Multi-Agent Planning for Pursuing Multiple Objects via Proximal Policy Optimisation

Hoa Van Nguyen[1], Diluka Moratuwage[2], Tran Thien Dat Nguyen[1], Changbeom Shim[1], Ba-Ngu Vo[1]
[1]*School of Electrical Engineering, Curtin University, WA 6102, Australia*
[2]*School of Engineering and Technology, Central Queensland University, QLD 4701, Australia*

Abstract—**This paper addresses the challenge of efficient multi-agent planning for capturing multiple dynamic objects using deep reinforcement learning (DRL). We propose a scalable solution based on the multi-agent proximal policy optimisation (MAPPO) with three key innovations. First, we employ a dynamic state representation that adapts to a variable number of objects, ensuring scalability. Second, we introduce an optimal task allocation mechanism using the Hungarian algorithm to provide a cooperative signal for agent assignments. Finally, this is coupled with a multi-component reward function designed to incentivise both rapid and coordinated pursuit. Experiments demonstrate that our method enables agents to efficiently and collaboratively pursue all objects across various scenarios, significantly outperforming strong multi-agent reinforcement learning (MARL) baselines in both coordination and efficiency. The results highlight a practical pathway toward developing effective, adaptive multi-agent systems for complex, real-world tasks.**

Index Terms—**MAPPO, Multi-Agent Planning, MARL, Cooperative AI, Multi-agent Pursuit**

I. Introduction

Multi-agent planning in the pursuit tasks, where multiple autonomous agents coordinate to pursue several moving objects, are a fundamental problem in robotics, security, and surveillance applications [1], [2]. These scenarios require agents to perceive the dynamic environment, plan actions, and collaborate efficiently in the presence of uncertainty and a time-varying number of objects [3]. The computational complexity of such tasks increases exponentially when the number of agents and the number of objects increase, resulting in a combinatorial growth in both the state and action spaces [4]. Therefore, efficient solutions to this multi-agent planning problem are of significant theoretical and practical interest.

Traditional approaches to multi-agent pursuit-evasion problems included rule-based coordination strategies [5], centralised task assignment algorithms [6], and heuristic search methods [7]. However, these approaches might suffer from limited scalability, restricted adaptability to dynamic changes in the environment, or a reliance on predefined behaviours that may not generalise to novel scenarios [1].

More recently, reinforcement learning (RL) has emerged as a promising paradigm for learning efficient pursuit strategies autonomously, allowing agents to adapt their behaviours based on experience and implicit environmental feedback [8]–[12]. Existing reinforcement learning methods for multi-agent pursuit generally fall into two categories: independent learners, where each agent learns its policy in isolation; and joint learners, where a centralised policy is utilised for the entire team [13], [14]. While independent learners can suffer from suboptimal coordination, centralised approaches are typically infeasible for problems involving many agents or objects due to the exponential growth of the joint state-action space [15]. Some recent works have explored dynamic state abstraction and shared reward mechanisms to improve scalability and cooperation [16]–[18], but often do not fully address the need for flexible state representations and robust coordination in the face of varying numbers of agents and objects.

This paper presents an efficient and adaptive deep reinforcement learning solution for the multi-agent pursuit of multiple objects. Our approach leverages MAPPO [18], in which each agent receives a time-varying state vector, encoding the current state of all agents and active objects, including their identities. This representation enables seamless adaptation to the removal of captured objects and avoids the weakness of a fixed state-space representation.

The main contributions of this work are threefold: (i) a scalable and dynamic state representation for multi-agent pursuit that addresses the variable number of objects; (ii) an optimal task allocation mechanism via the Hungarian algorithm to provide a global coordination signal; and (iii) a multi-component reward mechanism designed to encourages both rapid and coordinated capture. Our empirical experiments demonstrate scalability and superior coordination across various settings, outperforming a strong MAPPO baseline [18] and a planning-based, non-learning Rollout method [14]. Our results suggest that such adaptive, reward-driven agent cooperation and the numerical efficiency of DRL provide a practical pathway to effective multi-agent planning in complex, real-world environments.

II. Related Work

Our work is situated at the intersection of multi-agent reinforcement learning (MARL) and robotic search and pursuit. We build upon key developments in both areas. One major line of research in MARL focuses on overcoming the challenges of non-stationarity and credit assignment. Early methods treated each agent as an independent learner, e.g., the Deep Q-Network (DQN) [8], which simplifies the problem but often fails to con-

The ARC projects LP200301507 and the AKF of the Department of Foreign Affairs and Trade support our research.

