

Article

An Experience-Guided MAPPO Framework for Multi-UAV Cooperative Tracking in Continuous Action Spaces

Hao Xiong, Minghu Tan, Xiaoyu Li and Haoyu Li * 

School of Astronautics, Northwestern Polytechnical University, Xi'an 710072, China; haox@mail.nwpu.edu.cn (H.X.); mhtan@mail.nwpu.edu.cn (M.T.)

* Correspondence: haoyuli@nwpu.edu.cn

Highlights

What are the main findings?

- A novel experience-guided multi-agent proximal policy optimization (E-MAPPO) method is proposed to improve the training efficiency of multi-UAV cooperative guidance.
- A Markov decision process (MDP) framework is constructed to overcome the challenges of large action spaces and difficult convergence in multi-agent reinforcement learning.

What is the implication of the main finding?

- The E-MAPPO algorithm is used to devise a cooperative strategy, enhancing the overall cooperative effectiveness of multiple UAVs in civilian tasks such as collaborative search and rescue and environmental monitoring.
- The designed cooperative strategy exhibits excellent robustness and generalization capabilities, ensuring its reliable execution in complex dynamic environments.

Abstract

A cooperative guidance law based on the experience-guided multi-agent proximal policy optimization (E-MAPPO) algorithm is proposed for multiple unmanned aerial vehicles (UAVs) to track dynamic points of interest in civilian applications, such as collaborative search and rescue and environmental monitoring. In multi-UAV cooperative tracking, accurate arrival-time coordination is important for improving collaborative task execution, but it remains challenging because of continuous action spaces, target maneuvering, uncertain time-to-go estimation, and inefficient exploration in multi-agent reinforcement learning. Specifically, a multi-UAV cooperative guidance environment is formulated, and the problem is modeled as a Markov decision process. To address the challenges of large action spaces and poor convergence in multi-agent reinforcement learning, an experience-guided MAPPO framework is introduced to enhance training efficiency and policy stability. Different from standard MAPPO, the proposed E-MAPPO introduces proportional-navigation-guided experience only during the early training stage to guide exploration, while the final policy is still optimized through the MAPPO objective. Subsequently, a composite reward function is designed by integrating distance-based heuristic terms with auxiliary guidance signals, thereby improving exploration efficiency and facilitating coordinated rendezvous and tracking of dynamic references. Comparative simulations with cooperative proportional navigation guidance (CPNG), sliding mode control (SMC), and standard MAPPO are conducted under different target motion scenarios. The results show that E-MAPPO reduces the average convergence step by 17.07% compared with MAPPO. In the straight-moving target scenario, E-MAPPO reduces the cooperative time error by 55.10% compared with CPNG and by 8.33% compared with MAPPO. In the S-type maneuvering



Academic Editor: Pablo Rodríguez-Gonzálvez

Received: 25 May 2026

Revised: 26 June 2026

Accepted: 29 June 2026

Published: 30 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

target scenario, E-MAPPO reduces the cooperative time error by 55.81% compared with CPNG and by 9.52% compared with MAPPO. Monte Carlo experiments further verify its effectiveness and robustness. Additional robustness tests under Gaussian measurement noise, observation bias, and communication delay show that the proposed method maintains acceptable tracking accuracy and cooperative timing performance under different uncertainty conditions. In addition, the results indicate that the proposed method generalizes well to different types of maneuvering targets.

Keywords: uncontrollable velocity; cooperative guidance; MAPPO; reinforcement learning

1. Introduction

Recently, the operational effectiveness of flying vehicle swarms has been significantly enhanced, enabling the execution of complex tasks that cannot be accomplished by a single flying vehicle operating independently [1,2]. This development has motivated the concept of cooperative missions in civilian applications, such as collaborative search and rescue and environmental monitoring, in which multiple flying vehicles coordinate to track a dynamic point of interest simultaneously [3–5]. Meanwhile, targets increasingly exhibit intelligence and autonomous decision-making capabilities, posing substantial challenges to conventional navigation strategies. To overcome these issues, intelligent cooperative navigation strategies have become a research focus. In particular, integrating learning-based methods into guidance law design can significantly enhance UAV performance in multi-agent coordination and trajectory optimization under uncertain and dynamic environments [6].

In general, navigation strategies designed for multi-UAV simultaneous tracking are typically divided into two primary types: impact time control guidance (ITCG), which is implemented for each individual vehicle, and cooperative guidance, which relies on the cooperation of arrival times and communication between flying vehicles [7]. To achieve simultaneous tracking of the target, ITCG was proposed [8] and then a number of ITCG laws were developed based on the Proportional Navigation Guidance (PNG) law [9,10], optimal control theory [11,12], sliding mode control [13,14], and polynomial trajectory shaping approach [15,16]. The difficulty in the practical application of ITCG is the requirement to pre-assign a common arrival time. Due to constraints on the speed variation and limited flight performance, not all desired arrival times are feasible, and infeasible assignments may lead to failure of simultaneous tracking. In contrast, cooperative navigation strategies achieve real-time arrival-time synchronization through local interactions among flying vehicles via a communication network. For instance, in [17], a distributed cooperative guidance law was proposed to achieve simultaneous tracking under fixed and switching communication topologies. In [18], the cooperative simultaneous tracking was then transformed to a consensus problem by using a cyclic digraph, and solved with a fixed finite-time consensus algorithm. In [19], a three-dimensional (3D) target-tracking model was reformulated as a linear second-order dynamical system, enabling simultaneous tracking via consensus-based control. This framework was further extended in [20] to address cooperative guidance under communication delays. Moreover, to track maneuvering targets, Wang et al. [21] proposed a robust 3D cooperative guidance strategy, guiding the trajectory to align with the yaw plane.

Extensive research has been conducted on cooperative navigation strategies for both stationary and maneuvering targets. In particular, several distributed cooperative guidance strategies have been proposed for stationary target tracking. For example, a fixed-time

distributed cooperative guidance strategy was proposed in H Yu [22]. To adjust the flying vehicle speed in the line-of-sight direction, a guidance law was proposed in [23], so that the flying vehicle with longer remaining flight time accelerates while the flying vehicle with shorter remaining flight time decelerates. Chen et al. [24] proposed a 3D cooperative guidance strategy considering field-of-view constraints, consisting of a cooperative phase based on feedback linearization and velocity-constrained consensus, followed by a PN phase for rendezvous while avoiding singularities. Utilizing coupled nonlinear tracking kinematics and the event-triggered strategy, Sinha et al. [25] developed two 3D cooperative navigation strategies for simultaneous tracking of stationary targets. For maneuvering targets, cooperative navigation strategies are typically designed under angle constraints, where LOS errors converge via consensus theory and sliding mode control [26]. The cooperative guidance methods proposed in [27,28] can simultaneously track the target under angle constraints. The game theory was employed to achieve simultaneous tracking of the maneuvering targets [29]. Z Zhang [30] proposed a unified cooperative guidance law for both stationary and maneuvering targets by combining proportional navigation with coordination terms to ensure simultaneous tracking. These methods effectively address simultaneous tracking problems and maintain performance under communication loss. However, challenges remain, including cooperative guidance design under velocity constraints and inaccuracies in time-to-go estimation, which limit precision and robustness in dynamic environments and continuous trajectory planning [31].

Driven by the continuous progress in artificial intelligence, intelligent navigation strategies have increasingly been developed by using neural networks, fuzzy logic, and reinforcement learning, etc. Among these approaches, reinforcement learning serves as an essential mechanism to improve how UAVs adapt and execute decisions across challenging and dynamic environments [32], particularly for multi-agent cooperative path planning, searching, and following in 3D scenarios [33]. For instance, navigation strategies based on reinforcement learning are used to improve guidance efficiency, allowing agents to generate effective policies when facing complex environments [34–37]. These approaches are still constrained by their dependence on the traditional guidance law framework. J Guo [38] proposed a comprehensive framework for the multi-flying vehicle and multi-target tracking challenges. This framework utilizes reinforcement learning to optimize target assignments, ensuring that UAVs can track targets effectively and accurately. In the context of intelligent cooperative active tracking scenarios, W Ni [39] proposed a smart guidance policy to coordinate the movement of both the flying vehicles and the target under conditions of limited prior knowledge and observational noise. In addition, W H Zhou [40] proposed a distributed deep reinforcement learning approach to achieve coordinated tracking in UAV swarms, which is based on maximum mutual reward learning, using a multi-agent actor-critic framework to learn collaborative policies among homogeneous UAVs. Despite the proven capability of reinforcement learning for intelligent guidance design, its application to multi-UAV cooperative guidance remains relatively limited.

Although reinforcement learning has been increasingly introduced into multi-agent cooperative guidance problems, the differences among standard MAPPO, imitation learning, curriculum learning, demonstration-based reinforcement learning, and the proposed E-MAPPO still need to be clarified. Standard MAPPO mainly relies on random policy exploration and does not explicitly introduce guidance experience, which may lead to low exploration efficiency in the early training stage. Imitation learning usually trains a policy to reproduce expert actions, while curriculum learning improves training by gradually changing task difficulty. Demonstration-based reinforcement learning often depends on fixed offline expert datasets or expert demonstrations. In contrast, the proposed E-MAPPO algorithm introduces PN-guided experience only during the early stage of MAPPO training.

The guidance actions are generated online according to the current UAV-target relative motion state, and they are used to improve early-stage exploration rather than to make the policy simply imitate the expert. After the early guided stage, the agents interact with the environment only according to the policy network, and the final policy is still optimized through the MAPPO objective. To provide a clearer comparison, the main differences between the proposed E-MAPPO and existing related methods are summarized in Table 1.

Table 1. Comparison between E-MAPPO and related learning-based methods.

