda_2(\mathcal{L}(t))$.

Note that MCEC relies not only on relative position but also requires that each agent in its vicinity, including adversarial agents, share its estimator states. Since the objective of the adversarial agents is opposite to that of the cooperative vehicles, it is assumed that the adversaries share a constant vector $[\eta_i(t), z_{i,1}(t), \omega_{i,1}(t), z_{i,2}(t), \omega_{i,2}(t)]^\top = [c_i, c_i, c_i^2, 0, 0]$, where $c_i \in (0, 1)$ is a random value.

IV. Proposed MARL-based Deceptive Network-Fragmentation Attack

The proposed reinforcement-based framework employs multiple adversarial agents to induce and sustain fragmentation of a cooperative network of vehicles through motion-based interactions. Rather than directly modifying communication links or disabling individual vehicles, the adversarial agents exploit the vehicles' behavior. The framework combines a learned high-level motion policy with a reactive low-level controller. At each decision instant $t_k \in \{kT\}_{k=0}^{\infty}$, where $T > 0$ is the decision or sampling period, every adversarial agent constructs a local observation and independently selects a relative displacement as an action. The low-level controller tracks this waypoint, given by the desired relative displacement, while incorporating inter-vehicle avoidance control. The framework is decentralized because each adversarial policy selects its action using its own local observation, without requiring a global knowledge of the cooperative and adversarial swarms.

A. Adversarial RL Agents

Each RL agent interacts with other cooperative and adversarial vehicles, whose states evolve according to (1). At each time t_k, each agent receives a new reward, $r_i(t_k) \in \mathbb{R}$, and a new observation, $\mathbf{o}_i(t_k) \in \mathbb{R}^{3N+3M-1}$, from the environment:

$$\mathbf{o}_i(t_k) = [\mathbf{x}_i^\top(t_k), \{\Delta\mathbf{x}_{ij}^\top(t_k), m_{ij}(t_k)\}_{j\in\mathbb{N}\cup\mathbb{M}j\neq i}]^\top, \quad \forall i \in \mathbb{M} \tag{13}$$

where $[\Delta\mathbf{x}_{ij}^\top(t_k), m_{ij}(t_k)] = [\mathbf{x}_j^\top(t_k) - \mathbf{x}_i^\top(t_k), 1]$ if $\|\mathbf{x}_j(t_k) - \mathbf{x}_i(t_k)\| \leq R$ and $[\Delta\mathbf{x}_{ij}^\top(t_k), m_{ij}(t_k)] = [\mathbf{0}^\top, 0]$ otherwise. The binary visibility mask, $m_{ij}(t_k) \in \{0, 1\}$, serves to identify vehicles outside the ith agent's range. Without loss of generality, the observation pairs $\{\Delta\mathbf{x}_{ij}^\top(t_k), m_{ij}(t_k)\}$ are ordered as $j \in \{1, \cdots, M + N\}$ (i.e., the adversaries know which position information comes from, if it comes from a cooperative or adversarial agent). Based on its local observation, the RL agent selects an action from its policy, π_θ: $\mathbf{a}_i(t_k) \sim \pi_\theta(\cdot|\mathbf{o}_i(t_k))$, $\mathbf{a}_i(t_k) \in [-\bar{u}T, \bar{u}T] \times [-\bar{u}T, \bar{u}T]$, where $\mathbf{a}_i(t_k)$ represents the desired displacement over a time horizon or time step, $T > 0$. The agents use the selected action to compute a new velocity command according to

$$\mathbf{u}_i(t) = \mathrm{sat}_{\bar{u}}(\mathbf{u}_i^*(t)), \quad \mathbf{u}_i^*(t) = K_P\mathbf{a}_i(t_k) + \mathbf{u}_i^A(t) \tag{14}$$

for all $t \in [t_k, t_{k+1})$, where $\{t_k = kT\}_{k=0}^{\infty}$ is the sequence of time steps, $K_P > 0$ is a proportional gain, and $\mathbf{u}_i^A(t)$ is given by (4).

