DACOOP-A

Decentralized Adaptive Cooperative Pursuit via Attention

Zhang, Zheng; Zhang, Dengyu; Zhang, Qingrui; Pan, Wei; Hu, Tianjiang

DOI
10.1109/LRA.2023.3331886

Publication date
2024

Document Version
Final published version

Published in
IEEE Robotics and Automation Letters

DACOOP-A: Decentralized Adaptive Cooperative Pursuit via Attention

Zheng Zhang, Dengyu Zhang, Qingrui Zhang, *Member, IEEE*, Wei Pan, *Member, IEEE*,
and Tianjiang Hu, *Member, IEEE*

***Abstract*—Integrating rule-based policies into reinforcement learning promises to improve data efficiency and generalization in cooperative pursuit problems. However, most implementations do not properly distinguish the influence of neighboring robots in observation embedding or inter-robot interaction rules, leading to information loss and inefficient cooperation. This letter proposes a cooperative pursuit algorithm named Decentralized Adaptive COOperative Pursuit via Attention (DACOOP-A) by empowering reinforcement learning with artificial potential field and attention mechanisms. An attention-based framework is developed to emphasize important neighbors by concurrently integrating the learned attention scores into observation embedding and inter-robot interaction rules. A KL divergence regularization is introduced to alleviate the resultant learning stability issue. Improvements in data efficiency and generalization are demonstrated through numerical simulations. Extensive quantitative analyses are performed to illustrate the advantages of the proposed modules. Real-world experiments are performed to justify the feasibility of DACOOP-A in physical systems.**

***Index Terms*—Attention mechanism, cooperative pursuit, multi-robot systems, reinforcement learning.**

I. Introduction

COOPERATIVE pursuit aims to coordinate multiple pursuers for capturing one evader in a decentralized manner [1], as shown in Fig. 1. Most existing algorithms employ manually designed rules with domain knowledge [2], [3]. For example, the pursuit problem with collision avoidance is addressed by combining several forces in [4]. However, manually designing cooperative rules in complicated scenarios is intractable, as robots might encounter numerous environment states. Furthermore, the performance of rule-based methods is sensitive to problem settings and parameter configurations, making them inapplicable in real-world tasks [5].

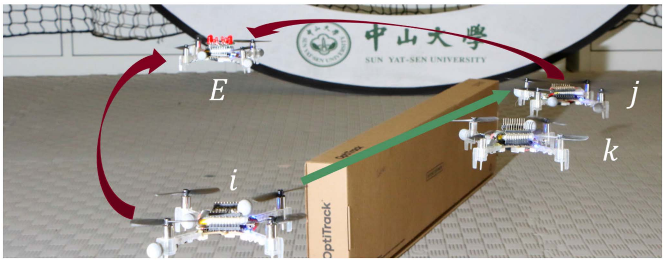


Fig. 1. Cooperative pursuit. E is the evader while i, j, k are pursuers. The red arrows denote the encirclement formed by i and j. It implies j chases E from the right side, while i cuts off the escape route of E from the left side. The green arrow denotes the neighboring robot that i should attend to. k is not attended to because it has less potential to cooperate.

Compared with rule-based methods, reinforcement learning (RL) is more promising for learning sophisticated cooperation because it is possible to obtain various abilities to maximize rewards [6], [7]. However, most RL methods are notorious for the data inefficiency issue that is more severe in multi-robot environments [8]. One of the reasons is an inherent non-stationarity of the environment challenges that value-based RL algorithms. At the same time, policy-based RL methods suffer from a variance that increases as the number of robots [9]. Another challenge for RL is the limited generalization ability. Most RL algorithms focus on maximizing accumulated rewards in predefined training environments. However, the implementation environments commonly have different setups. Such differences would degenerate the performance of the learned policies in real applications [10].

To improve data efficiency and generalizability, our previous work, DACOOP, introduces a hybrid design that integrates rule-based policies, artificial potential field (APF), into RL for cooperative pursuit [11]. Though DACOOP performs better than vanilla RL algorithms, its performance is still limited. The first reason is that the Q network of DACOOP takes the mean embedding of all neighboring robots as input. Since the importance of neighboring robots varies in each state, arbitrary average operation leads to inevitable information loss and data

Manuscript received 21 June 2023; accepted 23 October 2023. Date of publication 10 November 2023; date of current version 3 May 2024. This letter was recommended for publication by Associate Editor A. M. Metelli and Editor J. Kober upon evaluation of the reviewers' comments. This work was supported in part by the National Natural Science Foundation of China under Grants 62103451, 61973327, and 62373386, in part by Shenzhen Science and Technology Program under Grant JCYJ20220530145209021, and in part by Industry-University-Research Fund Project, Ministry of Education of China under Grant 2021ZYA02017. *(Corresponding authors: Tianjiang Hu; Qingrui Zhang.)*

Zheng Zhang, Dengyu Zhang, and Qingrui Zhang are with the School of Aeronautics and Astronautics, Sun Yat-sen University, Shenzhen 518107, China (e-mail: zhangzh363@mail2.sysu.edu.cn; zhangdy56@mail2.sysu.edu.cn; zhangqr9@mail.sysu.edu.cn).

Wei Pan is with the Department of Computer Science, The University of Manchester, M13 9PL Manchester, U.K., and also with the Department of Cognitive Robotics, Delft University of Technology, 2628 CD Delft, The Netherlands (e-mail: wei.pan@tudelft.nl).

Tianjiang Hu is with the School of Aeronautics and Astronautics, Sun Yat-sen University, Shenzhen 518107, China, and also with the School of Artificial Intelligence, Sun Yat-sen University, Zhuhai 519082, China (e-mail: hutj3@mail.sysu.edu.cn).

This letter has supplementary downloadable material available at https://doi.org/10.1109/LRA.2023.3331886, provided by the authors.

Digital Object Identifier 10.1109/LRA.2023.3331886

inefficiency. The second reason is that the mean embedding results usually deviate from those in training scenarios once the system size changes, thus deteriorating the generalization capability of the learned policies. The third reason is that APF is inflexible and suboptimal for multi-robot pursuit problems because it considers all neighboring robots equally.

To tackle the aforementioned problems, a Decentralized Adaptive COOperative Pursuit via Attention (DACOOP-A) algorithm is proposed in this letter by enhancing DACOOP with attention. Our first contribution is an attention-based framework that concurrently integrates the learned attention scores into observation embedding and APF. The attention module is first synthesized with the observation encoding to distinguish important neighboring robots. Compared with mean embedding, it can mitigate information loss and exclude unnecessary information, thus leading to improved generalization. Secondly, the learned attention scores are also employed in APF to weigh the influence of neighboring robots in evaluating inter-robot forces, resulting in an artificial potential field with attention (APF-A) method.