Method	Difference from E-MAPPO
MAPPO	Lacks PN-guided experience for early exploration
Imitation learning	Focuses on action imitation rather than reward-driven policy optimization
Curriculum learning	Changes task difficulty instead of introducing trajectory-level guidance
Demonstration-based RL	Relies on offline demonstrations, while E-MAPPO generates PN-guided actions online
E-MAPPO	Uses online PN-guided experience to improve early exploration while preserving MAPPO optimization

Although observer-based fault-tolerant consensus control is not the focus of this paper, it is also relevant to the reliability of multi-agent cooperative systems. For example, recent work on LPV multi-agent systems designed fault-tolerant observers and virtual actuators to achieve leader-following consensus control under actuator faults [41]. Different from these model-based fault-tolerant control methods, this paper focuses on experience-guided reinforcement learning for multi-UAV cooperative tracking in continuous action spaces.

The main contributions of this paper are summarized as follows. First, a multi-UAV cooperative tracking environment with continuous action spaces is constructed for maneuvering target tracking. Second, an experience-guided MAPPO algorithm, named E-MAPPO, is proposed, in which PN-guided experience is introduced only in the early training stage to improve exploration efficiency and convergence performance. Third, a composite reward function is designed to guide cooperative tracking and arrival-time coordination among multiple UAVs. Finally, comparative simulations and statistical analyses are conducted under different target motion scenarios to verify the effectiveness of the proposed method.

2. Problem Statement

2.1. Planar Relative Motion Model for UAV-Target Tracking

To simplify the problem, the coupling between orthogonal motion components is neglected, and the UAV-target relative motion is described in a planar two-dimensional framework [42]. This simplification helps focus on the key cooperative guidance issue considered in this study, namely arrival-time coordination among multiple UAVs in continuous action spaces. Nevertheless, the two-dimensional assumption ignores altitude variation, climb-rate constraints, and coupled lateral-vertical dynamics, which may limit its direct applicability to realistic three-dimensional UAV scenarios. For extension to three-dimensional space, the relative-motion model should further include azimuth and elevation line-of-sight angles, altitude-related relative states, and two acceleration commands in the lateral and vertical channels. The corresponding planar motion dynamics are illustrated in Figure 1.

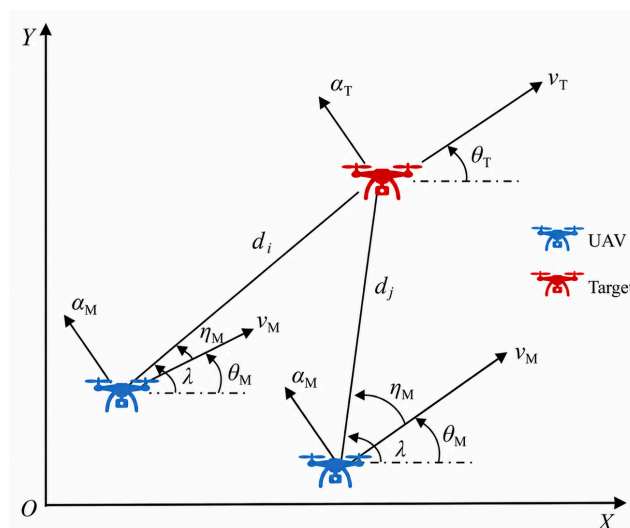


Figure 1. UAV Tracking Scenario Schematic.

As shown in Figure 1, v_M and v_T denote the velocities of the UAV and the target, respectively, while a_M and a_T represent their normal accelerations. The line-of-sight (LOS) angle between the UAV and the target is denoted by λ , and $\dot{\lambda}$ represents its rate of change. The flight-path angles of the UAV and the target are defined as θ_M and θ_T , respectively. The lead angle of the UAV is denoted by η_M . In addition, d_i and d_j denote the relative distance.

Suppose the UAV operates at a uniform speed with an adjustable velocity heading. According to planar kinematic principles, the tracking models for the UAV and the target are formulated as:

$$\begin{cases} \dot{x} = v \cos \theta \\ \dot{y} = v \sin \theta \\ \dot{\theta} = a/v \end{cases} \quad (1)$$

where x and y denote the planar position coordinates, v is the speed magnitude, a represents the normal acceleration perpendicular to the velocity direction, and θ denotes the flight-path angle. Given that the UAV and the target share identical motion models, a UAV–target tracking scenario is established. The corresponding relative kinematic equations are defined as:

$$\begin{cases} r = \sqrt{(x_T - x_M)^2 + (y_T - y_M)^2} \\ \lambda = \arctan\left(\frac{y_T - y_M}{x_T - x_M}\right) \\ \dot{r} = -v_T \cos(\lambda - \theta_T) - v_M \cos(\lambda - \theta_M) \\ r \dot{\lambda} = v_M \sin(\lambda - \theta_M) - v_T \sin(\lambda - \theta_T) \end{cases} \quad (2)$$

2.2. Multi-Agent Reinforcement Learning Formulation

The task of multi-UAV dynamic cooperative guidance is modeled as a Markov decision process (MDP). The MDP mainly includes four elements: state space, action space, state transition probability, and reward function. It is assumed that each UAV can obtain the necessary relative motion information with respect to the target, including relative position, velocity, and line-of-sight information. Through information interaction among multiple UAVs, each UAV can also obtain cooperative information related to other UAVs. Based on this MDP formulation, the state space, action space, and reward function are defined in this section, while the network architecture and the detailed E-MAPPO training process are presented in Section 3.

2.2.1. State Space Design

In multi-agent reinforcement learning algorithms, actions are computed by the policy network based on the agent's local observations. The selection of observation variables directly affects the convergence efficiency of the algorithm. Therefore, the selected observations must include essential UAV-target tracking information to ensure that the agent can make decisions based on the current situation. Consequently, the observations for the UAV are defined as follows:

$$o_i = [r, \dot{r}, \lambda, \dot{\lambda}, t_{go}] \quad (3)$$

In multi-agent reinforcement learning algorithms, normalizing the observation space is an effective optimization strategy. Each observation value in the normalized state space is represented as a dimensionless ratio, reflecting the relative change between the current state and the initial state. The processed observation space is defined as follows:

$$o_i = \left[\frac{r}{r_0}, \frac{\dot{r}}{\dot{r}_0}, \frac{\lambda}{\lambda_0}, \frac{\dot{\lambda}}{\dot{\lambda}_0}, \frac{\bar{t}_{go}}{\bar{t}_{go0}} \right] \quad (4)$$

where t_{go} denotes the estimated time-to-go and is calculated online according to the current relative motion state. Specifically, t_{go} is estimated from the current relative distance, UAV velocity, and line-of-sight geometry as:

$$t_{go} = \frac{r}{v_M} \left[1 + \frac{\theta_M^2}{2(2N+1)} \right] \quad (5)$$

where N is the navigation constant. During both training and execution, this value is recalculated at each time step using the latest observation information. For maneuvering targets, t_{go} is an approximate online estimate rather than an exact prediction of the future arrival time, because the target motion may vary during the engagement. In this study, it is used as an auxiliary coordination variable to help the agents reduce arrival-time differences, and its effectiveness is evaluated through the cooperative timing performance in the simulation results.

The construction of the global state space is directly realized through connecting the observation space information from each UAV. Specifically, the construction process of the global state space is as follows:

$$S = [o_i, o_j], i \neq j \quad (6)$$

2.2.2. Action Space Design

In this paper, the normal acceleration of the UAV is directly controlled. Since the MAPPO algorithm can handle continuous action spaces, continuous action commands are employed as the action space, providing the agent with a larger exploration space. Therefore, the action space is designed below:

$$a_M = [a_i, a_j], i \neq j \text{ and } a_{i,j} \in [-40g, 40g] \quad (7)$$

2.2.3. Reward Function Design

The reward function aims to guide the agent to take appropriate actions, and its design directly affects the convergence performance of the reinforcement learning algorithm. In this paper, a series of composite reward functions are designed to guide the agent to

consider not only individual performance but also the overall cooperative effect. The reward function designed in this paper is as follows:

$$R_{dist} = \begin{cases} \sum_{i,j} (k_1 - d_i(t)/k_2), & t = 0 \\ \sum_{i,j} [d_i(t-1) - d_i(t)/k_2], & t > 0 \end{cases} \quad (8)$$

where k_1 and k_2 are parameters, and $d_i(t)$, $d_j(t)$ represent the distances from the target to UAV i and j , respectively. This part of the reward encourages the UAV to approach the target.

The terminal discrete reward formulation is expressed below:

$$R_{ter} = \begin{cases} 0, & \text{otherwise} \\ k_3, & d_{i,j}(t) \leq d_z \end{cases} \quad (9)$$

where k_3 is a parameter, and d_z is the miss distance of the UAV. Consequently, when both agents track the target simultaneously, each will receive a fixed terminal reward.

The time-to-go error penalty reward is defined as follows:

$$R_{time} = \begin{cases} k_4 \times |\bar{t}_{go}|, & d_{i,j}(t) > d_z \\ k_5, & \min(d_{i,j}(t)) \leq d_z \text{ and } \max(d_{i,j}(t)) > d_z \end{cases} \quad (10)$$

where k_4 and k_5 serve as negative parameters, and $\bar{t}_{go}$ denotes the time-to-go error of the UAV. When a single UAV tracks the target, a penalty of size k_5 is given. This design takes into account the arrival time error between UAVs, ensuring that the UAVs can track the target at the same time point or a similar time point.

The overall reward function is as follows:

$$r = R_{dist} + R_{ter} + R_{time} \quad (11)$$

The main challenges of this problem are threefold. First, the maneuvering motion of the target makes the time-to-go estimation uncertain, which increases the difficulty of cooperative arrival-time coordination. Second, the actions of different UAVs are coupled through the cooperative timing objective, and independent optimization of each UAV may not lead to satisfactory cooperative behavior. Third, the continuous action space and sparse terminal tracking information make pure reinforcement learning difficult to converge efficiently. These challenges motivate the introduction of an experience-guided multi-agent reinforcement learning method.