*Corresponding author: hoa.v.nguyen@curtin.edu.au

verge to coordinated policies. To address this, the paradigm of Centralised Training with Decentralised Execution (CTDE) was proposed. Seminal works such as MADDPG [10] introduced agent-specific critics that have access to global information during training, allowing for more stable learning of coordinated behaviours while keeping execution decentralised. Other value-based methods like VDN [19], QMIX [20], and QPLEX [21] learn a joint action-value function that is factorised from individual agent utilities, ensuring that a better joint action corresponds to better individual actions.

Our approach aligns with the policy-based side of CTDE, building on MAPPO [18]. Recent advancements in policy-based MARL have explored communication protocols and attention mechanisms to improve coordination. For instance, TarMAC [22] introduced a signature-based soft attention mechanism for targeted communication. Another recent trend focuses on improving credit assignment within cooperative settings. For example, Weighted QMIX [23] introduces a weighting scheme to better attribute global rewards to individual agent contributions. While these methods provide strong foundations, they often assume a fixed number of agents and can be sensitive to the reward structure. Our work adopts the CTDE principle within a MAPPO framework but focuses on adapting to dynamic environments with a novel reward structure that explicitly promotes cooperation, informed by an optimal global assignment rather than learned communication.

Advances in RL, especially deep reinforcement learning, have changed to swarm robotics and pursuit-evasion games. Researchers have successfully applied DRL to teach swarms complex behaviours [9] and to solve pursuit-evasion problems in various settings [11], [12]. For instance, [2] uses DRL for a heterogeneous unmanned aerial vehicle (UAV) swarm to pursue multiple targets. While this type of work demonstrates the feasibility of DRL for pursuit, they often rely on simpler reward schemes or do not explicitly tackle the problem of optimal, dynamic task allocation among pursuers. In contrast, our work integrates a global, optimal task assignment mechanism via the Hungarian algorithm directly into the learning loop, which provides a coordinated signal that complements the multi-component reward, leading to more efficient team strategies. We also draw inspiration from classical robotics approaches that use visibility graphs and centralised planning [6], but translate the high-level goal of coordinated coverage into a learnable, decentralised policy.

III. Problem Formulation and Methodology

This section formalises the multi-agent, multi-object pursuit task and details our proposed MAPPO-based framework, which integrates dynamic task allocation and a multi-component reward structure.

A. Problem Formulation

We model the multi-agent pursuit problem as a Decentralised Markov Decision Process (Dec-MDP), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, P, r, N, \gamma \rangle$:

- N is the number of pursuer agents.
- $\mathcal{S}$ is the global state space. Each state s includes positions and unique identities of N agents and M active objects.
- $\mathcal{A} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is the joint action space, where $\mathcal{A}_i$ denotes the set of discrete actions of each agent i.
- $P(s'|s, \mathbf{a})$ is the state transition probability of moving from state s to s' given the joint action $\mathbf{a} = (a_1, \ldots, a_N) \in \mathcal{A}$. Agents move deterministically based on their actions, objects move randomly. In the event of a collision with a boundary or another agent, the acting agent's position remains unchanged for that time step.
- $r \triangleq r(s, \mathbf{a})$ is the shared team reward function.
- $\gamma \in [0, 1)$ is the discount factor, which balances immediate and future rewards.

The objective is to find a joint policy $\boldsymbol{\pi}(\mathbf{a}|s) = \prod_{i=1}^{N} \pi(a_i|s)$ that maximizes the expected return. To formalise this, we first define the **return**, G_t, as the total discounted reward from time-step t until the episode terminates at $T_{\max}$ [24, p.57]:

$$G_t = \sum_{k=t+1}^{T_{\max}} \gamma^{k-t-1} r(s_k, \mathbf{a}_k). \tag{1}$$

The **state-value function** is the expected return when starting in state s and following policy $\boldsymbol{\pi}$ thereafter:

$$V^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\boldsymbol{\pi}} \left[G_t | s_t = s \right], \tag{2}$$

where $\mathbb{E}_{\boldsymbol{\pi}}[\cdot]$ denotes the expectation under the policy $\boldsymbol{\pi}$.