The second contribution of this letter is to improve the learning stability by augmenting standard RL loss functions with a KL divergence regularization. Introducing attention scores in APF-A would make the state transition probability non-stationary in the training process. Hence, the KL divergence regularization is used to penalize foolhardy updates in the outputs of the attention module, which is key to alleviating the non-stationarity issue.

The third contribution is that ablation studies are conducted to show the efficiency of different modules in the proposed algorithm. Extensive quantitative analysis is thereafter performed to illustrate the potential reasons for the advantages of those respective modules. Additionally, the learned policies are deployed directly in physical quadrotor systems to verify the effectiveness of DACOOP-A.

The remainder of this letter is organized as follows. Related works and preliminaries are provided in Sections II and III. Section IV presents the implementation details of the proposed algorithm. Experiment results are given in Section V. Finally, conclusions and future works are available in Section VI.

II. Related Works

Most RL methods solve multi-robot pursuit problems in an independent manner [12], [13] or following the centralized training decentralized execution paradigm [9], [14]. Although their performance has been proven in various problem settings, the problem of varying numbers of neighboring robots is intractable in partially observable environments because fully connected networks necessitate fixed-length inputs. To address this problem, Hüttenrauch et al. proposed mean embedding that averages the embedding of all neighboring robots firstly [15]. To the same end, Everett et al. processed the information of neighboring robots with LSTM [16] and then fed the last hidden state into policy networks [17]. However, both mean embedding and LSTM compress the information of neighboring robots regardless of their significance, usually resulting in the loss of significant information.

Attention mechanisms aim to identify significant elements in sequences and have witnessed exciting success in various domains [18]. In the domain of multi-agent RL, Wen et al. employed Transformer and multi-agent advantage decomposition theorem to transform the joint policy search problem into a sequential decision-making process [19]. Based on the structure of MADDPG, Iqbal et al. used attention mechanisms to synthesize observations of neighboring robots before feeding them into centralized critics [20]. Besides, multi-robot graph attention networks allow robots to focus on communication channels connected to significant neighboring robots [21], [22]. However, all the aforementioned works only employ attention mechanisms as information processing approaches. In comparison, we additionally use the attention scores to improve robots' predefined behavior rules, promising to facilitate the learning process further.

III. Problem Formulation

A. Multi-Robot Pursuit Problem

This letter considers the multi-robot pursuit problem for N robots capturing one faster evader in a confined environment with obstacles. The set of pursuers is denoted by $\mathcal{V} = \{1, 2, \ldots, N\}$, and the evader is indicated by E. The evader is considered captured by $i \in \mathcal{V}$ if $d_{E,i} < d_c$, where $d_{E,i}$ is the distance between E and i. d_c is the capture range. All robots have a safe radius of d_s. Collision occurs if $d_{j,i} < 2d_s$ or $d_{o,i} < d_s$, where $d_{j,i}$ is the distance between i and j. $d_{o,i}$ is the distance between i and the nearest obstacle. Overall, the objective of the pursuers is formulated as follows.

$$\begin{cases} d_{E,i}(t_{max}) < d_c \\ d_{o,i}(t) > d_s \\ d_{j,i}(t) > 2d_s \end{cases} \quad \forall i, j \in \mathcal{V}, i \neq j, \forall t \in [0, t_{max}], \tag{1}$$

where t_{max} is the task horizon. The environment is partially observable. It implies that only neighboring robots within the perception range d_p can be detected by i. Denote the azimuth angle of the evader, the nearest obstacle, the neighboring robot j in the local frame of i as $\phi_{E,i}$, $\phi_{o,i}$, and $\phi_{j,i}$, respectively. The observations of i include the evader information, the nearest obstacle information, and the neighboring robot information, $\{d_{E,i}, \phi_{E,i}, d_{o,i}, \phi_{o,i}, \{d_{j,i}, \phi_{j,i}\}_j\}$, where $j \in \mathcal{N}_i$ and $\mathcal{N}_i$ is the set of observable pursuers of i. The first-order point-mass model is assumed for all robots in this letter.

$$\dot{\boldsymbol{p}} = \boldsymbol{v}, \tag{2}$$

where $\boldsymbol{p}$ is the position and $\boldsymbol{v}$ is the velocity. All robots are assumed to move at a constant speed, which is v_P for pursuers and v_E for the evader. They only change their heading angles. To make learned policies applicable in physical systems, the steering commands of pursuers are limited to $[-30^\circ, 30^\circ]$.

B. Decentralized Adaptive COOperative Pursuit (DACOOP)

The multi-robot pursuit problem can be formulated as a partially observable Markov Game (POMG) that is described by $(\mathcal{S}, \mathcal{V}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \mathcal{Z}, \mathcal{O})$, where $\mathcal{S}$, $\mathcal{V}$, $\mathcal{A}$, $\mathcal{P}$, $\mathcal{R}$, γ, $\mathcal{Z}$, $\mathcal{O}$ are global state space, the set of agents, joint action space, state transition probability, reward function, discount factor, observation function, and local observation space, respectively.

At each timestep, agent i chooses an action a_i according to its local observations o_i that are sampled from the global state s, and then receives its reward r_i. The environment state transits according to the joint action from all agents. The objective of each agent is to learn the optimal policy $\pi^*(a_i|o_i)$ that maximizes its own accumulated rewards.

Our previous work DACOOP employs APF to improve the data efficiency and generalization ability of vanilla RL in multi-robot pursuit problems [11]. Specifically, the APF navigates pursuers through the combination of three predefined forces. For each pursuer $i \in \mathcal{V}$, the attractive force is

$$\boldsymbol{F}_{a,i} = \frac{\boldsymbol{p}_E - \boldsymbol{p}_i}{d_{E,i}}. \tag{3}$$

The repulsive force $\boldsymbol{F}_{r,i}$ is

$$\boldsymbol{F}_{r,i} = \begin{cases} \eta\left(\frac{1}{d_{o,i}} - \frac{1}{\rho}\right)\frac{\boldsymbol{p}_i - \boldsymbol{p}_{o,i}}{d_{o,i}^3}, & \text{if } d_{o,i} \leq \rho \\ \boldsymbol{0}, & \text{if } d_{o,i} > \rho \end{cases} \tag{4}$$

where η is the scale factor and ρ is the influence range of obstacles. The inter-robot force is

$$\boldsymbol{F}_{in,i} = \sum_{j \in \mathcal{N}_i}\left(0.5 - \frac{\lambda}{d_{j,i}}\right)\frac{\boldsymbol{p}_j - \boldsymbol{p}_i}{d_{j,i}}, \tag{5}$$

where $\mathcal{N}_i$ is the set of observable pursuers of i. λ regulates the compactness of the multi-robot system, which is significant for the emergence of cooperation as demonstrated in [11]. The resulting force $\boldsymbol{F}_i = \boldsymbol{F}_{a,i} + \boldsymbol{F}_{r,i} + \boldsymbol{F}_{in,i}$ is used to specify the expected heading of i. To alleviate the local minima issue, the wall following rules are introduced.