3. Formulation of the 2D Cooperative Guidance Framework

3.1. Multi-Agent Proximal Policy Optimization

Reinforcement learning (RL) is a training methodology where an agent learns decision-making policies through interactions with its environment. Among existing RL methods, the multi-agent proximal policy optimization (MAPPO) algorithm has emerged as an effective framework for multi-agent systems. MAPPO extends proximal policy optimization (PPO) and is well suited for continuous action spaces. Sharing a fundamental actor-critic framework with PPO, it relies on the policy network for action generation and the value network for steering policy updates. The defining feature of the MAPPO algorithm is its centralized value function, which leverages global information, including the states of other agents and environmental context, to improve learning efficiency. The algorithm updates policies via a proximal objective that constrains the magnitude of policy changes,

thereby enhancing training stability and sample efficiency [43]. In this study, a continuous action space is considered. The optimization objective is to maximize the expected return while limiting excessive policy updates through the clipped surrogate objective, which contributes to stable convergence [44].

During training, MAPPO samples historical transitions from an experience buffer to compute the loss functions and update network parameters via gradient-based optimization. θ and ϕ are the parameters of the policy and value networks, in which π_θ indicates the policy. The corresponding loss functions are formulated below:

$$\mathcal{L}(\theta) = \left[\frac{1}{B_n} \sum_{i=1}^B \sum_{k=1}^n \min(r_{\theta,i}^{(k)} A_i^{(k)}, \text{clip}(r_{\theta,i}^{(k)}, 1 - \varepsilon, 1 + \varepsilon) A_i^{(k)}) \right] + \sigma \frac{1}{B_n} \sum_{i=1}^B \sum_{k=1}^n S[\pi_\theta(o_i^{(k)})] \quad (12)$$

$$\mathcal{L}(\phi) = \frac{1}{B_n} \sum_{i=1}^B \sum_{k=1}^n (V_\phi(s_i^{(k)}) - \hat{R}_i)^2 \quad (13)$$

In the above formulation, $r_{\theta,i}^{(k)} = \frac{\pi_\theta(a_i^{(k)}|o_i^{(k)})}{\pi_{\theta_{\text{old}}}(a_i^{(k)}|o_i^{(k)})}$ denotes the probability ratio between the updated and previous policies. The advantage term $A_i^{(k)}$ is estimated using the generalized advantage estimation (GAE) algorithm. The term S represents the policy entropy, which is defined as the Shannon entropy of the action distribution. Since a continuous Gaussian policy is adopted in this study, this term corresponds to the differential entropy of the Gaussian action distribution and is used to encourage policy exploration, with σ denoting the corresponding entropy coefficient. B is the batch size, n is the number of UAVs, and $\hat{R}_i$ denotes the discounted return. The clipping operator $\text{clip}(\cdot)$ constrains the policy update by limiting the variation in the probability ratio, thereby preventing excessively large updates. This mechanism improves training stability and facilitates convergence to a more optimal policy.

3.2. Improvement of the E-MAPPO Algorithm

In the E-MAPPO algorithm, trajectory experience generated by the proportional navigation (PN) guidance law is integrated into the training process to enhance learning efficiency. Specifically, the current UAV overload is first calculated according to the PN guidance law, and this overload value is then mapped to the interval $[-1, 1]$. Subsequently, the observational data obtained based on the PN guidance are input into the policy network, which outputs a Gaussian distribution. According to this Gaussian distribution and the mapping relationship of the overload, the probability of the current action policy calculated by PNG can be obtained. Following this, the current environmental state is input into the value network to calculate the value of the current state and acquire the corresponding reward value. Trajectory data based on the proportional navigation are stored in the experience replay buffer. Since the training process requires the agent to explore independently, the amount of trajectory data based on the proportional navigation is set to $1/a$ of the entire batch size. During the initial $1/b$ phase of the total training steps, these experience data are mainly used to guide the agent's learning process. In this study, $a = 4$ and $b = 6$. Therefore, PN-guided trajectories are introduced only when the number of stored episodes in the replay buffer is no larger than one quarter of the batch size and the current training steps are within the first one sixth of the maximum training steps. Under this condition, the PN guidance law generates a reference overload command, which is mapped to the normalized action interval and then used to guide the policy network in action selection. The corresponding transition data generated during this guided interaction are stored in

the same replay buffer together with the trajectories generated by autonomous exploration. When either the batch-ratio condition or the training-stage condition is not satisfied, the PN-guided action is no longer injected, and the agents interact with the environment only according to the actions sampled from the policy network. After the replay buffer accumulates one batch of episodes, the actor and critic networks are updated, and the replay buffer is reset for the next round of data collection. As the agents gradually become capable of independently identifying the correct exploration direction, the dependence on these experiences is progressively reduced, thereby allowing the agents to learn relying on their own exploration. Figure 2 illustrates the schematic diagram of trajectory storage in the replay buffer of the E-MAPPO algorithm.

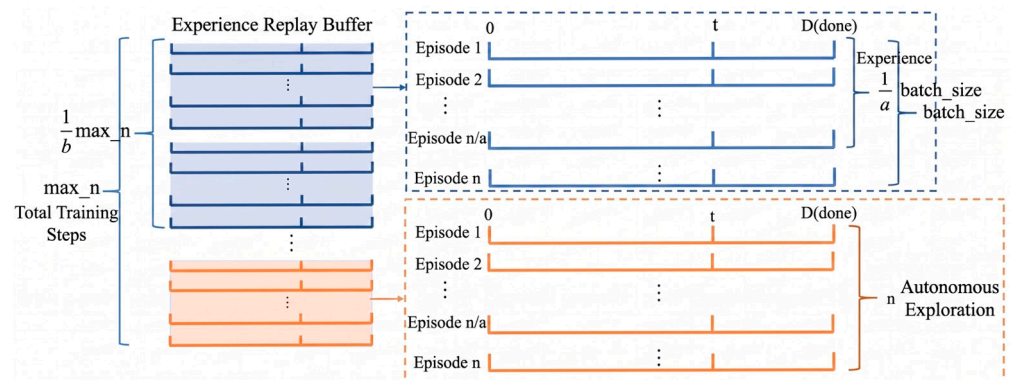


Figure 2. Schematic of trajectory storage in the experience replay buffer based on the E-MAPPO algorithm.

3.3. Network Architecture Design

The MAPPO algorithm mainly includes two networks: the policy network and the global value network. The policy network is responsible for receiving local observations as input and outputting actions. Each agent has its own policy network. These networks can be independent or shared among all agents. In this paper, the policy networks of the agents are independent. The output of the policy network is a probability distribution of actions, based on which the agent can select an action. The value network receives the global state information. The output of the value network is a value function, which is used to evaluate the quality of the actions output by the policy network, i.e., to evaluate the action value under the current policy.

The policy network is composed of three layers: an input layer, a hidden layer, and two output layers, where n is the number of agents. Specifically, each policy network is implemented as a fully connected multilayer perceptron. For each UAV, the local observation with a dimension of 5 is used as the input of the policy network. The input layer is connected to a fully connected hidden layer with 128 neurons, followed by a Tanh activation function. The two output layers generate the mean and variance of the Gaussian action distribution, respectively. Subsequently, specific actions can be obtained by sampling from this Gaussian distribution. The Gaussian distribution is adopted because the normal acceleration command is continuous, and a Gaussian policy can provide smooth stochastic exploration while allowing the action probability to be conveniently calculated for the PPO policy-ratio update. Given that the dimension of the joint observation space is $5n$ and the dimension of the joint action space is n , the input layer of the policy network is configured with 5 neurons to match the local observation dimension, and the output layer is configured with 1 neuron to accommodate the local action dimension. The hidden layer is configured with 128 hidden units.

The value network provides feedback to assist the policy network in learning an optimal policy. Through the global state, it can evaluate the state value for the policy. The

value network shares a similar structure with the policy network. Specifically, the value network is also implemented as a fully connected multilayer perceptron. Different from the policy network, the value network takes the global state as its input. The hidden layer adopts a fully connected neural network structure, and the Tanh function is selected as the activation function for all layers to process the output. The global state value is obtained by concatenating the observations of each agent, so the value network's input layer contains $n \times 5$ neurons, its output layer has 1 neuron, and the hidden layer contains 128 neurons.

3.4. Overall Architecture and Training Procedure of E-MAPPO

The structure of the E-MAPPO algorithm proposed in this paper is illustrated in Figure 3. In the initial phase of training, the expert trajectories generated by the PN guidance law are pre-loaded into the experience replay buffer, so that these data can participate in the agent's learning process, and these experience data are used to guide the agent's learning process. As the agents gradually become capable of independently discovering the correct exploration direction, the reliance on these expert experiences is reduced, allowing the agents to learn through their own exploration. During the guidance process, each UAV is equipped with its own policy network, which determines the action policy based on its respective local observations. Simultaneously, all UAVs share a centralized value network, which is used to evaluate and perceive the state of the entire environment.