Similarly, the **action-value function** is the expected return after taking action $\mathbf{a}$ in state s and then following policy $\boldsymbol{\pi}$:

$$Q^{\boldsymbol{\pi}}(s, \mathbf{a}) = \mathbb{E}_{\boldsymbol{\pi}} \left[G_t | s_t = s, \mathbf{a}_t = \mathbf{a} \right]. \tag{3}$$

Thus, we have $V^{\boldsymbol{\pi}}(s) = \mathbb{E}_{\mathbf{a} \sim \boldsymbol{\pi}(\cdot|s)}[Q^{\boldsymbol{\pi}}(s, \mathbf{a})]$, i.e., the state-value function $V^{\boldsymbol{\pi}}(s)$ is the expectation of the action-value function $Q^{\boldsymbol{\pi}}(s, \mathbf{a})$ under policy $\boldsymbol{\pi}$.

The advantage function, assessing whether an action is better or worse than the policy's default behaviour, thereby identifying actions that lead to better-than-expected outcomes, is given by:

$$A^{\boldsymbol{\pi}}(s, \mathbf{a}) = Q^{\boldsymbol{\pi}}(s, \mathbf{a}) - V^{\boldsymbol{\pi}}(s). \tag{4}$$

B. MAPPO with Cooperative Task Allocation

We propose a solution based on Multi-Agent Proximal Policy Optimisation (MAPPO), a robust *on-policy*[1] algorithm that adheres to the Centralised Training with Decentralised Execution (CTDE) paradigm.

1) MAPPO with a Centralised Critic: Our approach employs a standard MAPPO framework where each agent i maintains a dedicated actor network[2] $\pi_{\theta_i}(a_i|s)$ to make decentralised decisions. The joint policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{a}|s) = \prod_{i=1}^{N} \pi_{\theta_i}(a_i|s)$ is parameterised by the set of all actor parameters, $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_N\}$. To update the policy, agent trajectories are sampled using the pre-updated (old) parameters $\boldsymbol{\theta}_{-}$.

[1] An on-policy algorithm updates its strategy using only experience collected from its most recent policy.

[2] A neural network is required to approximate the policy function, enabling generalization in state spaces too large for tabular methods.

A single, centralised critic network $V^{\phi}(s)$ approximates the state-value function (defined in (2)). It is trained by updating its parameters, ϕ, to minimise the difference between its prediction, $V^{\phi}(s)$, and the observed empirical return, G_t (defined in (1)). The loss function for training the critic is the mean squared error between its predictions and these empirical returns:

$$L^{VF}(\phi) = \hat{\mathbb{E}}_{s_t,\mathbf{a}_t}\left[(V^{\phi}(s_t) - G_t)^2\right], \tag{5}$$

where $\hat{\mathbb{E}}_{s_t,\mathbf{a}_t} = \mathbb{E}_{s_t\sim\upsilon^{\boldsymbol{\pi}_{\boldsymbol{\theta}_-}},\ \mathbf{a}_t\sim\boldsymbol{\pi}_{\boldsymbol{\theta}_-}(\cdot|s_t)}$, is an empirical average operator with the stationary distribution $\upsilon^{\boldsymbol{\pi}}(s) \propto \sum_{t=0}^{T_{\max}} \gamma^t \Pr(s|\boldsymbol{\pi})$;

The joint policy $\boldsymbol{\pi}_{\boldsymbol{\theta}}$ is updated by maximising the PPO clipped surrogate objective [25], which prevents overly aggressive policy changes and ensure the learning process is reliable:

$$\begin{aligned} &L^{CLIP}(\boldsymbol{\theta}) \\ &= \hat{\mathbb{E}}_{s_t,\mathbf{a}_t}\left[\min\left(\vartheta_t(\boldsymbol{\theta})\hat{A}_t, \text{clip}(\vartheta_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \end{aligned} \tag{6}$$

where $\vartheta_t(\boldsymbol{\theta}) = \dfrac{\boldsymbol{\pi}_{\boldsymbol{\theta}}(\mathbf{a}_t|s_t)}{\boldsymbol{\pi}_{\boldsymbol{\theta}_-}(\mathbf{a}_t|s_t)}$ is the probability ratio, measuring the change in likelihood of an action between the new and old policies. $\hat{A}_t$ is the estimated advantage function from (4), computed using Generalised Advantage Estimation (GAE) [26]. The $\text{clip}(\vartheta_t(\boldsymbol{\theta}), 1-\epsilon, 1+\epsilon)$ function constrains the ratio $\vartheta_t(\boldsymbol{\theta})$ to the interval $[1-\epsilon, 1+\epsilon]$, thereby preventing overly large policy updates and ensuring training stability. ϵ is a hyperparameter that defines the clipping range, which restricts the extent of the policy update.