DACOOP uses the classical RL algorithm D3QN [23] to learn a shared optimal policy $\pi^*(\lambda, \eta|s)$ that outputs the optimal parameter pair (λ, η) for each pursuer in each time step. Each pursuer calculates its resulting force and moves accordingly after receiving (λ, η).

IV. Methodology

The proposed algorithm DACOOP-A follows the fundamental structure of DACOOP. It implies that DACOOP-A employs vanilla RL to learn the optimal parameter pair (λ, η) at each timestep as usual. However, compared with DACOOP, DACOOP-A introduces an attention module, an artificial potential field with attention (APF-A) method, and a KL divergence regularization to improve the data efficiency and generalization. This section will describe the proposed algorithm from the three aforementioned aspects.

A. Observation Embedding With Attention

Traditional multi-robot pursuit algorithms average the embeddings of neighbors to form fixed-length state representations in partially observable environments [11], [15]. However, such mean embedding makes the information of significant neighbors unrecoverable. Additionally, the resultant embeddings are unreliable once the system size changes because the weights of significant information deviate from those in training scenarios. Therefore, an attention module is introduced to process the observation information in the proposed algorithm.

For each pursuer $i \in \mathcal{V}$, DACOOP-A embeds the information of each observable neighboring robot first,

$$\boldsymbol{e}_{j,i} = f_e(d_{j,i}, \phi_{j,i}), \tag{6}$$

where f_e is a one-layer fully-connected network. The embedding $\boldsymbol{e}_{j,i}$ is then transformed into the *key* $\boldsymbol{k}_{j,i}$ via another one-layer fully-connected network f_k,

$$\boldsymbol{k}_{j,i} = f_k(\boldsymbol{e}_{j,i}). \tag{7}$$

Let $\boldsymbol{o}_{loc,i} = \{d_{E,i}, \phi_{E,i}, d_{o,i}, \phi_{o,i}\}$. The *query* consists of $\boldsymbol{o}_{loc,i}$ and the mean embedding $\boldsymbol{e}_{m,i}$, where

$$\boldsymbol{e}_{m,i} = \frac{1}{|\mathcal{N}_i|}\sum_{j \in \mathcal{N}_i}\boldsymbol{e}_{j,i}. \tag{8}$$

$|\mathcal{N}_i|$ is the number of observable neighboring robots of i. The attention score $\alpha_{j,i}$ is calculated by feeding the query and the key to a one-layer fully-connected attention network f_a followed by a softmax function,

$$\hat{\alpha}_{j,i} = f_a(\boldsymbol{o}_{loc,i}, \boldsymbol{e}_{m,i}, \boldsymbol{k}_{j,i}), \tag{9}$$

$$\alpha_{j,i} = \frac{e^{\hat{\alpha}_{j,i}}}{\sum_{j \in \mathcal{N}_i} e^{\hat{\alpha}_{j,i}}}. \tag{10}$$

Finally, the information of all observable neighboring robots is summarized by taking the weighted mean as follows.

$$\boldsymbol{e}_i = \sum_{j \in \mathcal{N}_i}\alpha_{j,i}\boldsymbol{e}_{j,i}. \tag{11}$$

After concatenated with $\boldsymbol{o}_{loc,i}$, $\boldsymbol{e}_i$ is taken as input by a multi-layer perception (MLP) for further inference, as shown in Fig. 2. All the aforementioned networks f_e, f_k, f_a are trained with the MLP to maximize the accumulated rewards.

B. Artificial Potential Field With Attention (APF-A)

Adjustments to the inter-robot distance characterize most cooperative behaviors. For example, encirclement requires pursuers to keep away from neighbors [3]. Therefore, DACOOP learns the optimal inter-robot forces $\boldsymbol{F}_{in,i}$ to prompt the emergence of cooperation [11]. However, it is inappropriate to average the influence of all observed robots in the computation of inter-robot forces (see (5)) as the cooperation potential of neighboring robots varies. Therefore, the proposed APF-A weights the influence of neighboring robots according to the learned attention scores when computing inter-robot forces,

$$\boldsymbol{F}_{in,i} = \sum_{j \neq i, j \in \mathcal{N}_i}\alpha_{j,i}\left(0.5 - \frac{\lambda}{d_{j,i}}\right)\frac{\boldsymbol{p}_j - \boldsymbol{p}_i}{d_{j,i}}. \tag{12}$$

The intuition behind (12) is that the neighboring robots with large attention scores usually have great potential to cooperate. Learning to adjust the distance from them is more promising for the emergence of cooperation.

C. KL Divergence Regularization

Since the inter-robot forces of APF-A depend on attention scores that are updated as the training proceeds, the expected headings of pursuers are nondeterministic under given local observations and APF-A parameters. Different expected headings