B. Reward Shaping

Let $\mathcal{V}_i(t) \subseteq \mathbb{N}$ be the set of $n_i(t)$ cooperative vehicles within a distance R from the ith adversarial agent and define $\mathcal{E}_i(t) = \{(\ell, j) \in \mathcal{V}_i(t) \times \mathcal{V}_i(t) \mid \ell \neq j, \|\mathbf{x}_\ell(t) - \mathbf{x}_j(t)\| < R\}$. Let $\mathcal{G}_i(t) = (\mathcal{V}_i(t), \mathcal{E}_i(t))$ be the induced subgraph of $\mathcal{G}(t) = (\mathbb{N}, \mathcal{E}(t))$ visible to the ith adversarial agent, along with the $n_i(t) \times n_i(t)$ weighted Adjacency matrix, $A_i(t) = [a_{\ell j}(t)]_{\ell,j\in\mathcal{V}_i(t)}$, defined according to (2), and the weighted Laplacian matrix

$$L_i(t) = \mathrm{diag}\left(\sum_{j\in\mathcal{V}_i(t)} a_{1j}, \cdots, \sum_{j\in\mathcal{V}_i(t)} a_{n_i(t)j}\right) - A_i(t). \tag{15}$$

Since the eigenvalues of $L_i(t)$ inform the ith agent about the connectivity of $\mathcal{G}_i(t)$, it is convenient to define the following normalized connectivity index for the induced subgraph

$$I(L_i(t)) = \begin{cases} \dfrac{\lambda_2(L_i(t))}{n_i(t)} + \displaystyle\sum_{j=2}^{n_i(t)} \frac{\lambda_j(L_i(t))}{n_i(t)(n_i(t)-1)}, & \text{if } n_i(t) \geq 2 \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

Note that the first term is the algebraic connectivity of the graph normalized by its maximum possible value $n_i(t)$. The second term is the first spectral moment of the Laplacian, which offers information about the density of the group, also normalized by its maximum possible value $(n_i(t) - 1)$. Since both terms are normalized between $[0, 1]$, we have that $I(L_i(t)) \in [0, 2]$ and that $I(L_i(t))$ is 0 for subgraphs where each vehicle is a disconnected component and approaches 2 for complete graphs whose vertices are very close to each other (i.e., when the graph collapses into a single point). Using this connectivity index, we propose the following reward

$$r_i(t_k) = \gamma_F r_{i,F}(t_k) + \gamma_C r_{i,C}(t_k) + \gamma_P r_{i,P}(t_k) \tag{17}$$

where γ_F, γ_C, and γ_P are all positive hyperparameters and $r_{i,F}(t_k)$, $r_{i,C}(t_k)$, and $r_{i,P}(t_k)$ are the fragmentation reward, the low connectivity, and the critical proximity rewards, respectively.

1) Fragmentation Reward: Let $F_i(t_k)$ be the number of components (or clusters) in $\mathcal{G}_i(t_k)$, i.e., the multiplicity of zero Laplacian eigenvalues. We define the fragmentation reward as

$$r_{i,F}(t_k) = F_i(t_k) - 1. \tag{18}$$

Note that the reward increases with the number of components in $\mathcal{G}_i(t_k)$, is 0 if the subgraph is connected, and -1 if the agent has no cooperative vehicle within its sensing range.

2) Low Connectivity Reward: Ideally, the connectivity index of $\mathcal{G}_i(t)$ should be small and decrease over time. Therefore, we propose the following connectivity reward

$$r_{i,C}(t_k) = I(L_i(t_{k-1})) - I(L_i(t_k)). \tag{19}$$

Note that reward is positive if the connectivity index decreases over time and negative if it increases.