By maximising L^{CLIP}, the policy is driven to prioritise actions with a high, positive advantage $\hat{A}_t$. According to the Policy Improvement Theorem [24, p.78], favouring these actions results in a better overall policy.

2) Dynamic Task Allocation via Optimal Assignment: A key challenge in multi-agent pursuit is determining "who pursues whom." A greedy approach, where each agent targets its nearest object, often results in suboptimal team behaviour, such as multiple agents chasing the same object. To mitigate this, we propose a dynamic, cooperative task allocation mechanism.

At each time step, we address an unbalanced optimal assignment problem, accounting for unequal numbers of agents N and objects M. We construct a cost matrix $C \in \mathbb{R}^{N\times M}$, with each element C_{ij} representing the Manhattan distance between agent i and object j. The objective is to minimize the total cost through a one-to-one assignment, formalised as finding a binary matrix $B \in \{0,1\}^{N\times M}$ that solves:

$$\min_{B} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} C_{ij}B_{ij} \tag{7}$$

$$\text{subject to} \quad \sum_{j=1}^{M} B_{ij} \le 1, \quad \forall i \in \{1,\ldots,N\}, \tag{8}$$

$$\sum_{i=1}^{N} B_{ij} \le 1, \quad \forall j \in \{1,\ldots,M\}. \tag{9}$$

The constraints ensure that each agent and object is assigned at most once. While this linear optimal assignment problem can be solved using various methods (e.g., the Auction algorithm or Integer Linear Programming), we employ the classic Hungarian algorithm for its robust simplicity.

To ensure all agents remain engaged, especially when $N > M$, unassigned agents are paired with their closest object, ensuring each has an assigned object even if shared. This assignment indirectly shapes the reward function, providing a globally aware coordinated signal for learning without directly controlling actions.

3) Multi-Component Reward Shaping: To guide agents towards efficient and collaborative capture, a multi-reward function is designed. The reward for an individual agent i at time step t, denoted r^i_t, is a sum of several components. The overall team reward is the sum of these individual rewards, $r_t = \sum_{i=1}^{N} r^i_t$. The individual reward is formulated as:

$$r^i = r^i_{\text{capture}} + r^i_{\text{prox}} + r^i_{\text{explore}} + r^i_{\text{coll}} + r^i_{\text{step}} + r^i_{\text{bound}} + r^i_{\text{osc}}.$$

The key components are defined as follows:

- The **capture reward**, $r^i_{\text{capture}} = 50 \cdot 1^i_c \cdot \cdot f_{\text{time}}(t)$, provides a large, time-scaled incentive for capturing an object. 1^i_c denotes the indicator function of whether agent i captures an object ($1^i_c = 1$) or not ($1^i_c = 0$). The time-decay factor $f_{\text{time}}(t) = (1 - t/T_{max})^2$ ensures that earlier captures are rewarded more heavily, promoting speed.
- The **proximity incentive**, $r^i_{\text{prox}} = 0.3 \cdot (d^i_{t-1} - d^i_t)$, rewards an agent for reducing the distance to its assigned object from the Hungarian algorithm in aforementioned subsection. This encourages purposeful movement aligned with the team's global strategy.
- A set of **behavioural penalties and bonuses** refines agent behaviour, including a step penalty ($r_{\text{step}} = -0.1$) for efficiency, a heavy collision penalty ($r^i_{\text{coll}} = -10$) for spatial awareness, and a small exploration bonus ($r^i_{\text{explore}} = 0.2$) to prevent stagnation. Additional minor penalties discourage hitting boundaries ($r^i_{\text{bound}} = -0.2$) and oscillatory movement ($r^i_{\text{osc}} = -0.1$).

IV. Experiments

We conduct a series of experiments to evaluate the performance of our proposed method against relevant baselines and perform an ablation study to analyse the contribution of its key components.

A. Experimental Setup

Environment: The experiments are run in a 2D grid world of size 10×10 called "*Spiders and Flies*" as in Fig. 1, adopted from [14]. At the start of each episode, a specified number of spiders (pursuers) and flies (objects) are placed randomly. At each discrete time step, spiders choose an action (move up, left, down, or right), and flies execute a random move to an adjacent cell. An episode concludes when all flies are captured or a maximum of 500 time steps is reached. A fly is considered captured if a spider moves onto its cell.