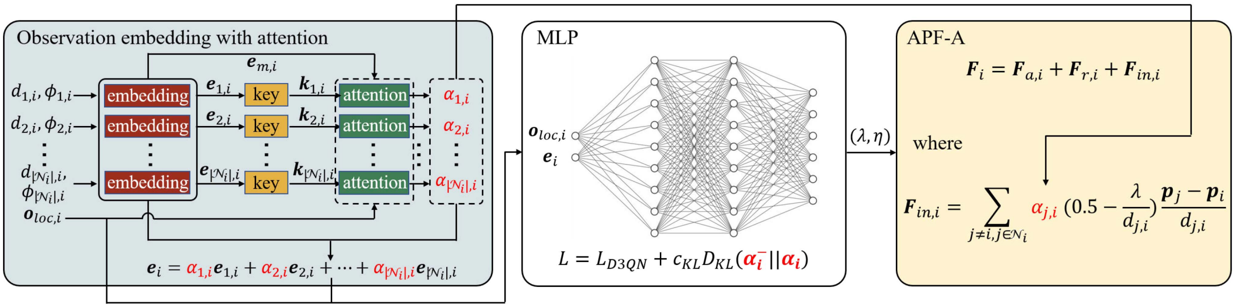

Observation embedding with attention
$d_{1,i}, \phi_{1,i}$
$d_{2,i}, \phi_{2,i}$
$d_{|\mathcal{N}_i|,i}, \phi_{|\mathcal{N}_i|,i}$
$o_{loc,i}$
embedding
$e_{m,i}$
$e_{1,i}$
$e_{2,i}$
$e_{|\mathcal{N}_i|,i}$
key
$k_{1,i}$
$k_{2,i}$
$k_{|\mathcal{N}_i|,i}$
attention
$\alpha_{1,i}$
$\alpha_{2,i}$
$\alpha_{|\mathcal{N}_i|,i}$
$e_i = \alpha_{1,i}e_{1,i} + \alpha_{2,i}e_{2,i} + \cdots + \alpha_{|\mathcal{N}_i|,i}e_{|\mathcal{N}_i|,i}$
MLP
$o_{loc,i}$
e_i
$L = L_{D3QN} + c_{KL}D_{KL}(\alpha_i^- || \alpha_i)$
(λ, η)
APF-A
$F_i = F_{a,i} + F_{r,i} + F_{in,i}$
where
$F_{in,i} = \sum_{j \neq i, j \in \mathcal{N}_i} \alpha_{j,i} (0.5 - \frac{\lambda}{d_{j,i}}) \frac{p_j - p_i}{d_{j,i}}$

Fig. 2. Overview of DACOOP-A. The red, yellow, and green blocks are one-layer fully-connected networks f_e, f_k, and f_a in Section IV-A, respectively. The information from the neighboring robots is summarized by taking the weighted mean, where the weights are the attention scores $\alpha_{j,i}$. The results $\boldsymbol{e}_i$ are taken as input by an MLP together with the information of the evader and the nearest obstacle $\boldsymbol{o}_{loc,i}$. The RL policy outputs parameter pairs (λ, η) for APF-A that weight the influence of neighboring robots according to the learned attention scores in the computation of inter-robot forces $\boldsymbol{F}_{in,i}$. To alleviate the non-stationarity issue, a KL divergence regularization $D_{KL}(\boldsymbol{\alpha}_i^- || \boldsymbol{\alpha}_i)$ is attached to standard D3QN loss function L_{D3QN}.

lead to different observations at the next timestep. So additional non-stationarity is introduced to the state transition probability $p(o'|o, \lambda, \eta)$ by APF-A, presenting dramatic learning stability challenges. Therefore, it is significant to prevent overlarge update steps of attention scores.

Denote the attention score vector of pursuer i as $\boldsymbol{\alpha}_i = [\alpha_{1,i}, \alpha_{2,i}, \ldots, \alpha_{|\mathcal{N}_i|,i}]$. It can be treated as a categorical distribution that indicates the probability of which neighboring robot is significant. Therefore, it is possible to regulate the update of APF-A's behavior rules via minimizing the KL divergence of $\boldsymbol{\alpha}_i$ at adjacent training steps. Since calculating such KL divergence requires additional memory to store Q networks at the previous training step, the target network, which is introduced by DQN [24], is employed to provide the reference distribution instead. Specifically, the KL divergence regularization is defined as

$$D_{KL}(\boldsymbol{\alpha}_i^- || \boldsymbol{\alpha}_i) = \sum_{j=1}^{|\mathcal{N}_i|} \alpha_{j,i}^- \log \frac{\alpha_{j,i}^-}{\alpha_{j,i}}, \tag{13}$$

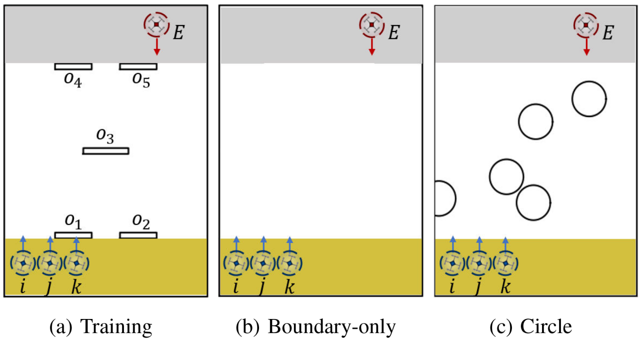
where $\alpha_{j,i}^-$ is the attention score calculated by the target network. Since the weights of the target network are updated periodically, $\boldsymbol{\alpha}_i^-$ is constant in a period of training steps. So minimizing the KL divergence in (13) encourages all $\boldsymbol{\alpha}_i$s during this period to keep close to $\boldsymbol{\alpha}_i^-$, which implicitly prevents overlarge update steps of attention scores. Overall, the loss function of DACOOP-A is

$$L = L_{D3QN} + c_{KL} D_{KL}(\boldsymbol{\alpha}_i^- || \boldsymbol{\alpha}_i), \tag{14}$$

where c_{KL} is a hyperparameter and $L_{D3QN} = [Q(s,a) - (r + \gamma \max_a Q(s', a))]^2$.

V. Results

In this section, the improvement of data efficiency and generalization of DACOOP-A is demonstrated via numerical simulations.[1] The feasibility of learned policies is then evaluated in physical multi-robot systems. Besides, three research questions are investigated in Section V-C, V-D, and V-E, respectively. First, who is attended to in the pursuit process? Second, does APF-A provide better behavior rules? Third, does KL divergence regularization stabilize the learning?



(a) Training (b) Boundary-only (c) Circle

Fig. 3. Pursuit arenas. The evader E is initialized randomly in the gray region while pursuers i, j, k in the yellow region. (a) The training arena where o_1, o_2, o_3, o_4, o_5 are obstacles. (b) The validation arena where obstacles are removed beside the boundary. (c) The validation arena where five circular obstacles are generated with random positions.

TABLE I
Problem Settings and RL Hyperparameters

Parameters	Values	Parameters	Values
N	3	Arena size	$3.6\ m \times 5\ m$
d_c	$0.2\ m$	d_s	$0.1\ m$
d_p	$2\ m$	Step size	$0.1\ s$
v_P	$0.3\ m/s$	v_E	$0.6\ m/s$
t_{max}	$100\ s$	ρ	$0.8\ m$
Learning rate	3e-5	Memory size	1e6
Discount factor	0.99	λ_{max}	4000
Maximal episode	4000	final exploration episode	2000
Minibatch size	128	c_{KL}	0.05

A. Training Settings

The pursuit arena is shown in Fig. 3(a). The problem setting parameters are listed in Table I. The escape policy of the evader is adapted from [4]. It is a force-based method. Pursuers and obstacles repulse the evader via defining multiple forces similar to (3) and (4). The resultant force $\boldsymbol{F}_t$ is employed to guide the evader. The wall following rules are introduced to help the evader

[1] Please refer to https://github.com/Zero8319/DACOOP-A for attached codes and videos.

move along the surface of obstacles when it is in between pursuers and obstacles. As [4] does, a slip rule is employed to help the evader slip through the gap between pursuers when encircled. Totally, the escape policy is $[\boldsymbol{F}_t \vee \text{wall\_following}] \vee slip$, where $\vee$ denotes *or*.