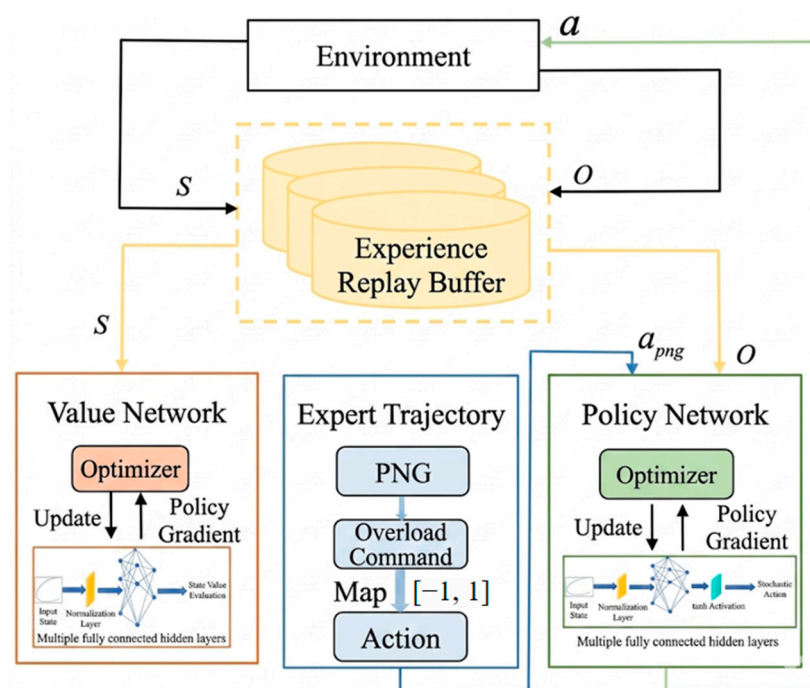


Figure 3. E-MAPPO algorithm architecture.

During the training process, the trajectory data based on PN under the initial UAV-target state is first collected, followed by the agents' autonomous exploration. In Figure 3, o denotes the joint observation composed of the local observations of all UAVs, and S denotes the global state constructed by concatenating these local observations. Specifically, the agent initially obtains the local observation o_t at the current time step, and selects and executes an action a_t according to the policy network. The evaluation value S_t of the current situation is output by the value network based on the current environmental state. Subsequently, the agent enters a new environmental state S_{t+1} according to the transition probability $P(S_{t+1}|S_t, a_t)$ and obtains the reward R_{t+1} . The above trajectory data is stored in the experience replay buffer D . Following this, after obtaining the environmental state

S_{t+1} and reward R_{t+1} , the policy model continues to obtain a new policy. Throughout the entire process, the policy model is continuously optimized and adjusted according to the magnitude of the reward until the set conditions are satisfied, and the interaction between the agent and the environment model ends. To further illustrate the training process of the proposed method, the E-MAPPO algorithm training flowchart is shown in Figure 4.

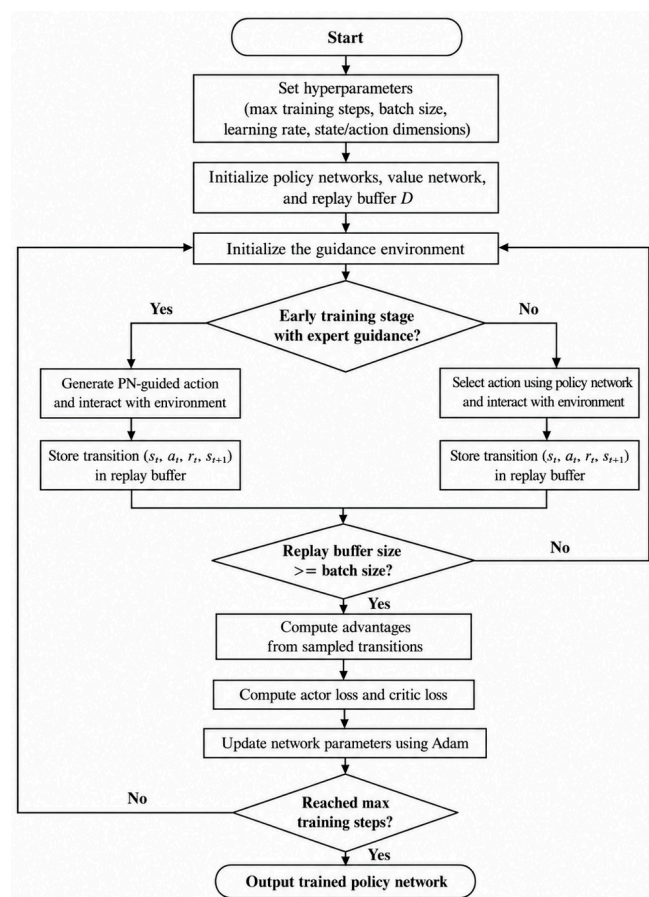


Figure 4. E-MAPPO algorithm training flowchart.

4. Numerical Results and Analysis

All numerical simulations presented in this study are programmed in Python 3.13.0 and hosted within the PyCharm Community 2024.2.4. The underlying hardware responsible for executing these experiments consists of an Intel(R) Xeon(R) Gold 5118 CPU, 64 GB of RAM, and a dedicated NVIDIA GeForce RTX 3090 GPU.

In order to validate the performance of the guidance law proposed in this paper, simulation experiments are designed for a target in the form of S-type cosine maneuver, and comparative experiments with other guidance laws are also designed. Furthermore, robustness and generalization analyses of the proposed guidance policy are also performed. The simulation step size of the experiments is set to 0.01 s.

4.1. Simulation Setup

A simulation scenario is designed in which UAVs track an S-type cosine maneuvering target. It is assumed that the initial position of the target is $[10,000, 11,000 \pm 250]$ m, executing a cosine maneuver with a velocity of 300 m/s, and an initial flight path angle of 0° . Three UAVs are deployed from distinct positions to cooperatively track the target. The UAVs are initialized with the following conditions, as listed in Table 2:

Table 2. Settings of the guidance environment.

Parameters	UAV 1	UAV 2	UAV 3
x-axis position (m)	0	2000	5000
y-axis position (m)	12,000	4000	0
velocity (m/s)	800	800	800
pitch angle (°)	0	75	100

The traditional algorithm compared in this paper is CPNG proposed in Ref. [45]. Specifically, a time error bias term is incorporated into the foundation of the proportional navigation guidance law, which is formulated as follows:

$$\begin{cases} a_{Mi} = \bar{N}_i V_{Mi} \dot{\theta}_{Li} \\ \bar{N}_i = N_i \{1 - k_{1i} \text{sgn}(\xi_i) - k_{2i} \xi_i\}, i \neq j \\ \xi_i = \sum_{j=1}^n c_{ij} (t_{goj} - t_{goi}) \end{cases} \quad (14)$$

where N is the navigation constant, and $N = 3$, $N_i = 3$, $k_{11} = 2.8$, $k_{12} = 2.5$, $k_{21} = k_{22} = 1.2$,

$$c_{ij} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

The guidance law proposed in Ref. [45] achieves control over the curvature of the UAV trajectory by establishing a mathematical model between the time-to-go and the navigation constant. This approach influences the flight trajectory by adjusting the navigation constant, and adjusts the time-to-go online in real time through network communication among UAVs, thereby achieving the goal of cooperatively rendezvous with the target.

To contrast the energy expenditure of different methods, the control energy E is defined as follows, where a denotes the UAV acceleration:

$$E = \int_{t_0}^t a^2(\tau) d\tau \quad (15)$$

4.2. Analysis of Experimental Results

The normal acceleration of the target is time-varying, and its expression is given as follows:

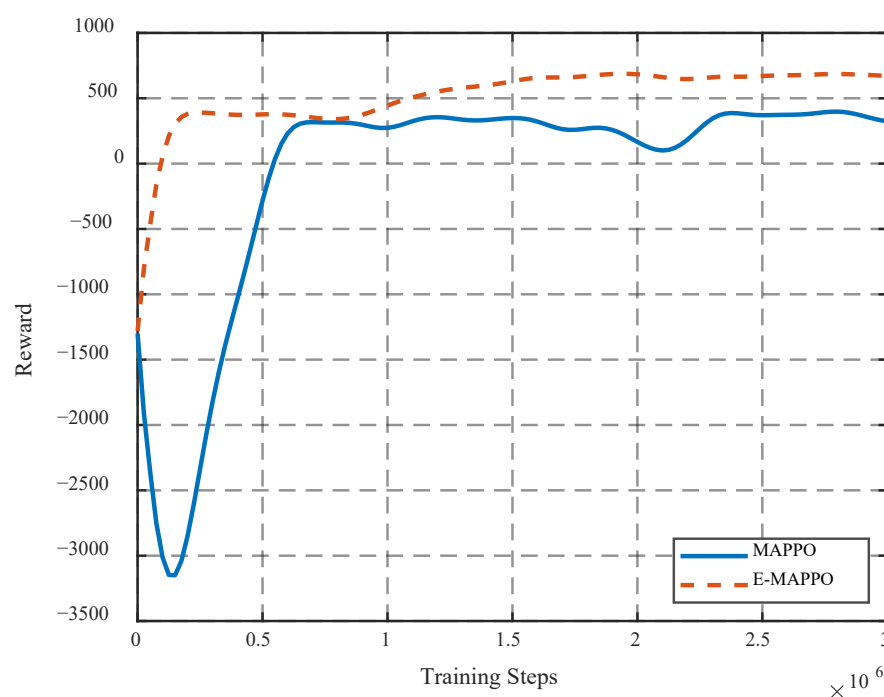
$$a = 5g \cos(\pi t / 2) \quad (16)$$

In this experiment, the parameters of the reward function for the guidance law are set as follows: $k_1 = 5$, $k_2 = 100$, $k_3 = 1$, $k_4 = -0.05$, and $k_5 = -0.05$. These coefficients were determined through preliminary empirical tuning according to the roles of different reward terms. In the first stage, k_1 , k_2 , and k_3 were adjusted mainly to ensure the basic tracking capability, so that the UAVs could approach and successfully track the dynamic point of interest. After stable tracking behavior was obtained, the timing-related coefficients k_4 and k_5 were introduced and adjusted to reduce the time-to-go error and improve the arrival-time coordination among multiple UAVs. During this tuning process, the convergence trend of the cumulative reward, the miss distance, and the cooperative time error were jointly considered. The final parameter setting was selected because it provided stable convergence and a reasonable balance between tracking accuracy and cooperative timing performance in the considered simulation scenarios. The hyperparameter settings of the E-MAPPO algorithm are summarized in Table 3.