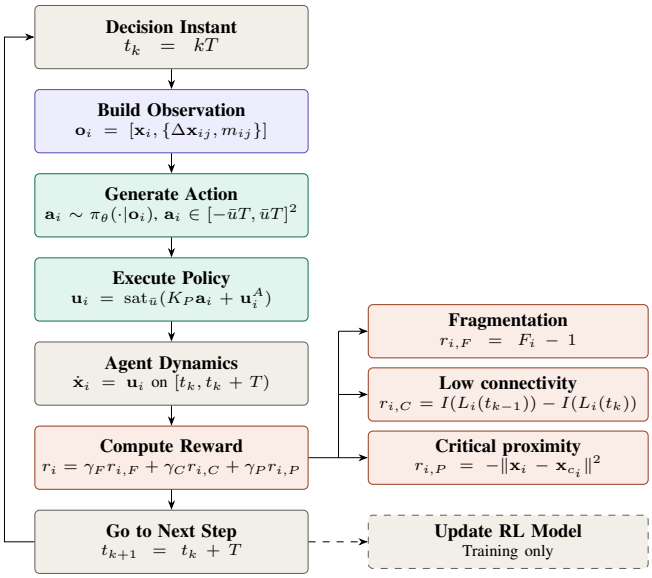


Fig. 2: MARL-based Framework for an Adversarial Agent $i \in \mathbb{M}$

3) Critical Proximity Reward: Finally, we incentivize actions that drive agents toward critical cooperative nodes. Herein, we define a critical or influential node as a cooperative vehicle that, when removed or disturbed, causes a major change in the connectivity of $\mathcal{G}_i(t)$, particularly in the connectivity index I. Accordingly, define $\mathcal{G}_{i/j}(t)$ as the induced weighted subgraph of $\mathcal{G}_i(t)$ obtained when the jth vehicle is removed and $L_{i/j}(t)$ as the corresponding Laplacian. A critical node can be defined as

$$c_i = \underset{j\in\mathcal{V}_i(t)}{\arg\max} \left\{ I(L_i(t)) - I(L_{i/j}(t)) \right\}. \tag{20}$$

Then, the critical proximity reward can be formulated as

$$r_{i,P}(t_k) = \begin{cases} -\|\mathbf{x}_i(t_k) - \mathbf{x}_{c_i}(t_k)\|^2, & \text{if } n_i(t) \geq 1 \\ -R^2, & \text{otherwise} \end{cases} \tag{21}$$

where $\mathbf{x}_{c_i}(t)$ is the position of the critical node.

The proposed RL-framework for a single agent is shown in Fig. 2. It is worth mentioning that the reward scheme incentivizes local fragmentation, which may not necessarily lead to global fragmentation. That is, an induced subgraph visible to an attacker can be disconnected while the full cooperative graph remains connected through vehicles outside its sensing range.

V. Experiments

In this section, we present simulation-based experiments with different groups of adversarial agents against a team of $N = 16$ cooperative vehicles under different connectivity maintenance control methods in a square domain, $\mathbb{Q} = [-40\,\mathrm{m}, 40\,\mathrm{m}]^2$. All vehicles are assumed to have a sensing or communication range of $R = 10$ m and a maximum velocity of $\bar{u} = 2$ m/s. Training and experiments were performed in MATLAB using the Reinforcement Learning Toolbox.

TABLE I: Swarm Connectivity Control Parameters

	SDUC	CNWC	RNUC	RNWC	MCEC
K_C	0.05	0.2	2	1	2
ε	–	–	0.05	0.05	–
μ	–	–	–	–	0.1
$\{K_i\}_{i=1}^6$	–	–	–	–	$\left\{\frac{1}{N+M}, 1, 2, 3, 1, \frac{1}{4}\right\}$

TABLE II: SAC Training Parameters

No. Episodes:	300	Steps per Episode:	200	Replay Buffer:	10^6
Hidden Units:	256	Discount Factor:	0.99	Mini-batch Size:	128
Entropy Weight:	1	Entropy Learn Rate:	0.0005	Target Entropy:	-0.5
Policy Update Frequency:	4	Target Update Frequency:	1	Target Smooth Factor:	10^{-3}

A. Training of Adversarial Agents

The adversarial agents were trained using the entropy-regularized Soft Actor–Critic (SAC) learning method [26]. SAC was adopted due to its continuous-action learning, off-policy sample efficiency, and maximum-entropy objective, which encourages exploration. In addition, SAC has been shown to be relatively stable [26] and well-suited for MARL [27]. The agents were trained under different scenarios: for each of the cooperative swarm control laws in Section III and for different group sizes of adversarial agents, $M = \{2, 4, 8, 12\}$. The group of cooperative vehicles was kept constant at $N = 16$ and the control parameters for each control law were chosen such that, on average and under no adversarial perturbation, the number of edges of their communication graph was $|\mathcal{E}_i(t)| = 40 \pm 5$ and their relative distance was $\|\mathbf{x}_i(t) - \mathbf{x}_j(t)\| = 6.7 \pm 0.2\,\mathrm{m}$ $\forall i \in \mathbb{N}, j \in \mathcal{N}_i(t)$. Other control parameters for the swarm were chosen as $r_A = 1$ m, $R_A = 9$ m, and $K_P = 0.5$ (see Table I for the parameters of each individual connectivity control law). The control parameters for the adversaries were chosen as $r_A = 1$ m and $R_A = 6$ m to allow more aggressive maneuvers around cooperative vehicles, and $T = 5$ s and $K_P = 0.5$ for the control action updates $\mathbf{a}_i(t_k)$. The reward function hyperparameters were chosen as $\gamma_F = 2$, $\gamma_C = 1$, and $\gamma_P = 0.1$.