The parameter sharing techniques [7] and the robust RL algorithm D3QN [23] are employed to train the pursuit policies. The action space is 24 pairs of APF-A parameters (λ, η) that are the Cartesian product of 8 λ candidates and 3 η candidates. Specifically, λ candidates linearly range from 0 to λ_{max} that is a hyperparameter, while η candidates are chosen empirically[2]. The reward function consists of three terms, $r = r_{main} + r_{col} + r_{app}$, where r_{main} gives a reward of 20 when the evader is captured. r_{col} gives a punishment of -20 if collisions occur. r_{app} is a reward shaping term that awards pursuers when they approach the evader.

In addition to the proposed algorithm DACOOP-A, several benchmark algorithms are trained as follows.

- *ME [15].* It combines D3QN with mean embedding. The action space is 24 discretized expected headings.
- *D3QN-att.* It combines D3QN with attention. The implementation details of attention are the same as DACOOP-A. The action space is 24 discretized expected headings.
- *DACOOP [11].* The action space is 24 parameter pairs (λ, η) for vanilla APF. Mean embedding is also used.
- *MAAC [20].* MAPPO is selected as the backbone [25]. The attention mechanisms are employed to synthesize the observations of pursuers in the centralized critic.
- *No-RL [4].* It is a non-learning method whose hyperparameters are tuned via the evolutionary algorithm. Since it is difficult for this non-learning method to accomplish the original task, the pursuit problem is simplified so that episodes end when *any* pursuer captures the evader.
- *No-KL.* It is an ablation study by removing KL divergence regularization from DACOOP-A.
- *DACOOP-att.* It is an ablation study by removing both KL divergence regularization and APF-A from DACOOP-A. It implies DACOOP-att is the pure combination of DACOOP and the attention mechanisms.

All algorithms are performed with five random seeds. RL hyperparameters and their values are listed in Table I. Note that the basic hyperparameters, e.g., learning rate, are tuned for ME and then employed by D3QN-att, DACOOP, and DACOOP-A without tuning for a fair comparison. Similarly, we tune λ_{max} for DACOOP and then directly adopt the results in DACOOP-A. Only c_{KL} is tuned for DACOOP-A. The hyperparameters of MAAC are tuned independently.

B. Data Efficiency and Generalization Ability

The learning curves are demonstrated in Fig. 4, and the relative statistical results are listed in Table II. The area under the learning curve (AUC), i.e. the mean success rate achieved at nine checkpoints in Fig. 4, is used as the metric of data efficiency. From Table II, it is observed that DACOOP-based algorithms outperform all baselines in terms of data efficiency due to the introduction of APF. DACOOP-A achieves the best AUC of 0.57 as it employs attention mechanisms to prompt information processing and refine the behavior rules. The poor performance of No-KL indicates the importance of KL regularization in alleviating the non-stationarity issue. MAAC does not obtain a satisfactory AUC because the underlying backbone, MAPPO, is on-policy, which is more data inefficient than off-policy methods in most multi-agent settings. The learned policies' maximal success rate (MSR) is also listed in Table II. Since the action space of DACOOP is not complete, i.e., there may be some expected headings unavailable no matter what value (λ, η) takes, the MSR of DACOOP is inferior to that of ME. However, DACOOP-A alleviates such issues by refining vanilla APF with attention scores, resulting in competitive asymptotic performance while maintaining better AUC. In addition, the collision rate in the training process (CRTP) of DACOOP-based algorithms is much lower than baselines as APF(-A) provides more knowledge in collision avoidance, which is significant for the safety issue in the physical systems training [26].

The generalization performance of different algorithms is shown in Fig. 5. It is observed from 5(a) that DACOOP-A is more robust than ME and D3QN-att when the system size changes. To investigate the underlying reasons, we attempt to

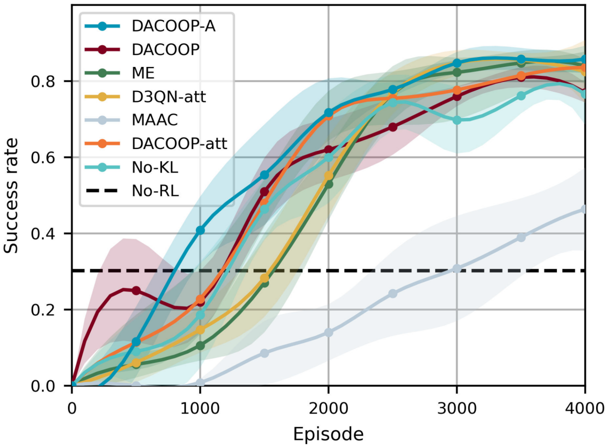


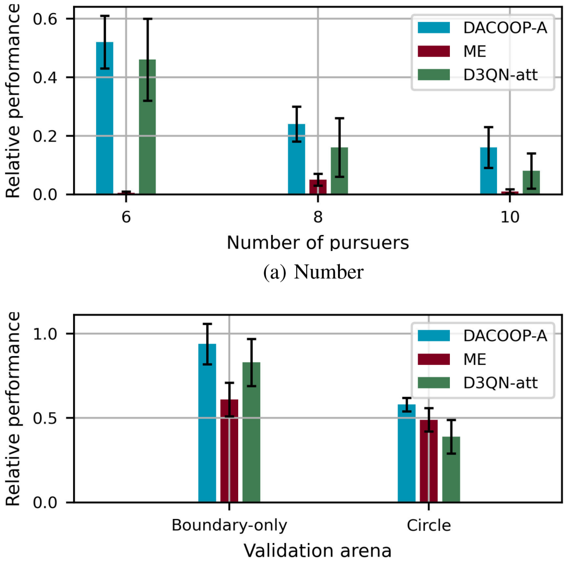
Fig. 4. Learning curves of different algorithms. All results are averaged over 1000 validation episodes and five random seeds. The shaded areas indicate the 95% confidence interval.