Table 3. Hyperparameter settings.

Hyperparameter	Value
Maximum training steps	4.0×10^6
Maximum steps per episode	2500
Batch size	10
Mini-batch size	2
Number of hidden-layer neurons	128
Actor learning rate	7×10^{-4}
Critic learning rate	7×10^{-4}
Discount factor γ	0.99
GAE parameter λ	0.95
PPO clipping parameter ϵ	0.2
Update epochs K	10
Policy entropy coefficient	0.01

Furthermore, orthogonal initialization is adopted for the neural networks, and the actor and critic networks are updated using the Adam optimizer when the replay buffer reaches the batch size. When the training reaches the maximum number of training steps, the training process is terminated. The reward convergence curve obtained from training is shown in Figure 5.

**Figure 5.** Reward function convergence curve.

As can be observed from the reward convergence curve in Figure 5, the E-MAPPO algorithm demonstrates better convergence performance compared with the standard MAPPO algorithm. During the initial phase of training, while the standard MAPPO algorithm still requires extensive exploration, the E-MAPPO algorithm can determine a more effective exploration direction with the guidance of experience, which is reflected in the faster increase in the reward value. To further provide a quantitative comparison, multiple independent training runs with different random seeds are conducted, and the corresponding results are summarized in Table 4. The results are reported as mean $\pm$ standard deviation. The convergence steps of MAPPO and E-MAPPO are $439,286 \pm 105,926$ and

$364,286 \pm 114,434$, respectively. Compared with MAPPO, E-MAPPO reduces the average convergence step by 17.07%, indicating improved sample efficiency. The paired statistical test gives a p -value of 0.0495 for the convergence step, which indicates that the reduction in convergence step is statistically significant at the 95% confidence level. In addition, the final cumulative rewards of MAPPO and E-MAPPO are 494.095 ± 10.879 and 641.362 ± 14.906 , respectively, showing that E-MAPPO also achieves a higher final reward. Overall, these results indicate that the experience-guided mechanism can improve early-stage exploration efficiency, accelerate convergence, and enhance the final training performance.

Table 4. Quantitative comparison of convergence performance.

Method	Convergence Step	Final Cumulative Reward	Sample Efficiency Improvement
MAPPO	$439,286 \pm 105,926$	494.095 ± 10.879	
E-MAPPO	$364,286 \pm 114,434$	641.362 ± 14.906	17.07%

4.3. Comparative Analysis

To further evaluate the performance of the proposed E-MAPPO method, comparative simulations are conducted with several representative guidance and reinforcement-learning-based methods, including CPNG, SMC, MAPPO, and E-MAPPO. Two target motion scenarios are considered: a straight-moving target scenario and an S-type maneuvering target scenario. All methods are tested under the same initial conditions and guidance environment parameters as those described in Section 4.2. The quantitative comparison results are summarized in Tables 5 and 6, respectively. The results are reported as mean $\pm$ standard deviation over multiple simulation trials.

Table 5. Quantitative comparison under the straight-moving target scenario.

Method	Miss Distance (m)	Cooperative Time Error (s)	Control Energy
CPNG	3.44 ± 0.93	0.49 ± 0.06	$180,475 \pm 28,020$
SMC	3.23 ± 0.71	0.92 ± 0.06	$113,401 \pm 12,270$
MAPPO	5.35 ± 0.67	0.24 ± 0.12	$160,070 \pm 21,701$
E-MAPPO	3.35 ± 0.75	0.22 ± 0.03	$138,867 \pm 13,094$

Table 6. Quantitative comparison under the S-type maneuvering target scenario.

Method	Miss Distance (m)	Cooperative Time Error (s)	Control Energy
CPNG	3.19 ± 0.48	0.43 ± 0.06	$191,647 \pm 26,010$
SMC	3.25 ± 0.76	0.87 ± 0.05	$111,168 \pm 23,963$
MAPPO	4.28 ± 0.48	0.21 ± 0.05	$168,749 \pm 13,040$
E-MAPPO	2.31 ± 0.56	0.19 ± 0.03	$134,707 \pm 14,897$

For the straight-moving target scenario, as shown in Table 5, all methods can complete target tracking with relatively small miss distances. Compared with CPNG, the proposed E-MAPPO method reduces the cooperative time error from 0.49 s to 0.22 s, corresponding to a reduction of 55.10%, and decreases the control energy by 23.05%. Compared with MAPPO, E-MAPPO reduces the miss distance by 37.38%, the cooperative time error by 8.33%, and the control energy by 13.25%. Although the SMC-based method has lower control energy, its cooperative time error is significantly larger than that of E-MAPPO. This indicates that E-MAPPO achieves a better balance between tracking accuracy, cooperative timing accuracy, and control energy consumption.

For the S-type maneuvering target scenario, the comparison results are given in Table 6. Compared with CPNG, E-MAPPO reduces the miss distance by 27.59%, the cooperative time error by 55.81%, and the control energy by 29.71%. Compared with MAPPO, E-MAPPO reduces the miss distance by 46.03%, the cooperative time error by 9.52%, and the control energy by 20.17%. These results show that the proposed method can still maintain good cooperative tracking performance when the target performs an S-type maneuver. Similar to the straight-moving target case, the SMC-based method consumes less control energy, but its cooperative time error is much larger than that of E-MAPPO. Therefore, the proposed E-MAPPO method provides more balanced overall performance under different target motion scenarios.

Overall, the quantitative results show that E-MAPPO achieves lower cooperative time error than the compared methods in both target motion scenarios. Meanwhile, compared with MAPPO, E-MAPPO also reduces miss distance and control energy in both cases. These results indicate that the experience-guided mechanism improves cooperative timing accuracy and overall tracking performance while maintaining competitive control energy consumption.

4.4. Computational Complexity and Real-Time Feasibility

To evaluate the computational burden and real-time feasibility of the proposed method, the computational characteristics of different guidance methods are further analyzed. It should be noted that the training process of MAPPO and E-MAPPO is performed offline. During online execution, only the trained actor network is used to generate the acceleration command for each UAV, while the critic network, replay buffer, and policy update process are no longer required. Therefore, the online computational burden of E-MAPPO is much lower than its offline training burden.

For the proposed E-MAPPO algorithm, the main online computation comes from the forward propagation of the actor network. If the observation dimension of each UAV is denoted as d_o , the hidden-layer size is denoted as H , and the action dimension is denoted as d_a , the inference complexity of one UAV can be approximately expressed as:

$$O(d_o H + H d_a) \quad (17)$$

For N UAVs, the total online inference complexity is approximately:

$$O[N(d_o H + H d_a)] \quad (18)$$

Since each UAV only uses its own actor network during online execution, the inference process can be performed independently for different UAVs. Therefore, when the number of UAVs increases, the total online inference complexity increases approximately linearly with N . This linear scalability supports the applicability of the proposed method to multi-UAV cooperative tracking without requiring online centralized optimization or global trajectory replanning.

In this study, $d_o = 5$, $H = 128$, and $d_a = 1$. Therefore, the actor network used for online decision-making is relatively small. This indicates that the proposed method has feasible online computational burden after offline training and has potential for real-time execution on embedded UAV platforms with sufficient onboard computing resources.

As summarized in Table 7, CPNG and SMC have low online computational burden because they are analytical guidance methods. MAPPO and E-MAPPO require higher offline training cost, but their online execution only involves lightweight neural network inference. Compared with MAPPO, E-MAPPO introduces PN-guided experience only in the early training stage, so it does not increase the online computational burden. Therefore,

after offline training, the proposed E-MAPPO method is feasible for real-time execution and has potential for deployment on embedded UAV platforms with sufficient onboard computing resources.

Table 7. Computational complexity comparison of different methods.

Method	Offline Training Burden	Online Execution Burden	Real-Time Feasibility
CPNG	None	Low, analytical computation	High
SMC	None	Low, analytical control law	High
MAPPO	High, multi-agent policy training	Moderate, actor-network inference	Feasible after offline training
E-MAPPO	High, with early PN-guided data collection	Moderate, actor-network inference	Feasible after offline training

4.5. Robustness Analysis

To verify the robustness of the proposed method under uncertainty disturbances, Monte Carlo simulations are conducted for the S-type cosine maneuvering target. In addition to Gaussian measurement noise, two additional uncertainty conditions, namely observation bias and communication delay, are further considered. For the Gaussian-noise case, 25 dB Gaussian white noise is added to the UAV-target relative distance, relative velocity, and line-of-sight angle. For the observation-bias case, a bounded bias within the range of $([-0.05, 0.05])$ is added to the observation information. For the communication-delay case, a delay of 5 simulation steps, corresponding to 0.05 s, is introduced into the information transmission process. For each uncertainty condition, 1000 Monte Carlo simulation trials are performed. The initial states of the UAVs in the Monte Carlo experiments are listed in Table 2, and the statistical results are summarized in Table 8.

Table 8. Robustness evaluation under different uncertainty conditions.

Uncertainty Condition	Miss Distance (m)	Cooperative Time Error (s)	Success Rate (%)
Gaussian noise	3.44 ± 0.55	0.54 ± 0.03	100
Observation bias	3.35 ± 0.67	0.63 ± 0.12	100
Communication delay	5.35 ± 0.75	0.81 ± 0.03	96

As shown in Table 8, the values are reported as mean $\pm$ standard deviation over 1000 Monte Carlo simulation trials. The proposed method can still maintain acceptable tracking accuracy and cooperative timing performance under different uncertainty conditions. Under Gaussian measurement noise and observation bias, the success rate remains 100%, and the average miss distances are 3.44 m and 3.35 m, respectively. This indicates that the learned policy has a certain robustness to sensing noise and bounded observation bias. When a communication delay of 0.05 s is introduced, the miss distance and cooperative time error increase to 5.35 m and 0.81 s, respectively, and the success rate decreases to 96%. This result shows that communication delay has a more significant influence on cooperative tracking performance than measurement noise and observation bias. Nevertheless, the proposed method still maintains a relatively high success rate under delayed information transmission. Overall, these results indicate that E-MAPPO has a certain degree of robustness under measurement noise, observation bias, and communication delay, while more complex uncertainties such as packet loss, actuator faults, and modeling errors will be further investigated in future work.