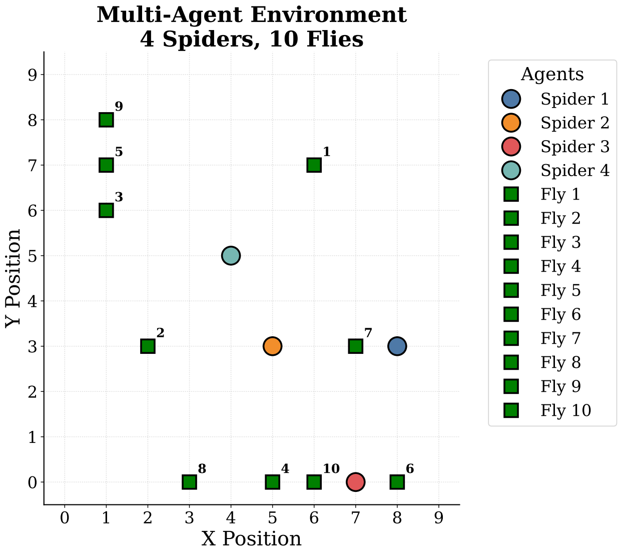

Fig. 1. Multi-Agent 2D Discrete Environment with 4 spiders to catch 10 flies.

Scenarios: We consider several challenging scenarios with the number of flies is increased from 1 to 10, and the number of spiders is increased from 1 to 4.

Metrics: We evaluate performance based on two key metrics:

i) **Average Episode Length:** The number of time steps to complete an episode, used as a measure of efficiency.

ii) **Average Planning Time:** The average time (in milliseconds) required to compute control actions for all spiders per step.

Notably, we omit the capture rate usually used in other search-and-pursuit literature since the proposed multi-component reward function yields 100% of capture rates for all considered methods. All results are computed over 100 Monte Carlo runs.

B. Baselines

We compare our method against two strong and relevant baselines, chosen to benchmark our cooperative, policy-based approach against fundamentally different paradigms.

(i) **DQN:** An independent learning approach where each spider is a Deep Q-Network agent [8]. This baseline is chosen to highlight the benefits of coordinated learning over independent decision-making. Agents learn individually using their own local observations and do not explicitly coordinate or share information.

(ii) **Rollout:** A planning-based method without deep learning [14]. At each step, it uses a simple heuristic-based policy (move towards the closest fly) and simulates future outcomes to a depth of 15 steps to select the best action. This serves as a strong model-based opponent to contrast learning-based approaches.

C. Implementation Details

Our method and the baselines are implemented in PyTorch. The MAPPO agent is trained for $1,000,000$ total environment steps with an adaptive stopping mechanism that terminates training early if the averaged episode reward has not improved for more than $100,000$ steps. Key hyperparameters for our MAPPO agent include a learning rate of 3×10^{-4}, a PPO clipping parameter $\epsilon = 0.2$, discount factor $\gamma = 0.99$, and GAE $\lambda = 0.95$. For each update cycle, we collect 2048 steps of experience and train for 10 epochs with a minibatch size of 64. All experiments were conducted on a single NVIDIA A6000 GPU.

D. Main Results

We compare the performance of our proposed method (MAPPO w/ Hungarian) against the DQN and Rollout baselines in Fig. 2. The results clearly demonstrate the superiority of our MAPPO approach, particularly as the number of agents and objects increases.

Efficiency (Average Steps): As shown in the top row of Fig. 2, our MAPPO approach consistently requires the fewest steps to capture all flies. The performance gap widens significantly in scenarios with more spiders (e.g., 3 or 4 spiders), where effective coordination becomes paramount. This suggests that the combination of centralised training and the Hungarian-informed reward signal allows agents to avoid redundant actions (e.g., multiple spiders pursuing the same fly) and distribute themselves more effectively than the uncoordinated DQN agents. While it is expected that episode length increases with more flies, our method shows a much flatter curve than the baselines, highlighting its superior efficiency and scalability.

Scalability (Planning Time): The bottom row of Fig. 2 reveals a critical advantage of learning-based approaches. The planning time for both DQN and our MAPPO method remains constant and negligible, as it only requires a single forward pass through the neural networks. In contrast, the planning time for the model-based Rollout algorithm increases linearly with both the number of spiders and flies. This is due to the nature of its online planning, which must simulate outcomes for each agent, making it computationally prohibitive for larger-scale problems. This demonstrates the practical advantage of our method for real-world deployment in complex, many-agent scenarios.