TABLE II
MEAN AND STANDARD DEVIATION OF THE LEARNING RESULTS ACROSS DIFFERENT RANDOM SEEDS

	AUC	MSR	CRTP
DACOOP-A	$\mathbf{0.57 \pm 0.03}$	$\mathbf{0.87 \pm 0.03}$	$\mathbf{0.32 \pm 0.01}$
DACOOP	0.51 ± 0.05	0.81 ± 0.02	0.35 ± 0.02
ME	0.47 ± 0.02	0.86 ± 0.02	0.48 ± 0.01
D3QN-att	0.48 ± 0.03	$\mathbf{0.87 \pm 0.03}$	0.47 ± 0.01
MAAC	0.18 ± 0.06	0.52 ± 0.13	0.73 ± 0.04
DACOOP-att	0.54 ± 0.05	0.85 ± 0.04	0.33 ± 0.03
No-KL	0.48 ± 0.04	0.80 ± 0.06	0.35 ± 0.01

The bold values indicate the best performance among the algorithms of interest.

[2] We evaluate the value of η_{min} that could turn the pursuers into the wall following mode when they are very close to obstacles. We then choose $\{\eta_{min}, 10\eta_{min}, 100\eta_{min}\}$ as η candidates.



(a) Number

(b) Arena

Fig. 5. Generalization performance of different algorithms. The relative performance is defined as the ratio of the success rate in the validation scenarios to that in the training scenarios. The error bars indicate the standard deviation across different random seeds.

TABLE III
MEAN AND STANDARD DEVIATION OF AHD ACROSS DIFFERENT RANDOM SEEDS

	DACOOP-A	ME	D3QN-att
AHD	$\mathbf{0.235 \pm 0.018}$	0.587 ± 0.031	0.243 ± 0.046

The bold value indicate the best performance among the algorithms of interest.

measure the variation in observation embedding results when algorithms are deployed in validation scenarios. Firstly, we collect 2×10^6 environment states for each algorithm via uniformly sampling robots' positions and headings. Thereinto, 10^6 states are 3-pursuer scenarios while the other 10^6 states are 10-pursuer scenarios. The integrated embedding $\boldsymbol{e}_i$ is calculated for all pursuers at all states. Note that $\boldsymbol{e}_i$ is the mean embeddings for ME, while is calculated according to (11) for DACOOP-A and D3QN-att. Two matrices $A \in \mathcal{R}^{3e6\times 128}$ and $B \in \mathcal{R}^{10e6\times 128}$ are acquired for each algorithm, where A consists of all integrated embedding results $\boldsymbol{e}_i$ in 3-pursuer scenarios while B consists of that in 10-pursuer scenarios. 128 is the length of embedding vectors. Considering each matrix as a point set in a 128-dimensional space, we use the Hausdorff distance to measure the difference between A and each point in B. The results are averaged over all points in B and denoted as AHD (averaged Hausdorff distance) in Table III. It could be observed that the integrated embeddings $\boldsymbol{e}_i$ of ME change dramatically when they are deployed in systems with different sizes. So, strange state representations are the major impediment to generalization. In comparison, the attention module of DACOOP-A preserves significant information while suppressing redundant information, making $\boldsymbol{e}_i$ more invariant. Although the AHD of D3QN-att is similar to that of DACOOP-A, its success rate is still much lower. The reason is that the optimal policy changes when the system size differs. However, similar observation embeddings result in similar actions in D3QN-att. In comparison, the behavior rules of APF-A depend on the system size, which is promising to provide the desirable adaption based on invariant observation embeddings.[3] Fig. 5(b) shows the generalization ability of each algorithm in validation arenas. The Boundary-only arena provides more free space for both pursuers and the evader, making cooperation more significant than obstacle avoidance for pursuers (see Fig. 3(b)). Therefore, the better generalization of DACOOP-A in this scenario demonstrates that more sophisticated and intelligent cooperative behaviors are learned due to the direct regulation of the distance from significant neighboring robots. The Circle arena employs different obstacles as shown in Fig. 3(c). The better performance of DACOOP-A in this arena verifies the contributions of wall following rules to obstacle avoidance (see Fig. 5(b)).

TABLE IV
FREQUENCY OF CORRESPONDING EVENTS

Event	critic1	critic2	critic3	critic4	critic5
E_1	**75.4%**	**73.7%**	**72.5%**	**73.2%**	**74.5%**
E_2	24.6%	26.3%	27.5%	26.8%	25.5%

The bold values indicate the most frequent events evaluated by different critics.

C. Effects of Attention

To investigate who is attended to in the pursuit process, 3000 episodes are rolled out with policies learned by DACOOP-A. For each pursuer i, we measure the influence of neighboring robots on i's state value by $|V(s_{i,-j}) - V(s_i)|$ and $|V(s_{i,-k}) - V(s_i)|$, where $V(\cdot)$ is the state value function. s_i is the local observations of i. $s_{i,-j}$ denotes removing j from i' observations, while $s_{i,-k}$ denotes removing k. Let E_1 denote the event satisfying

$$(|V(s_{i,-j}) - V(s_i)| - |V(s_{i,-k}) - V(s_i)|)\,(\alpha_{j,i} - \alpha_{k,i}) > 0$$

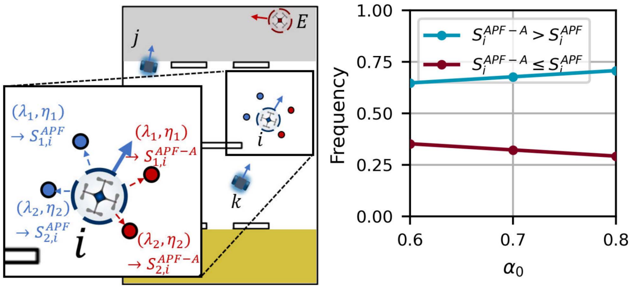
while E_2 denotes events where

$$(|V(s_{i,-j}) - V(s_i)| - |V(s_{i,-k}) - V(s_i)|)\,(\alpha_{j,i} - \alpha_{k,i}) \leq 0.$$

Their frequencies are calculated over trajectories collected in the aforementioned 3000 episodes. Since state values depend on the policy, five critics trained by ME with different random seeds are employed to evaluate $V(\cdot)$ for justice[4]. The results shown in Table IV demonstrate that all critics consistently think E_1 is more frequent than E_2. It implies that the neighboring robots influential on state values are more likely to be attended to. Given the reward function used in this letter, it could be concluded that pursuers attend to neighboring robots mainly for collision avoidance and cooperation.