4.6. Generalization Analysis

To verify the generalization capability of the cooperative guidance strategy proposed in this paper, the test scope of the cooperative guidance network originally designed for the S-type cosine maneuvering target is extended to different target maneuvering scenarios, including an S-type maneuvering target and a circular maneuvering target. In these generalization tests, the initial states of the UAVs are the same as those shown in Table 2. A total of 1000 simulation experiments are carried out for each target maneuvering scenario to evaluate the performance of the guidance strategy under different target motion patterns.

4.6.1. Scenario 1: S-Type Maneuvering Target

In this scenario, the generalization capability of the proposed method is evaluated under an S-type maneuvering target. The target performs a periodic lateral maneuver, and its acceleration is defined as:

$$a = 5g\cos(\pi t/5) \quad (19)$$

Compared with the original S-type maneuver $a = 5g\cos(\pi t/2)$, the maneuver considered here keeps the same acceleration amplitude but uses a longer period, thereby forming a different target motion pattern for generalization evaluation.

Table 9 presents the experimental results of 1000 simulations conducted for the S-type maneuvering target.

Table 9. Guidance results under the S-type maneuvering target scenario.

Metric	Value
Miss Distance of UAV 1 (m)	3.31 ± 0.70
Miss Distance of UAV 2 (m)	4.30 ± 0.80
Miss Distance of UAV 3 (m)	7.28 ± 1.39
Cooperative Time Error (s)	0.54 ± 0.03

From Table 9, it can be seen that the average cooperative timing error of the three UAVs is 0.54 ± 0.03 s, and the average miss distances are 3.31 ± 0.70 m, 4.30 ± 0.80 m, and 7.28 ± 1.39 m, respectively. This indicates that the proposed method can maintain small tracking errors and acceptable cooperative timing accuracy when facing an S-type maneuvering target.

For the test results of the S-type maneuvering target, one operating condition is randomly selected for analysis, and the experimental results are shown in Figure 6. Figure 6a,b show the UAV trajectories and relative range curves, respectively. It can be observed that the three UAVs can successfully approach the maneuvering target and complete cooperative tracking. As shown in Figure 6c, the time-to-go curves of the three UAVs gradually converge during the guidance process, indicating that the proposed method can coordinate the arrival times of multiple UAVs under target maneuvering conditions. Figure 6d presents the acceleration profiles of the UAVs. At the initial stage, the UAVs generate relatively large acceleration commands to adjust their trajectories. In the middle stage, the acceleration commands remain within a moderate range, while larger fluctuations appear near the terminal stage to reduce the cooperative arrival error. Overall, the results demonstrate that the proposed method has good generalization capability under the S-type maneuvering target scenario.

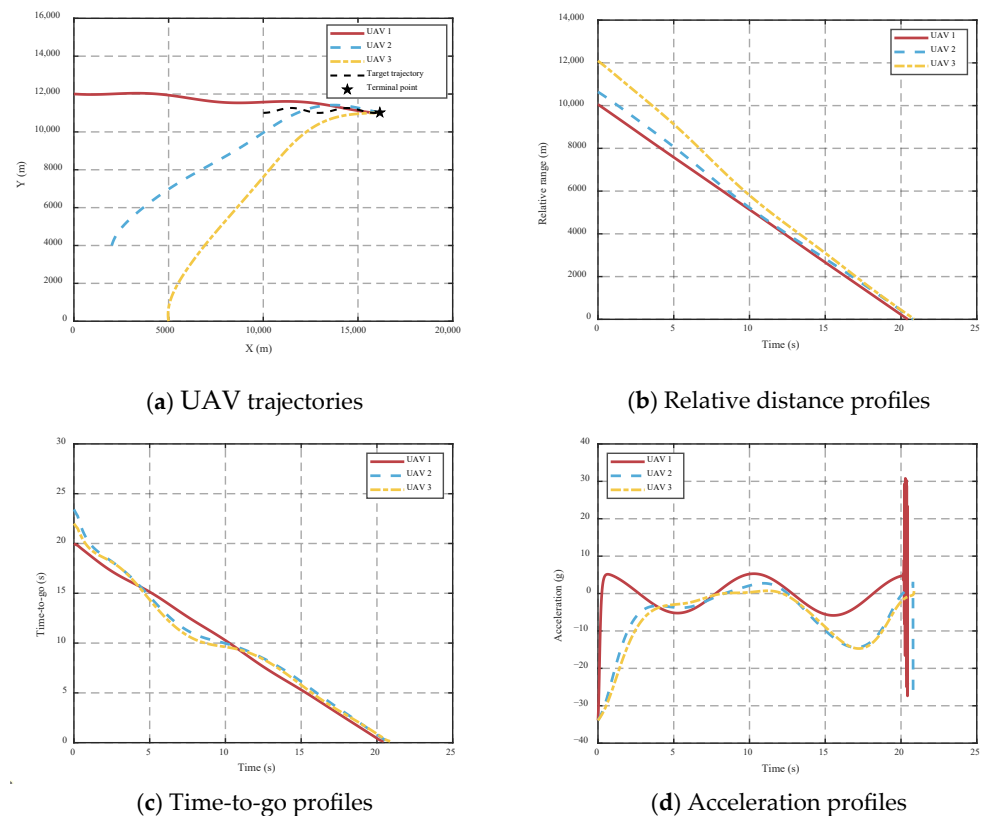


Figure 6. Test results for the S-type maneuvering target.

4.6.2. Scenario 2: Circular Maneuvering Target

In this scenario, the generalization capability of the proposed method is further evaluated under a circular maneuvering target, and its expression is given by:

$$a = g \quad (20)$$

where g denotes the gravitational acceleration. The acceleration direction is perpendicular to the target velocity, so that the target maintains a circular maneuver during the guidance process.

Table 10 presents the experimental results of 1000 simulations conducted for the circular maneuvering target.

Table 10. Guidance results under the circular maneuvering target scenario.

Metric	Value
Miss Distance of UAV 1 (m)	5.41 ± 0.75
Miss Distance of UAV 2 (m)	3.27 ± 0.73
Miss Distance of UAV 3 (m)	7.34 ± 1.47
Cooperative Time Error (s)	0.46 ± 0.04

From Table 10, it can be seen that the average cooperative timing error of the three UAVs is 0.46 ± 0.04 s, and the average miss distances are 5.41 ± 0.75 m, 3.27 ± 0.73 m, and 7.34 ± 1.47 m, respectively. This indicates that the proposed guidance law can still maintain relatively small timing error and acceptable guidance accuracy when facing a circular maneuvering target.

For the test results of the circular maneuvering target, one operating condition is randomly selected for analysis, and the experimental results are shown in Figure 7. Figure 7a,b show the UAV trajectories and relative range curves, respectively. It can be observed that

all UAVs successfully approach the circular maneuvering target, and the relative ranges gradually decrease to zero. As depicted by the time-to-go trends in Figure 7c, the time-to-go values of the three UAVs gradually converge during the guidance process, indicating that the proposed method can coordinate the arrival times of multiple UAVs under continuous target turning. Figure 7d presents the acceleration profiles of the UAVs. At the initial stage, the UAVs generate relatively large acceleration commands to adjust their flight trajectories. During the middle stage, the acceleration commands remain within a small range. Near the terminal stage, relatively large fluctuations appear in order to reduce the cooperative arrival error. Overall, the results demonstrate that the proposed method has good generalization capability under the circular maneuvering target scenario.

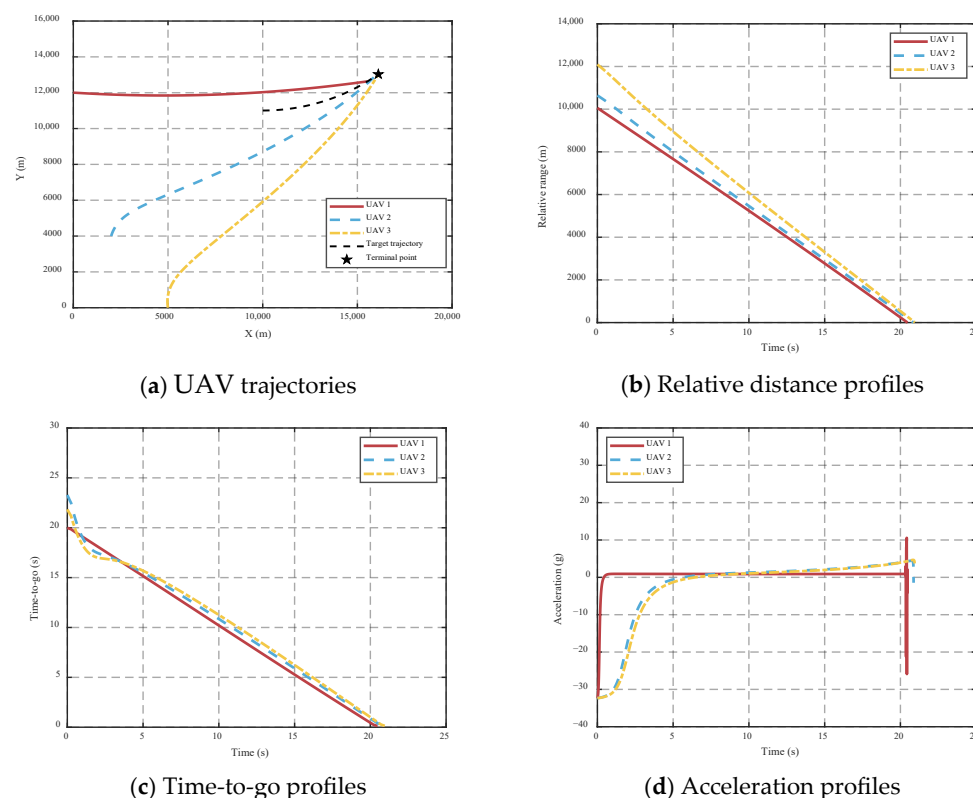


Figure 7. Test results for the circular maneuvering target.