The training parameters were constant across all training runs and are listed in Table II. Each of the four groups of adversarial agents was trained against one connectivity control model in Section III, for a total of 20 training sets, using a decentralized learning strategy. We chose the best checkpoint model based on the highest cumulative episode reward per training set, using a 20-episode moving average.

B. Baseline

As baselines, we compare the proposed MARL-based framework with two heuristic control approaches. The first baseline approach, BS1, consists of moving all adversarial agents to the center of the space while avoiding collision using the following control law

$$\mathbf{u}_i(t) = \mathrm{sat}_{\bar{u}}(\mathbf{u}_i^*(t)), \quad \mathbf{u}_i^*(t) = -\mathbf{x}_i(t) + \mathbf{u}_i^A(t). \tag{22}$$

Since the cooperative swarm is also attracted to the center of $\mathbb{Q}$ but prioritizes collision avoidance, the adversarial move toward the center should disrupt the cooperative network, $\mathbb{N}$.

The second baseline approach, BS2, consists of moving the agents towards the center of the space but also closer to a critical node using the following decentralized control

$$\begin{aligned}\mathbf{u}_i^*(t) =& 0.5(\mathbf{x}_{c_i}(t) - \mathbf{x}_i(t)) - 0.1\mathbf{x}_i(t) + \mathbf{u}_i^A(t),\\ \mathbf{u}_i(t) =& \text{sat}_{\bar{u}}(\mathbf{u}_i^*(t))\end{aligned} \tag{23}$$

where $\mathbf{x}_{c_i}(t)$ is the position of the critical node according to (20). This approach not only uses the fact that the cooperative swarm is attracted to the same center, but also directly influences the cooperative network by disturbing highly critical nodes.

C. Experimental Scenarios and Performance Criteria

In total, we ran 60 experimental scenarios: one for each combination of adversarial vehicles' group size (i.e., $M \in \{2, 4, 8, 12\}$), cooperative connectivity maintenance control law (SDUC, CNWC, RNUC, RNWC, and MCEC), and adversarial strategy (baselines BS1 and BS2 and proposed RL control approach). We simulated 100 episodes per experimental scenario, each with different initial conditions. Cooperative vehicles were initialized as a connected graph centered at the origin. The adversarial agents were initialized outside the cooperative group. Experiments with the same number of adversarial agents, M, used the same initial conditions. Each episode consisted of 200 *RL*-steps, with a total duration of $1000\,\text{s}$.

The performance of each adversarial control was evaluated based on the percentage of time that the cooperative network was disconnected, the percentage of time it was fragmented into three or more connected groups, the cooperative swarm's average algebraic connectivity value, $\lambda_2(L(t))$, across all times and episodes, and the graph's average first spectral moment, $\sum_{i=i}^{16} \lambda_i(L(t))$, across all times and episodes.

D. Results

Tables III through VII illustrate the results for all experimental scenarios. Overall, the proposed RL-based network-fragmentation attack outperformed both baseline strategies across nearly all metrics and all cooperative control laws for large adversarial groups (i.e., $M \in \{8, 12\}$). This suggests that the adversarial agents learned a decentralized coordination strategy superior to the baseline methods. Furthermore, the proposed approach achieved a higher reduction in algebraic connectivity and first spectral moment in 19 and 18 experiments (out of 20), respectively. For smaller adversarial groups (i.e., $M \in \{2, 4\}$), the proposed RL-based strategy tended to be less effective at splitting the cooperative swarm. It, however, attained the major disruption in connectivity as indicated by the mean algebraic connectivity and spectral moment. As expected, the success of the strategy increased with the size of the group, disconnecting the cooperative network $48\% - 79\%$ of the time for $M = 12$. In terms of safety, Table VII reports the minimum distance between adversarial and cooperative vehicles across all experiments and trials. The adversarial agents always kept a safe distance greater than 1.80 m.