E. Ablation Study

To validate the contributions of our core components—the cooperative task assignment and the shaped reward function—we conduct an ablation study. We focus on a challenging scenario with 4 spiders and 10 flies to test for scalability and coordination under pressure. We compare our full method against two variants:

(i) **Closest Object:** Our MAPPO method, but with the Hungarian algorithm replaced by a simple greedy heuristic where each spider is rewarded for moving towards its closest fly.

(ii) **Baseline Reward:** Our MAPPO method (using the greedy "Closest Object" assignment) with the multi-component reward function replaced by a simple baseline reward (-1 for every step, as proposed in [14]).

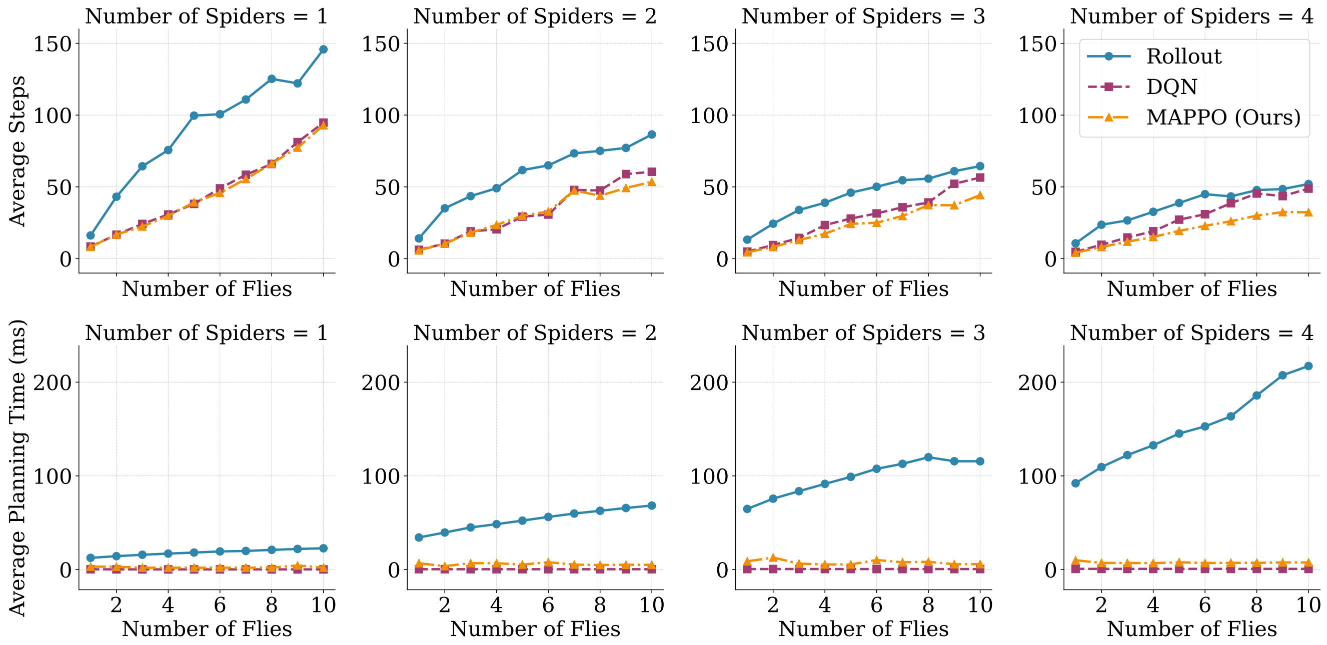

Fig. 2. Performance comparison between our proposed MAPPO algorithms versus two baselines: DQN and Rollout for the Spiders and Flies 2D environment. The number of flies is increased from 1 to 10, and the number of spiders is increased from 1 to 4. All results are computed over 100 Monte Carlo runs.