4.7. Extension of the Action Space to Three Dimensions

To further illustrate how the proposed framework can be extended beyond the planar case, this subsection provides a preliminary three-dimensional analysis by expanding the action space from one acceleration command to two acceleration commands. In the two-dimensional formulation, each UAV outputs a one-dimensional continuous action corresponding to the normal acceleration in the guidance plane. In the three-dimensional case, the action of UAV i is extended to a two-dimensional continuous vector as:

$$a_i^{3D} = [a_{i,y}, a_{i,z}] \quad (21)$$

where $a_{i,y}$ and $a_{i,z}$ denote the acceleration commands of UAV i in the two maneuvering channels. Therefore, compared with the two-dimensional case, the main difference is that the policy network output is expanded from a one-dimensional continuous action to a two-dimensional continuous action. The overall E-MAPPO training framework remains unchanged, while the action dimension of each UAV is increased.

The initial conditions used in the three-dimensional extension scenario are listed in Table 11. The target starts from [10,000, 1400, 11,000] m and moves in a straight line at a speed of 300 m/s, while the UAVs start from different initial positions to verify the spatial coordination capability after action-space extension.

To examine the training process after extending the action space, the reward convergence curve is shown in Figure 8. The raw reward exhibits noticeable fluctuations during the exploration stage because the policy needs to search for suitable acceleration commands in two maneuvering channels. Nevertheless, the smoothed reward gradually increases and finally stabilizes at a relatively high level, indicating that the proposed E-MAPPO framework can still achieve stable learning after the action dimension is increased.

Table 11. Initial conditions for the three-dimensional extension scenario.

Parameters	UAV 1	UAV 2	UAV 3
x-axis position (m)	0	2000	4000
y-axis position (m)	1000	1000	1000
z-axis position (m)	12,000	3000	0
velocity (m/s)	800	800	800
pitch angle (°)	0	0	0
heading angle (°)	0	−45	−75

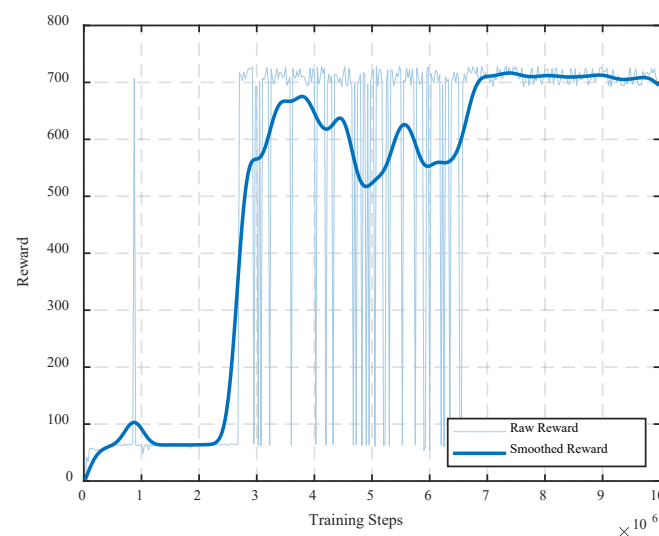


Figure 8. Reward function convergence curve for the three-dimensional extension.

A simple straight-moving target scenario is used for verification, and the corresponding results are shown in Figure 9. Figure 9a presents the three-dimensional trajectories of the UAVs and the target, where the three UAVs start from different spatial positions and gradually approach the target in three-dimensional space. Figure 9b shows the corresponding acceleration profiles, in which the upper subplot presents the acceleration commands in the y -direction and the lower subplot presents the acceleration commands in the z -direction. The results indicate that the UAVs can adjust their acceleration commands in two channels to complete spatial trajectory coordination, which demonstrates that the proposed method can be extended from planar cooperative guidance to a three-dimensional action-space setting. Since this subsection only considers a straight-moving target, more complex three-dimensional scenarios with target maneuvers will be further investigated in future work.

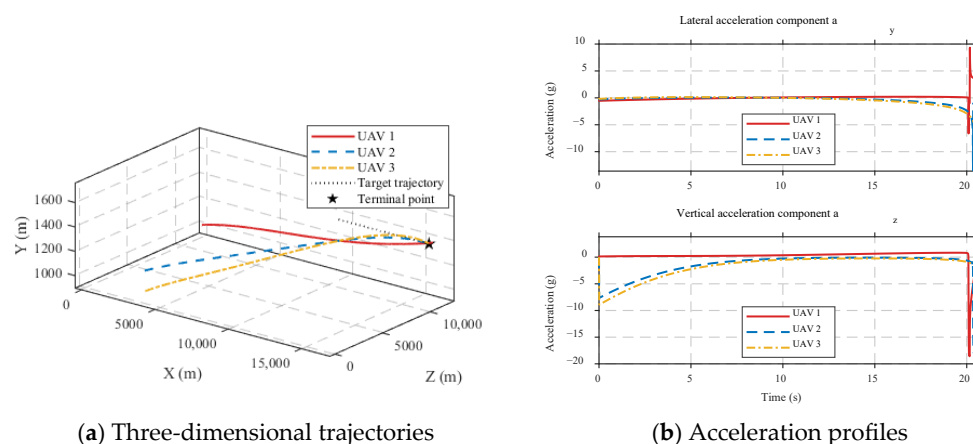


Figure 9. Preliminary three-dimensional extension results under a straight-moving target scenario.

5. Conclusions

In this paper, a cooperative guidance law based on the experience-guided MAPPO algorithm is designed for multi-UAV cooperative tracking of maneuvering targets. The neural network used in this guidance law can process the states of UAVs and output corresponding guidance commands. First, an experience-guided MAPPO (E-MAPPO) algorithm is proposed to address the convergence difficulties caused by multi-agent systems and continuous action spaces. Subsequently, a multi-UAV cooperative guidance environment is formulated, and a Markov decision process is constructed. Considering the performance requirements of cooperative guidance, the state space, action space, and reward function are carefully designed.

Simulation results verify the cooperative tracking capability of the proposed method against the S-type cosine maneuvering target. Although a certain timing error still exists during the cooperative guidance process, all UAVs can successfully track the target with relatively small miss distances. Furthermore, comparative experiments with CPNG, SMC, and standard MAPPO show that the proposed E-MAPPO method achieves smaller cooperative timing error and more balanced overall performance under different target motion scenarios. The convergence analysis also shows that the experience-guided mechanism can improve early-stage exploration efficiency and accelerate training convergence.

To further evaluate the applicability of the proposed guidance law, robustness and generalization tests are conducted. The results demonstrate that the proposed method can maintain acceptable tracking accuracy and cooperative timing performance under Gaussian measurement noise, observation bias, and communication delay. In addition, the trained policy can be extended to different target maneuvering scenarios, including S-type maneuvering and circular maneuvering targets. At present, research on cooperative navigation strategies based on reinforcement learning is still limited. This paper provides a feasible learning-based cooperative guidance framework by directly controlling UAV overload commands, and its effectiveness is verified through simulation experiments.

Future work will focus on the following aspects. First, the proposed E-MAPPO framework will be extended to more realistic three-dimensional cooperative tracking scenarios, including altitude variation, coupled lateral–vertical dynamics, and more complex target maneuvers. Second, additional uncertainties, such as packet loss, actuator faults, modeling errors, and communication constraints, will be further considered to improve the robustness of the proposed method. Third, the deployment of the trained policy on embedded UAV platforms will be investigated to further evaluate its real-time feasibility in practical applications.

Author Contributions: Conceptualization, M.T. and H.L.; methodology, X.L.; software, H.X.; validation, H.L.; formal analysis, H.X.; data curation, X.L.; writing—original draft preparation, H.X.; writing—review and editing, M.T. and H.L.; supervision, M.T.; project administration, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: Foundation of National Key Laboratory of Air-based Information Perception and Fusion and Aeronautical Science Foundation of China.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