Against the five cooperative connectivity maintenance laws, the proposed approach consistently outperformed both baseline methods across all control laws. However, it performed better against the SDUC and CNWC strategies, achieving the best results in 14 and 15 experiments, respectively, out of 16.

TABLE III: Percentage of Time $\mathcal{G}(t)$ was Disconnected[a,b]

Swarm	$M=2$			$M=4$			$M=8$			$M=12$		
Control	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL
SDUC	1.0	**0.9**	**1.0**	**4.3**	3.5	**1.5**	9.2	**9.0**	**60.0**	**28.0**	28.4	**77.4**
CNWC	**2.0**	0.2	**0.0**	**2.1**	3.8	**4.6**	13.8	**11.4**	**22.6**	**19.0**	19.7	**51.7**
RNUC	**1.0**	0.3	**0.0**	**4.1**	1.4	**0.3**	**4.4**	4.8	**11.6**	16.6	**14.2**	**47.6**
RNWC	**9.7**	5.8	**0.2**	**37.5**	20.7	**5.1**	55.3	**31.1**	**55.7**	69.9	**36.9**	**79.2**
MCEC	7.2	**8.4**	**6.8**	**13.6**	**9.8**	10.9	**17.3**	18.0	**40.0**	36.6	**34.2**	**71.1**

[a] In this and all subsequent tables, green values highlight the top-performing adversarial strategy, whereas red values denote the poorest performance.

[b] In this and all subsequent tables, ties are resolved using unrounded values where applicable.

TABLE IV: Percentage of Time $\mathcal{G}(t)$ was Fragmented into Three or More Connected Components

Swarm	$M=2$			$M=4$			$M=8$			$M=12$		
Control	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL
SDUC	0.00	0.00	0.00	0.01	**0.08**	**0.01**	0.89	**0.57**	**17.1**	**3.21**	3.46	**37.4**
CNWC	0.00	**0.00**	0.00	**0.00**	0.01	**0.08**	**0.04**	0.29	**1.40**	**0.97**	2.11	**11.7**
RNUC	0.00	0.00	0.00	0.00	**0.00**	0.00	**0.03**	0.13	**0.51**	2.29	**0.99**	**10.9**
RNWC	0.00	**0.04**	0.00	**5.71**	3.28	**0.05**	**20.4**	**3.29**	15.9	34.2	**8.24**	**37.7**
MCEC	**2.04**	2.01	**1.07**	**3.87**	2.20	**2.10**	2.51	**1.26**	**6.66**	**4.10**	7.56	**31.1**

TABLE V: Mean Algebraic Connectivity

Swarm	$M=2$			$M=4$			$M=8$			$M=12$		
Control	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL
SDUC	**0.08**	0.08	**0.05**	**0.07**	0.06	**0.03**	**0.04**	0.04	**0.01**	**0.019**	0.018	**0.002**
CNWC	**0.11**	0.11	**0.07**	**0.09**	0.09	**0.04**	0.05	**0.05**	**0.01**	**0.030**	0.028	**0.006**
RNUC	**0.13**	0.13	**0.09**	**0.11**	0.10	**0.05**	**0.07**	0.07	**0.02**	0.037	**0.037**	**0.007**
RNWC	0.06	**0.07**	**0.06**	0.02	**0.05**	**0.03**	0.01	**0.03**	**0.01**	**0.007**	0.017	**0.002**
MCEC	**0.09**	0.08	**0.05**	**0.22**	0.22	**0.03**	0.52	**0.60**	**0.49**	**0.118**	**0.021**	0.042

TABLE VI: Mean First Spectral Moment

Swarm	$M=2$			$M=4$			$M=8$			$M=12$		
Control	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL	BS1	BS2	RL
SDUC	7.2	**7.4**	**6.4**	6.9	**7.0**	**5.9**	6.2	**6.6**	**5.3**	5.4	**5.9**	**3.7**
CNWC	10.2	**10.4**	**8.6**	9.8	**9.9**	**7.7**	8.5	**9.1**	**6.2**	7.4	**8.2**	**5.6**
RNUC	11.5	**11.5**	**9.9**	10.7	**11.0**	**9.0**	9.6	**10.2**	**6.9**	8.3	**9.4**	**5.9**
RNWC	8.3	**9.1**	**7.8**	7.2	**8.6**	**6.7**	**6.4**	**7.8**	6.6	5.6	**7.0**	**4.6**
MCEC	**15.8**	15.5	**14.7**	**20.6**	19.2	**16.6**	**21.3**	21.0	**18.2**	12.5	**9.9**	**13.1**

TABLE VII: Minimum Distance Between Adversarial and Cooperative Agents: $\min_{i\in\mathbb{N},j\in\mathbb{M},t\geq 0} \|\mathbf{x}_i(t) - \mathbf{x}_j(t)\|$.