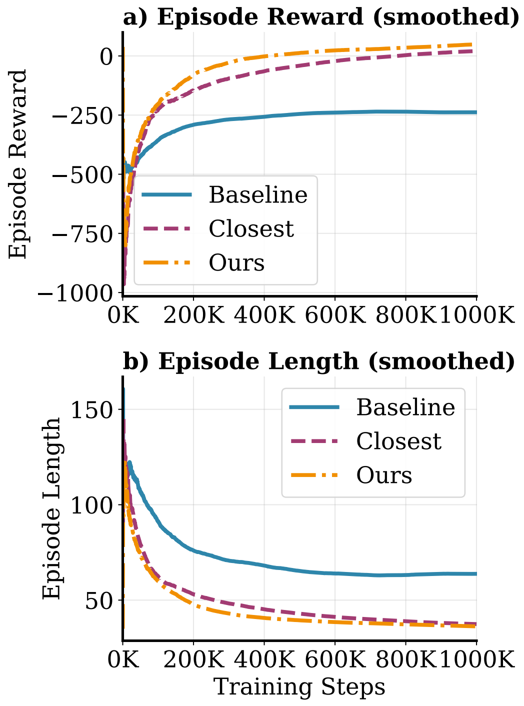

Fig. 3. Ablation study for the challenging scenario of 4 spiders to catch 10 flies in a 2D environment for comparing our MAPPO algorithms with two baselines: **Closest Object** and **Baseline Reward**.

TABLE I
ABLATION STUDY RESULTS FOR 4 SPIDERS AND 10 FLIES

Method	**Avg. Reward (↑)**	**Episode Length (↓)**
Baseline Reward	-235.7	64.5
Closest Object	35.0	36.5
Our Method	**55.7**	**35.9**

Higher average reward (↑) indicates better performance, while shorter episode length (↓) reflects greater efficiency. Bold text highlights the best-performing method.

Since the planning time is negligible for MAPPO, we replace it with another metric called **Average Reward (Avg. Reward)**, i.e., the mean cumulative reward per episode, reflecting overall task performance.

Fig. 3 and Table I depict the results of the ablation study, demonstrating the significant contributions of both our “Shaped Reward” function as well as the proposed cooperative task allocation method. The learning curves in Fig. 3 reveal that the “Baseline Reward” agent learns extremely slowly and plateaus at a significantly lower reward level, confirming that the sparse reward signal is insufficient for learning a complex, coordinated policy. Our multi-component “Shaped Reward” provides a dense and informative signal that is crucial for efficient learning for controlling agents in complex and coordinated tasks.

Furthermore, comparing the proposed method (“Ours”) with the “Closest Object” variant underscores the importance of the optimal assignment. While the greedy “Closest Object” strategy is a significant improvement over the baseline, our method achieves a notably higher final reward and lower episode length. This confirms that the globally optimal assignment provides a superior coordination signal, preventing the suboptimal emergent behaviour of agents pursuing the same object and leading to a more effective overall team strategy. The full model’s superior performance over both ablations validates our design choices.

V. Conclusion

In this paper, we propose a novel multi-agent deep reinforcement learning framework for pursuing and capturing multiple dynamic objects. Our approach enhances the MAPPO algorithm by integrating an adaptive state representation for varying numbers of agents and objects, a dynamic task allocation mechanism using the Hungarian algorithm for cooperative object assignment, and a multi-component reward function to promote efficient, coordinated behaviour. Experimental results demonstrated that our method significantly outperforms both independent learners (DQN) and planning-based (Rollout) baselines in complex scenarios. Furthermore, ablation studies confirmed that the cooperative assignment strategy and the shaped reward structure are both critical for achieving superior performance.

A key limitation of this work is its evaluation within a discrete grid-world featuring simplistic, random object behaviour. While this provides a controlled test-bed, transitioning to continuous state and action spaces for real-world applications presents a considerable challenge that could increase computational complexity. However, a core component of our method—the Hungarian-based reward shaping with adaptive state representation—is readily generalisable, as it depends only on agent-object distances, irrespective of whether the domain is discrete or continuous.

Future work might explore several research directions: enhancing robustness under noisy conditions like imperfect sensor data, extending the framework to handle *unknown* and time-varying numbers of objects using estimation techniques like Random Finite Set (RFS) theory [27], and testing against sophisticated object behaviours, including adversarial or evasive strategies, to create a more realistic and challenging test-bed for cooperative pursuit.

Acknowledgment

A large language model has been used for checking grammar and enhancing English language usage.

References

[1] T. H. Chung, G. A. Hollinger, and V. Isler, "Search and pursuit-evasion in mobile robotics: A survey," *Autonomous robots*, vol. 31, pp. 299–316, 2011.