DURC Statement: Current research is limited to the academic field of multi-agent reinforcement learning and cooperative guidance for unmanned aerial vehicles (UAVs), which is beneficial for non-offensive civilian applications such as collaborative search and rescue operations, large-scale environmental monitoring, and precision coordination in dynamic environments. Authors acknowledge the dual-use potential of the research involving multi-UAV cooperative tracking and time-synchronized rendezvous algorithms and confirm that all necessary precautions have been taken to prevent potential misuse. As an ethical responsibility, authors strictly adhere to relevant national and international laws about DURC. Authors advocate for responsible deployment, ethical considerations, regulatory compliance, and transparent reporting to mitigate misuse risks and foster beneficial outcomes.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zhang, G.M.Y.; Liu, A. Guidance law with impact time and impact angle constraints. *Chin. J. Aeronaut.* **2013**, *26*, 4960–4966. [\[CrossRef\]](#)
2. Zhao, J.; Zhou, R.; Jin, X. Progress in reentry trajectory planning for hypersonic vehicle. *J. Syst. Eng. Electron.* **2014**, *25*, 627–639. [\[CrossRef\]](#)
3. Balhance, N.; Weiss, M.; Shima, T. Cooperative guidance law for intrasalto tracking. *J. Guid. Control Dyn.* **2017**, *40*, 1441–1456. [\[CrossRef\]](#)
4. Zhao, Q.; Dong, X.; Song, X.; Ren, Z. Cooperative time-varying formation guidance for leader-following missiles to intercept a maneuvering target with switching topologies. *Nonlinear Dyn.* **2019**, *95*, 129–141.
5. Li, K.; Wang, J.; Lee, C.-H.; Zhou, R.; Zhao, S. Distributed cooperative guidance for multivehicle simultaneous arrival without numerical singularities. *J. Guid. Control Dyn.* **2020**, *43*, 1365–1373. [\[CrossRef\]](#)
6. Li, J.; Li, J.; Zhang, J.; Wenyue, M. A Comprehensive Review of Path-Planning Algorithms for Multi-UAV Swarms. *Drones* **2025**, *10*, 11. [\[CrossRef\]](#)
7. Dong, W.; Wang, C.; Wang, J.; Xin, M. Three-dimensional nonsingular cooperative guidance law with different field-of-view constraints. *J. Guid. Control Dyn.* **2021**, *44*, 2001–2015. [\[CrossRef\]](#)
8. Jeon, I.-S.; Lee, J.-I.; Tahk, M.-J. Impact-time-control guidance law for anti-ship missiles. *IEEE Trans. Control Syst. Technol.* **2006**, *14*, 260–266. [\[CrossRef\]](#)
9. Jeon, I.-S.; Lee, J.-I.; Tahk, M.-J. Impact-time-control guidance with generalized proportional navigation based on nonlinear formulation. *J. Guid. Control Dyn.* **2016**, *39*, 1885–1890. [\[CrossRef\]](#)
10. Guo, Y.; Li, X.; Zhang, H.; Cai, M.; He, F. Data-driven method for impact time control based on proportional navigation guidance. *J. Guid. Control Dyn.* **2020**, *43*, 955–966. [\[CrossRef\]](#)
11. He, S.; Lee, C.-H. Optimality of error dynamics in missile guidance problems. *J. Guid. Control Dyn.* **2018**, *41*, 1624–1633. [\[CrossRef\]](#)
12. Chen, X.; Wang, J. Optimal control based guidance law to control both impact time and impact angle. *Aerosp. Sci. Technol.* **2019**, *84*, 454–463. [\[CrossRef\]](#)
13. Hu, Q.; Han, T.; Xin, M. New impact time and angle guidance strategy via virtual target approach. *J. Guid. Control Dyn.* **2018**, *41*, 1755–1765. [\[CrossRef\]](#)
14. Sinha, A.; Kumar, S.R.; Mukherjee, D. Three-dimensional guidance with terminal time constraints for wide launch envelopes. *J. Guid. Control Dyn.* **2021**, *44*, 343–359. [\[CrossRef\]](#)
15. Tekin, R.; Erer, K.S.; Holzapfel, F. Polynomial shaping of the look angle for impact-time control. *J. Guid. Control Dyn.* **2017**, *40*, 2668–2673. [\[CrossRef\]](#)
16. Tekin, R.; Erer, K.S.; Holzapfel, F. Impact time control with generalized-polynomial range formulation. *J. Guid. Control Dyn.* **2018**, *41*, 1190–1195. [\[CrossRef\]](#)

17. Qilun, Z.; Xiwang, D.; Liang, Z.; Chen, B.; Jian, C.; Zhang, R. Distributed cooperative guidance for multiple missiles with fixed and switching communication topologies. *Chin. J. Aeronaut.* **2017**, *30*, 1570–1581. [[CrossRef](#)]
18. Kumar, S.R.; Mukherjee, D. Cooperative salvo guidance using finite-time consensus over directed cycles. *IEEE Trans. Aerosp. Electron. Syst.* **2019**, *56*, 1504–1514.
19. He, S.; Wang, W.; Lin, D.; Lei, H. Consensus-based two-stage salvo attack guidance. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *54*, 1555–1566.
20. He, S.; Kim, M.; Song, T.; Lin, D. Three-dimensional salvo attack guidance considering communication delay. *Aerosp. Sci. Technol.* **2018**, *73*, 1–9. [[CrossRef](#)]
21. Wang, C.; Ding, X.; Wang, J.; Shan, J. A robust three-dimensional cooperative guidance law against maneuvering target. *J. Frankl. Inst.* **2020**, *357*, 5735–5752. [[CrossRef](#)]
22. Yu, H.; Dai, K.; Li, H.; Zou, Y.; Ma, X.; Ma, S.; Zhang, H. Distributed cooperative guidance law for multiple missiles with input delay and topology switching. *J. Frankl. Inst.* **2021**, *358*, 9061–9085. [[CrossRef](#)]
23. Lyu, T.; Guo, Y.; Li, C.; Ma, G.; Zhang, H. Multiple missiles cooperative guidance with simultaneous attack requirement under directed topologies. *Aerosp. Sci. Technol.* **2019**, *89*, 100–110. [[CrossRef](#)]
24. Chen, Y.; Wang, J.; Wang, C.; Shan, J.; Xin, M. Three-dimensional cooperative homing guidance law with field-of-view constraint. *J. Guid. Control Dyn.* **2020**, *43*, 389–397. [[CrossRef](#)]
25. Sinha, A.; Kumar, S.R.; Mukherjee, D. Three-dimensional nonlinear cooperative salvo using event-triggered strategy. *J. Guid. Control Dyn.* **2021**, *44*, 328–342. [[CrossRef](#)]
26. Wang, Z.; Fang, Y.; Fu, W.; Wu, Z.; Wang, M. Cooperative guidance laws against highly maneuvering target with impact time and angle. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2022**, *235*, 1006–1016. [[CrossRef](#)]
27. Dong, W.; Wang, C.; Wang, J.; Zuo, Z.; Shan, J. Fixed-time terminal angle-constrained cooperative guidance law against maneuvering target. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *58*, 1352–1366. [[CrossRef](#)]
28. Wang, Z.; Fu, W.; Fang, Y.; Zhu, S.; Wu, Z.; Wang, M. Prescribed-time cooperative guidance law against maneuvering target based on leader-following strategy. *ISA Trans.* **2022**, *129*, 257–270. [[CrossRef](#)] [[PubMed](#)]
29. Tan, M.; Shen, H. Three-Dimensional Cooperative Game Guidance Law for a Leader-Follower System With Impact Angles Constraint. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *60*, 405–420. [[CrossRef](#)]
30. Zhang, Z.; Dong, X.; Yv, J.; Li, Q.; Ren, Z. Distributed cooperative tracking and cooperative guidance against maneuvering aerial target. *Aerosp. Sci. Technol.* **2024**, *144*, 108827. [[CrossRef](#)]
31. Tang, J.; Liang, Y.; Li, K. Dynamic scene path planning of UAVs based on deep reinforcement learning. *Drones* **2024**, *8*, 60. [[CrossRef](#)]
32. Shakya, A.K.; Pillai, G.; Chakrabarty, S. Reinforcement learning algorithms: A brief survey. *Expert Syst. Appl.* **2023**, *231*, 23541. [[CrossRef](#)]
33. Liu, Y.; Li, X.; Wang, J.; Wei, F.; Yang, J. Reinforcement-learning-based multi-UAV cooperative search for moving targets in 3D scenarios. *Drones* **2024**, *8*, 378. [[CrossRef](#)]
34. Zhang, R.; Zong, Q.; Zhang, X.; Dou, L.; Tian, B. Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 7900–7909. [[CrossRef](#)] [[PubMed](#)]
35. Wu, M.Y.; He, X.J.; Qiu, Z.M.; Chen, Z.H. Guidance law of interceptors against a high-speed maneuvering target based on deep Q-Network. *Trans. Inst. Meas. Control* **2022**, *44*, 1373–1387. [[CrossRef](#)]
36. Li, W.; Zhu, Y.; Zhao, D. Missile guidance with assisted deep reinforcement learning for head-on interception of maneuvering target. *Complex Intell. Syst.* **2022**, *8*, 1205–1216. [[CrossRef](#)]
37. Qin hao, Z.; Baiqiang, A.; Qinxue, Z. Reinforcement learning guidance law of Q-learning. *Syst. Eng. Electron.* **2020**, *42*, 414–419.
38. Guo, J.; Hu, G.; Guo, Z.; Zhou, M. Evaluation Model, Intelligent Assignment, and Cooperative Interception in multimissile and multitarget engagement. *IEEE Trans. Aerosp. Electron. Syst.* **2022**, *58*, 3104–3115. [[CrossRef](#)]
39. Ni, W.; Liu, J.; Li, Z.; Liu, P.; Liang, H. Cooperative guidance strategy for active spacecraft protection from a homing interceptor via deep reinforcement learning. *Mathematics* **2023**, *11*, 4211. [[CrossRef](#)]
40. Wen hong, Z.; Jie, L.; Zhi hong, L.; Lin cheng, S. Improving multi-target cooperative tracking guidance for UAV swarms using multi-agent reinforcement learning. *Chin. J. Aeronaut.* **2022**, *35*, 100–112. [[CrossRef](#)]
41. Vazquez Trejo, J.A.; Ponsart, J.-C.; Adam-Medina, M.; Valencia-Palomo, G. Fault-tolerant observer-based leader-following consensus control for LPV multi-agent systems using virtual actuators. *Int. J. Syst. Sci.* **2025**, *56*, 1816–1833. [[CrossRef](#)]
42. Ha, I.J.; Hur, J.S.; Ko, M.S.; Song, T.L. Performance analysis of PNG laws for randomly maneuvering targets. *IEEE Trans. Aerosp. Electron. Syst.* **1990**, *26*, 713–721. [[CrossRef](#)] [[PubMed](#)]
43. Yu, C.; Velu, A.; Vinitsky, E.; Gao, J.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of ppo in cooperative multi-agent games. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24611–24624. [[CrossRef](#)]

44. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
45. Ma, M.; Song, S. Multi-missile cooperative guidance law for intercepting maneuvering target. *Aero Weapon*. **2021**, *28*, 19–27.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.