M	Minimum Distance (m)				
	SDUC	CNWC	RNUC	RNWC	MCEC
2	5.38	4.92	4.46	5.08	4.16
4	5.01	4.02	4.30	4.16	4.27
8	3.35	3.53	3.04	2.78	2.09
12	3.20	1.82	1.80	3.17	3.07

E. Cross-Controller Results

The results from Section V-D assume that the adversarial agents are trained and evaluated under the same cooperative swarm control. Here, we evaluate the performance of the proposed MARL-based framework when the adversaries are trained under one cooperative swarm policy but evaluated under another. Tables VIII through XI list the results for the cases in which four and twelve adversarial agents are trained against the SDUC, RNUC, and MCEC cooperative swarm policies. Then, each trained model was evaluated on a swarm with a different cooperative control law. Similar to the previous section, we simulated 100 episodes per experimental scenario. Note that for almost every case, the results are within one standard deviation of the case when the adversaries are trained and evaluated on the same cooperative control policy (highlighted in gray). This means that the adversarial policies exhibit some cross-controller transfer: the trained agents can effectively disrupt a swarm with a control law that they were not trained against. However, performance degrades in most scenarios, with better fragmentation results when the adversarial policy is evaluated on the cooperative policy it was trained on.

VI. Conclusions and Future Work

This paper presented a decentralized, motion-based adversarial reinforcement learning framework for fragmenting the communication graph of a cooperative unmanned vehicle swarm. Rather than physically removing agents or tampering with communication links, the adversarial agents learn to negatively influence the behavior of the swarm by exploiting their cooperative behavior. Each adversarial agent executes its action using only local information, i.e., position information from other adversarial agents and cooperative vehicles within a bounded range.

We evaluated the performance of the adversarial policy against two heuristic strategies under five different cooperative connectivity maintenance control laws via experimental simulations. The learned strategy generally was able to reduce the cooperative swarm's algebraic connectivity and Laplacian first spectral moment more significantly than the baselines. It also produced substantially more frequent disconnections and fragmentation into three or more components in several settings for large groups of adversarial agents. The adversarial policy also exhibited effective cross-controller transfer. When a trained group was evaluated against a swarm with a different control law not part of the training set, the adversarial group was still effective at disrupting the swarm.

Future work includes ablation studies to isolate the effects of the fragmentation, connectivity, and critical-node proximity rewards. Similarly, we would like to conduct a larger study on how agents trained under a specific cooperative control perform when they encounter cooperative swarms with different behaviors. Other extensions include studying sensitivity to sensing range, observation error, communication delay, and vehicle dynamics.

References

[1] Z. Du, C. Luo, G. Min, J. Wu, C. Luo, J. Pu, and S. Li, "A survey on autonomous and intelligent swarms of uncrewed aerial vehicles (uavs)," *IEEE Transactions on Intelligent Transportation Systems*, vol. 26, no. 10, pp. 14 477–14 500, 2025.

[2] E. J. Rodríguez-Seda, J. J. Troy, C. A. Erignac, P. Murray, D. M. Stipanović, and M. W. Spong, "Bilateral teleoperation of multiple mobile agents: Coordinated motion and collision avoidance," *IEEE Transactions on Control Systems Technology*, vol. 18, no. 4, pp. 984–992, July 2010.

[3] K. Eshaghi, N. N. Sari, C. Haigh, D. Roman, G. Nejat, and B. Benhabib, "Restoring connectivity in robotic swarms–a probabilistic approach," *Journal of Intelligent & Robotic Systems*, vol. 110, no. 2, p. 90, 2024.

[4] X. Wang, Z. Zhao, L. Yi, Z. Ning, L. Guo, F. R. Yu, and S. Guo, "A survey on security of UAV swarm networks: Attacks and countermeasures," *ACM Computing Surveys*, vol. 57, no. 3, pp. 1–37, Nov. 2024.