[3] Note that it should not be expected that agents perform well with unseen observations. The generalization ability of DACOOP-A derives from the adaptation of the policy (Q network + APF-A) instead of the variance of the observation embeddings.

[4] The dueling network in D3QN has two streams to separately estimate the scalar state value and the advantages for each action [23]. Therefore, we take the output of the first stream as the state value in this work.



(a) (b)

Fig. 6. (a) Demonstrations of the evaluation of S_i^{APF} and $S_i^{APF\text{-}A}$. Here taking $|\mathcal{A}| = 2$ for example. Blue (red) circles denote the resultant positions of pursuer i if it moves according to APF (APF-A). The formation score S_i is evaluated for each resultant position. S_i^{APF} is the sum of formation scores evaluated at blue circles, while $S_i^{APF\text{-}A}$ is that at red circles. (b) The frequency of the events $S_i^{APF\text{-}A} > S_i^{APF}$ and $S_i^{APF\text{-}A} \leq S_i^{APF}$ when α_0 takes different values.

D. Effects of APF-A

The magnitude of inter-robot forces increases as inter-robot distance decreases in APF. So the resultant forces $\boldsymbol{F}_i$ of APF-A are similar to that of APF if closer robots are attended to. To distinguish the effects of APF-A, the environment states satisfying the following conditions are selected from the aforementioned 3000 episodes.

- The attention scores $\alpha_{j,i} > \alpha_0$ while $\alpha_{k,i} < 1 - \alpha_0$, where $\alpha_0 > 0.5$. It implies that i attends to j while k is neglected by i.
- The distance between i and E is less than $5d_c$. It implies that i is in a situation where cooperation is important.
- The distance between i and j is larger than that between i and k, meaning the distant robot is attended to.

Similar to [13], the formation score is defined for pursuer i to evaluate the potential for encirclement at a certain state,

$$S_i = \sum_{j=1, i\neq j}^{N} -(\boldsymbol{p}_i - \boldsymbol{p}_E)^T(\boldsymbol{p}_j - \boldsymbol{p}_E). \tag{15}$$

As shown in Fig. 6(a), the usefulness of behavior rules at a certain state s is measured by

$$S_i^{APF} = \sum_{a=1}^{|\mathcal{A}|} S_{a,i}^{APF}, \quad S_i^{APF\text{-}A} = \sum_{a=1}^{|\mathcal{A}|} S_{a,i}^{APF\text{-}A}, \tag{16}$$

where $S_{a,i}^{APF}$ is the formation score S_i of the state transited from s by moving pursuer i according to APF with a-th parameter pair. $S_{a,i}^{APF\text{-}A}$ is that by moving according to APF-A instead. Note that although the 24 parameter pairs (λ, η) are the same for APF-A and APF in this letter, the expected headings differ due to different behavior rules. Both S_i^{APF} and $S_i^{APF\text{-}A}$ are evaluated for all selected states. The proportion of states with $S_i^{APF\text{-}A} > S_i^{APF}$ is much more than that with $S_i^{APF\text{-}A} \leq S_i^{APF}$ no matter what value α_0 takes as shown in Fig. 6(b). It suggests that APF-A provides candidate headings with better quality due to the direct regulation of the distance from significant neighbors.

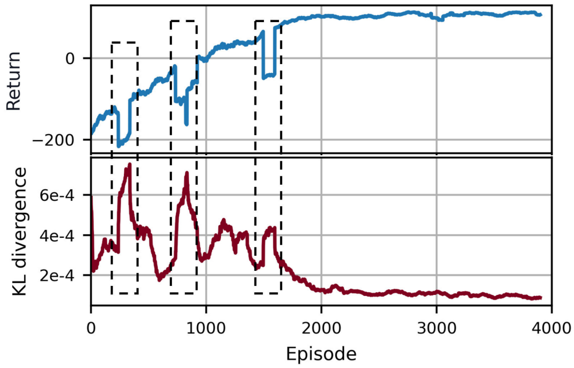


Fig. 7. Example of the learning curves of DACOOP-A. The rise of KL divergence of attention scores always induces drops in accumulated rewards.

TABLE V
STATISTIC RESULTS IN SELECTED DATA POINTS

c_{grad}	1e-5	3e-5	5e-5	7e-5
$Pearson(\nabla D, \nabla R)$	-0.33	-0.43	-0.46	-0.50
P-value	1e-19	5e-6	1e-3	0.01
Amount of data points (No-KL)	195	22	14	9
Amount of data points (DACOOP-A)	149	20	6	3

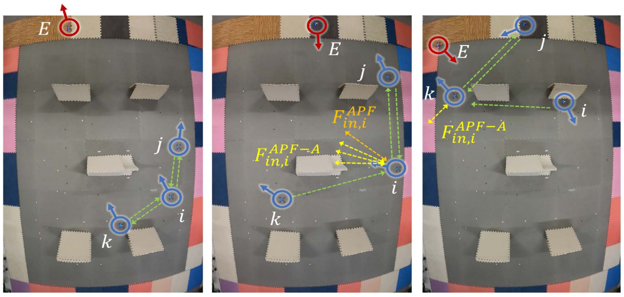
E. Effects of KL Divergence Regularization

The non-stationarity issue derived from evolutionary behavior rules usually induces unstable learning characterized by sudden drops of accumulated rewards, as shown in Fig. 7. To quantify the underlying relations, we record the KL divergence of attention scores D_j and accumulated rewards R_j at each training step j. The result is a list, $\{(D_j, R_j)\}_{j=1}^{T_{max}}$, where T_{max} is the maximum training step. Then, their gradients $\{(\nabla D_j, \nabla R_j)\}_{j=1}^{T_{max}}$ are calculated via the first-order forward difference. To make the result significant, we evaluate the Pearson correlation coefficient between ∇D and ∇R over data points $\{(\nabla D_j, \nabla R_j)\}_j$ that meet $\nabla D_j > c_{grad}$, where c_{grad} is a threshold. The results shown in Table V show that the Pearson correlation coefficient is less than -0.4 if c_{grad} is greater than 1e-5. The existence of moderate negative correlations suggests that an increase in KL divergence usually induces a decrease in the accumulated rewards. Therefore, avoiding an over-large divergence of KL attention scores in the learning process is significant. To investigate whether KL divergence regularization stabilizes the learning process, the overlarge ∇D_j amount is evaluated for DACOOP-A and No-KL, respectively. It can be observed from Table V that DACOOP-A always obtains fewer data points satisfying $\nabla D_j > c_{grad}$, which explains why KL divergence regularization enables greater data efficiency.