[2] M. Kouzeghar, Y. Song, M. Meghjani, and R. Bouffanais, "Multi-target pursuit by a decentralized heterogeneous UAV swarm using deep multi-agent reinforcement learning," in *Proc. of ICRA*, 2023, pp. 3289–3295.

[3] Y. Hou, J. Zhao, R. Zhang, X. Cheng, and L. Yang, "UAV swarm cooperative target search: A multi-agent reinforcement learning approach," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 568–578, 2023.

[4] H. V. Nguyen, B.-N. Vo, B.-T. Vo, H. Rezatofighi, and D. C. Ranasinghe, "Multi-objective multi-agent planning for discovering and tracking multiple mobile objects," *IEEE Transactions on Signal Processing*, 2024.

[5] J. Li, M. Li, Y. Li, L. Dou, and Z. Wang, "Coordinated multi-robot target hunting based on extended cooperative game," in *Proc. of ICRA*, 2015, pp. 216–221.

[6] B. P. Gerkey, S. Thrun, and G. Gordon, "Visibility-based pursuit-evasion with limited field of view," *The International Journal of Robotics Research*, vol. 25, no. 4, pp. 299–315, 2006.

[7] I. Chades, B. Scherrer, and F. Charpillet, "A heuristic approach for solving decentralized-pomdp: Assessment on the pursuit problem," in *Proc. of SAC*, 2002, pp. 57–62.

[8] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

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

[10] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. of NeurIPS*, 2017, pp. 6382–6393.

[11] C. De Souza, R. Newbury, A. Cosgun, P. Castillo, B. Vidolov, and D. Kulić, "Decentralized multi-agent pursuit using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4552–4559, 2021.

[12] Z. Zhang, D. Zhang, Q. Zhang, W. Pan, and T. Hu, "DACOOP-A: Decentralized adaptive cooperative pursuit via attention," *IEEE Robotics and Automation Letters*, vol. 9, no. 6, pp. 5504–5511, 2023.

[13] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. of ICML*, 1993.

[14] D. Bertsekas, "Multiagent reinforcement learning: Rollout and policy iteration," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 249–272, 2021.

[15] D. S. Bernstein, R. Givan, N. Immerman, and S. Zilberstein, "The complexity of decentralized control of Markov decision processes," *Math. of Op. Research*, vol. 27, no. 4, pp. 819–840, 2002.

[16] F. Christianos, L. Schäfer, and S. Albrecht, "Shared experience actor-critic for multi-agent reinforcement learning," *Proc. of NeurIPS*, vol. 33, pp. 10 707–10 717, 2020.

[17] C. Li, T. Wang, C. Wu, Q. Zhao, J. Yang, and C. Zhang, "Celebrating diversity in shared multi-agent reinforcement learning," *Proc. of NeurIPS*, vol. 34, pp. 3991–4002, 2021.

[18] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. I. WU, "The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games," in *Proc. of NeurIPS*, vol. 35. Curran Associates, Inc., 2022, pp. 24 611–24 624.

[19] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, and T. Graepel, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proc. of AAMAS*, 2018, p. 2085–2087.

[20] T. Rashid, M. Samvelyan, C. S. De Witt, G. Farquhar, J. Foerster, and S. Whiteson, "Monotonic value function factorisation for deep multi-agent reinforcement learning," *Journal of Machine Learning Research*, vol. 21, no. 178, pp. 1–51, 2020.

[21] J. Wang, Z. Ren, T. Liu, Y. Yu, and C. Zhang, "QPLEX: Duplex dueling multi-agent q-learning," in *Proc. of ICLR*, 2021.

[22] A. Das, T. Gervet, J. Romoff, D. Batra, D. Parikh, M. Rabbat, and J. Pineau, "Tarmac: Targeted multi-agent communication," in *Proc. of ICML*. PMLR, 2019, pp. 1538–1546.

[23] T. Rashid, G. Farquhar, B. Peng, and S. Whiteson, "Weighted QMIX: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning," *Proc. of NeurIPS*, vol. 33, pp. 10 199–10 210, 2020.

[24] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA and London, England: MIT Press, 2018.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[26] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. of ICLR*, 2016.

[27] B.-N. Vo, B.-T. Vo, and M. Beard, "Multi-sensor multi-object tracking with the generalized labeled multi-bernoulli filter," *IEEE Transactions on Signal Processing*, vol. 67, no. 23, pp. 5952–5967, 2019.