[5] A. Zilberman, A. Stulman, and A. Dvir, "Identifying a malicious node in a UAV network," *IEEE Transactions on Network and Service Management*, vol. 21, no. 1, pp. 1226–1240, Feb. 2024.

[6] M. Farina, E. Salvato, M. Trevisan, and A. Bartoli, "DRUID: Coordinating drone movements for compromised node identification," *Ad Hoc Networks*, vol. 184, 2026, art. no. 104135.

[7] Y. Tan, J. Liu, and J. Wang, "How to protect key drones in unmanned aerial vehicle networks? an SDN-based topology deception scheme," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 12, pp. 13 320–13 331, Dec. 2022.

[8] S. Li, J. Guo, J. Xiu, Y. Zheng, P. Feng, X. Yu, J. Wang, A. Liu, Y. Yang, B. An, W. Wu, and X. Liu, "Attacking cooperative multi-agent reinforcement learning by adversarial minority influence," *Neural Networks*, vol. 191, p. 107747, Nov. 2025.

[9] Y. Zhu, N. Xie, Z. Xu, and G. Bai, "Dynamics model for swarm disintegration based on behavioral deception," in *2025 16th International Conference on Reliability, Maintainability and Safety (ICRMS)*. IEEE, 2025, pp. 310–315.

[10] W. Hao, H. Liu, H. Lv, W. Li, W. Chen, and L. Chen, "Strategic behavioral deception in cooperative-competitive multi-agent systems," in *2025 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2025, pp. 1–10.

[11] W. Chen, X. Meng, J. Liu, H. Guo, and B. Mao, "Countering large-scale drone swarm attack by efficient splitting," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 9, pp. 9967–9979, 2022.

[12] N. Zhang, C. Liu, and J. Ba, "Decomposing FANET to counter massive UAV swarm based on reinforcement learning," *IEEE Communications Letters*, vol. 27, no. 7, pp. 1784–1788, 2023.

[13] X. Zhang, W. Yue, and L. Zhu, "Countermeasure design for large-scale UAV swarm based on splitting attack," *IEEE Transactions on Network and Service Management*, vol. 23, pp. 4369–4381, 2026.

[14] D. Yan, W. Xie, Y. Zhang, Q. He, and Y. Yang, "Hypernetwork dismantling via deep reinforcement learning," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 3302–3315, 2022.

[15] P. Tee, G. Parisis, and I. Wakeman, "Vertex entropy as a critical node measure in network monitoring," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 646–660, 2017.

[16] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*, ser. Princeton Series in Applied Mathematics. Princeton University Press, 2010.

[17] P. F. Hokayem, D. M. Stipanović, and M. W. Spong, "Coordination and collision avoidance for lagrangian systems with disturbances," *Applied Mathematics and Computation*, vol. 217, no. 3, pp. 1085–1094, 2010.

[18] E. J. Rodríguez-Seda and M. W. Spong, "Guaranteed safe motion of multiple lagrangian systems with limited actuation," in *Proceedings of the Conference on Decision and Control*, Maui, HI, USA, 2012, pp. 2773–2780.

[19] R. Olfati-Saber and R. M. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[20] M. Ji and M. Egerstedt, "Distributed coordination control of multiagent systems while preserving connectedness," *IEEE Transactions on Robotics*, vol. 23, no. 4, pp. 693–703, 2007.

[21] Y. Yang, D. V. Dimarogonas, and X. Hu, "Opinion consensus of modified hegselmann-krause models," *Automatica*, vol. 50, no. 2, pp. 622–627, 2014.

[22] G. T. Toussaint, "The relative neighbourhood graph of a finite planar set," *Pattern Recognition*, vol. 12, pp. 261–268, 1980.

TABLE VIII: Percentage of Time $\mathcal{G}(t)$ was Disconnected for Cross-Controller Evaluation. Values in parentheses denote standard deviations.