F. Physical Experiments

The learned policies of DACOOP-A are deployed in multi-quadrotor systems Crazyflie[5] directly. The positions and orientations of quadrotors are measured by the motion capture system OptiTrack. The key snapshots are shown in Fig. 8. In Fig. 8(a),

[5] https://www.bitcraze.io/products/crazyflie-2-1/



(a) (b) (c)

Fig. 8. Snapshots of real-world experiments. The green arrows point from ego-quadrotors to neighbors who are attended to. The yellow and orange arrows denote $F_{in,i}$ evaluated by APF-A and APF, respectively.

pursuers are distant from the evader. Since their main concern in this state is approaching the evader safely, all pursuers attend to the nearest neighboring robots for collision avoidance. Rather than giving similar attention to j and k in Fig. 8(a), pursuer i switches to attend to j in Fig. 8(b) because j is adjacent to the evader and has more potential to cooperate with. Note that the distance between i and j is similar to that between i and k in Fig. 8(b). It implies that the inter-robot forces evaluated by APF always align with the angular bisector because the forces exerted by neighboring robots are symmetric (see Fig. 8(b)). In comparison, APF-A weights the influence of neighboring robots according to attention scores in evaluating inter-robot forces, providing diverse candidate expected headings even if i is just between two neighboring robots. In Fig. 8(c), pursuer k attends to j since the encirclement they form is the necessity of successful capture. To verify the significance of directly regulating the distance between k and j, the formation scores $S_{a,k}^{APF}$ and $S_{a,k}^{APF\text{-}A}$ are evaluated for 24 parameters pairs (λ_a, η_a) in Fig. 8(c). The results show that only 1 out of 24 parameter pairs satisfies $S_{a,k}^{APF} > S_{a,k}^{APF\text{-}A}$, proving that taking the insignificant neighboring robots i into consider hinders the formation of encirclement. By APF-A, pursuer k cuts off all possible escape routes of the evader in Fig. 8(c) and then successfully captures it.

VI. Conclusions and Perspectives

This letter proposes a multi-robot pursuit algorithm named DACOOP-A by empowering vanilla RL with APF and attention mechanisms. Simulation results demonstrate better data efficiency, competitive asymptotic performance, and lower collision rate in the training process of DACOOP-A. It is also verified that DACOOP-A has greater generalization ability regarding system size and pursuit arena. Further analysis demonstrates that neighboring robots who influence state values are likely to be attended to. In addition, APF-A is proven to provide evolutionary behavior rules that are more promising for encirclement. Simulation results also show that a regularization could alleviate the non-stationarity issue by avoiding overlarge gradients of KL divergence of attention scores in the learning process. Physical experiments verify the feasibility of DACOOP-A in real-world multi-robot systems. However, the action space is not complete in DACOOP-A. Enabling robots to select all possible headings will be considered in future work.

References

[1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics," *Auton. Robots*, vol. 31, no. 4, pp. 299–316, 2011.

[2] Z. Zhou, W. Zhang, J. Ding, H. Huang, D. M. Stipanović, and C. J. Tomlin, "Cooperative pursuit with voronoi partitions," *Automatica*, vol. 72, pp. 64–72, 2016.

[3] X. Fang, C. Wang, L. Xie, and J. Chen, "Cooperative pursuit with multi-pursuer and one faster free-moving evader," *IEEE Trans. Cybern.*, vol. 52, no. 3, pp. 1405–1414, Mar. 2022.

[4] M. Janosov, C. Virágh, G. Vásárhelyi, and T. Vicsek, "Group chasing tactics: How to catch a faster prey," *New J. Phys.*, vol. 19, no. 5, 2017, Art. no. 053003.

[5] C. Muro, R. Escobedo, L. Spector, and R. Coppinger, "Wolf-pack (canis lupus) hunting strategies emerge from simple rules in computational simulations," *Behav. Processes*, vol. 88, no. 3, pp. 192–197, 2011.

[6] D. Silver, S. Singh, D. Precup, and R. S. Sutton, "Reward is enough," *Artif. Intell.*, vol. 299, 2021, Art. no. 103535, doi: 10.1016/j.artint.2021.103535.

[7] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2017, pp. 66–83.

[8] K. Zhang, Z. Yang, and T. Başar, "Multi-agent reinforcement learning: A selective overview of theories and algorithms," in *Handbook of Reinforcement Learning and Control*. Cham, Switzerland: Springer, 2021, pp. 321–384.

[9] R. Lowe, Y. WU, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 6379–6390.

[10] J. Tan et al., "Sim-to-real: Learning agile locomotion for quadruped robots," 2018, *arXiv:1804.10332*.

[11] Z. Zhang, X. Wang, Q. Zhang, and T. Hu, "Multi-robot cooperative pursuit via potential field-enhanced reinforcement learning," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2022, pp. 8808–8814.

[12] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative Markov games: A survey regarding coordination problems," *Knowl. Eng. Rev.*, vol. 27, no. 1, pp. 1–31, 2012.

[13] C. de Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized multi-agent pursuit using deep reinforcement learning," *IEEE Robot. Automat. Lett.*, vol. 6, no. 3, pp. 4552–4559, Jul. 2021.

[14] S. Li, Y. Wu, X. Cui, H. Dong, F. Fang, and S. Russell, "Robust multi-agent reinforcement learning via minimax deep deterministic policy gradient," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4213–4220.

[15] M. Hüttenrauch et al., "Deep reinforcement learning for swarm systems," *J. Mach. Learn. Res.*, vol. 20, no. 54, pp. 1–31, 2019.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] M. Everett, Y. F. Chen, and J. P. How, "Motion planning among dynamic, decision-making agents with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 3052–3059.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*.

[19] M. Wen et al., "Multi-agent reinforcement learning is a sequence modeling problem," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 16509–16521.

[20] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 2961–2970.

[21] Y. Niu, R. R. Paleja, and M. C. Gombolay, "Multi-agent graph-attention communication and teaming," in *Proc. Int. Conf. Auton. Agents Multiagent Syst.*, 2021, pp. 964–973.

[22] Y. Niu, "Adaptable and scalable multi-agent graph-attention communication," *Georgia Inst. Technol.*, 2022.

[23] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas, "Dueling network architectures for deep reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1995–2003.

[24] V. Mnih et al., "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[25] C. Yu et al., "The surprising effectiveness of PPO in cooperative multi-agent games," in *Proc. Adv. Neural Inf. Process. Syst.*, 2022, pp. 24611–24624.

[26] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," 2018, *arXiv:1801.08757*.