Trained On	Evaluated Against									
	SDUC		CNWC		RNUC		RNWC		MCEC	
	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$
SDUC	1.5 (4.8)	77.4 (5.9)	0.1 (0.1)	54.3 (13.5)	0.1 (0.1)	34.7 (12.1)	0.2 (0.7)	66.9 (13.3)	4.9 (19.6)	49.1 (19.0)
RNUC	2.7 (4.3)	78.8 (6.4)	1.5 (2.5)	64.7 (8.7)	0.3 (1.0)	47.6 (10.5)	7.2 (8.9)	78.0 (6.5)	9.7 (18.9)	62.6 (13.9)
MCEC	2.6 (7.0)	75.7 (6.8)	0.8 (1.1)	60.3 (7.0)	0.5 (1.1)	35.5 (7.6)	4.3 (7.4)	77.1 (5.9)	10.9 (20.9)	71.1 (12.8)

TABLE IX: Percentage of Time $\mathcal{G}(t)$ was Fragmented into Three or More Connected Components for Cross-Controller Evaluation. Values in parentheses denote standard deviations.

Trained On	Evaluated Against									
	SDUC		CNWC		RNUC		RNWC		MCEC	
	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$
SDUC	0.01 (0.03)	37.37 (7.64)	0.00 (0.01)	11.38 (7.86)	0.00 (0.01)	4.25 (4.04)	0.00 (0.01)	22.42 (12.66)	1.06 (10.00)	11.87 (10.31)
RNUC	0.02 (0.10)	39.68 (9.21)	0.03 (0.12)	21.76 (6.91)	0.00 (0.00)	10.86 (5.66)	0.10 (0.35)	38.21 (8.02)	2.08 (14.05)	22.94 (11.17)
MCEC	0.01 (0.05)	36.35 (8.16)	0.01 (0.04)	19.31 (5.23)	0.01 (0.01)	6.30 (2.66)	0.04 (0.16)	38.67 (6.86)	2.10 (14.04)	31.13 (14.02)

TABLE X: Mean Algebraic Connectivity for Cross-Controller Evaluation. Values in parentheses denote standard deviations.

Trained On	Evaluated Against									
	SDUC		CNWC		RNUC		RNWC		MCEC	
	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$
SDUC	0.03 (0.01)	0.00 (0.01)	0.07 (0.01)	0.01 (0.01)	0.09 (0.01)	0.02 (0.01)	0.06 (0.01)	0.01 (0.01)	0.06 (0.02)	0.81 (3.28)
RNUC	0.03 (0.01)	0.01 (0.01)	0.05 (0.01)	0.01 (0.01)	0.05 (0.01)	0.01 (0.00)	0.04 (0.01)	0.01 (0.01)	0.20 (1.60)	0.20 (1.39)
MCEC	0.03 (0.01)	0.01 (0.01)	0.05 (0.01)	0.01 (0.01)	0.06 (0.01)	0.01 (0.01)	0.04 (0.01)	0.01 (0.01)	0.03 (0.01)	0.04 (0.4)

TABLE XI: Mean First Spectral Moment for Cross-Controller Evaluation. Values in parentheses denote standard deviations.

Trained On	Evaluated Against									
	SDUC		CNWC		RNUC		RNWC		MCEC	
	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$	$M=4$	$M=12$
SDUC	5.9 (1.5)	3.7 (1.2)	9.6 (1.1)	8.0 (1.1)	10.2 (1.7)	7.0 (1.4)	8.0 (1.0)	6.4 (1.4)	14.4 (33.6)	22.4 (52.8)
RNUC	5.3 (1.7)	4.2 (1.0)	8.0 (1.8)	6.4 (1.0)	9.0 (1.3)	5.9 (1.2)	7.5 (1.0)	4.8 (1.5)	17.6 (44.8)	12.8 (32.0)
MCEC	6.0 (1.0)	4.0 (1.5)	8.0 (1.5)	4.8 (1.2)	9.5 (1.2)	6.7 (1.1)	8.2 (1.0)	4.8 (1.2)	16.6 (36.8)	13.1 (28.1)

[23] J. W. Jaromczyk and G. T. Toussaint, "Relative neighborhood graphs and their relatives," *Proceedings of the IEEE*, vol. 80, no. 9, pp. 1502–1517, 1992.

[24] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar, "Decentralized estimation and control of graph connectivity for mobile sensor networks," *Automatica*, vol. 46, pp. 390–396, 2010.

[25] L. Sabattini, N. Chopra, and C. Secchi, "Decentralized connectivity maintenance for cooperative control of mobile robotic systems," *The International Journal of Robotics Research*, vol. 32, no. 12, pp. 1411–1423, 2013.

[26] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 1861–1870.

[27] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 97. PMLR, 2019, pp. 2961–2970.