aerospace

MDPI

Article

Multi-Agent Reinforcement Learning-Based Intelligent Game Guidance with Complex Constraint

Fucong Liu, Yang Guo *, Shaobo Wang, Jin Wang and Zhengquan Liu

College of Missile Engineering, Rocket Force University of Engineering, Xi'an 710025, China; liufucong828@126.com (F.L.); wangshaobo0325@163.com (S.W.); jinwang202410@163.com (J.W.); liuzhengquan0910@163.com (Z.L.)

* Correspondence: guoyang820@foxmail.com

Abstract

For the complex problems of multi-aircraft cooperative game guidance with No-Fly Zone (NFZ) avoidance and cross-task constraint propagation, a deep deterministic policy gradient algorithm with temporal awareness and priority cooperative optimization (TP-MADDPG) is proposed. Based on the three-body cooperative guidance, a new coupled guidance task is formed by adding the NFZ avoidance constraint. At the same time, considering the constraint compatibility problem in dynamic task switching, the cooperative aircraft are modeled as independent agents with differentiated policy networks. First, a nonlinear kinematic model of the three-body game constructed by Evader–Pursuer–Defender is established. And four complex constraint conditions, namely homing guidance, NFZ avoidance, collision avoidance, and cooperative guidance, are modeled separately. Secondly, the Long Short-Term Memory-based (LSTM) Actor–Critic framework is proposed to dynamically capture the evolution patterns of adversarial scenarios by mining hidden correlations in historical state-action sequences. This enables smooth policy transitions between the cooperative guidance phase and subsequent homing guidance phase, effectively addressing the challenges of environmental non-stationarity and temporal task dependencies. Then, a priority-driven adaptive sampling mechanism is proposed along with a heterogeneous roles cooperative reward function to specifically address credit assignment imbalance and sparse reward problems, respectively. The sampling mechanism capitalizes on the efficient retrieval properties of SumTree data structures while integrating bias correction techniques to expedite policy gradient convergence. The reward function utilizes the reward shaping method to formulate cooperative reward components that explicitly capture behavioral correlations among agents. Finally, simulations show that the proposed method significantly outperforms multi-agent reinforcement learning baselines, effectively improving the performance of cooperative game guidance under complex constraints.

Keywords: multi-agent reinforcement learning; no-fly zone avoidance; complex constraint condition; three-body confrontation; cooperative game guidance; LSTM network

Check for updates

Academic Editors: Seokwon Lee and Mingu Kim

Received: 26 January 2026
Revised: 27 March 2026
Accepted: 9 April 2026
Published: 14 April 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) license.

1. Introduction

The modern warfare landscape is rapidly evolving, with battlefields and operational modes exhibiting diverse characteristics. Coupled with highly interconnected and interdependent tasks, aircraft face numerous complex constraints in highly dynamic and intense environments [1–3]. Furthermore, with the continuous advancement of missile

defense systems, the survivability of individual aircraft is further diminished, making it difficult to achieve operational effectiveness. This necessitates leveraging swarm cooperation capabilities to gain asymmetric advantages over adversaries.

As a future development trend for aircraft, cooperative guidance focuses on organically integrating the limited individual capabilities of different aircraft. This integration overcomes the shortcomings of a single aircraft, enabling more powerful and complex guidance behaviors [4]. The initial research focus of cooperative guidance was on cooperative strike, involving the design of guidance laws to meet terminal engagement requirements. This can be categorized into three types: time cooperation, spatial cooperation, and spatio-temporal cooperation. For example, Jeon et al. [5] specified a common expected arrival time in advance, and all aircraft reached the target in accordance with this time constraint. Ratnoo et al. [6] first investigated the impact of terminal angle constraints on the gain coefficient in proportional navigation (PN) and proposed a two-stage proportional navigation guidance law to achieve terminal angle control. Harl et al. [7] achieved cooperative control of terminal time and angle using a sliding mode control method. Beyond terminal cooperative engagement, cooperative guidance also addresses trajectory cooperation during the guidance process, with formation control of multi-vehicle systems being the most common application. Shi et al. [8] proposed a near-optimal tracking controller that incorporates an event-triggered mechanism to reduce control command execution frequency, while employing adaptive dynamic programming to resolve the leader-follower formation control problem for aircraft.

Both cooperative engagement and formation control are inherently single-mission tasks, differing only in their operational phases. However, as the detection and interception capabilities of defense systems strengthen and the intensity of battlefield gaming escalates, future research on cooperative guidance will increasingly focus on a series of novel tasks, including cooperative penetration and cooperative game-based confrontation. These mission scenarios typically involve the coupling of multiple discrete tasks, necessitating coordinated decision-making capabilities among aircraft. In the context of cooperative penetration, recent research has focused on synergistic collaboration between aircraft assuming distinct roles, with the three-body confrontation emerging as a canonical case study. This problem was first formulated by Boyell et al. [9], who derived a closed-form solution for constant-bearing trajectory motion. Shima et al. [10] proposed the Target–Attacker–Defender (TAD) model. Under known guidance strategies of Attacker, they developed a cooperative optimal control framework that transforms Target's behavior from traditional passive evasion to active deceptive maneuvers, thereby reversing Defender's inherent disadvantage. Some studies alternatively define this as the Evader–Pursuer–Defender (EPD) model, though both formulations are fundamentally equivalent in essence. Rubinsky et al. [11] adopted the pursuer's perspective, addressing its dual mission requirements of both evading defender interception and accurately engaging the evader. By employing a differential game framework with safety-function-based control strategy switching, they resolved the strategic conflict in dual guidance tasks and derived corresponding algebraic conditions. Wang et al. [12] conceptualized high-value aerial aircraft as the luring role while coordinating with Defender through cooperative guidance. They developed an explicit cooperative guidance law based on optimal control theory, incorporating both the luring mechanism and terminal angle constraints, thereby creating favorable conditions for high-value target penetration. Shi et al. [13] developed a cooperative predictive guidance law that induces predictable trajectories of attacker through targeted maneuvering by target, enabling Defender to execute intercepts based on predicted collision points. This approach requires only

Attacker's position and velocity, remaining agnostic to its specific guidance modes or acceleration profiles.

The prevailing research paradigm for cooperative game guidance focuses on guidance law design through differential game methodologies. Garcia et al. [14] formulated the active target defense scenario as a zero-sum differential game. Building upon a complete closed-form solution, they developed closed-loop state feedback optimal strategies, derived the game's value function, and explicitly characterized the target escape set. Sun et al. [15] adopted the attacker's perspective to derive optimal guidance laws for pairwise engagement scenarios in the three-body confrontation using differential game theory. Furthermore, they investigated the boundaries between evasion and pursuit strategies for the attacker. Unlike previous studies, this work considers scenarios with unknown defender acceleration, deriving an open-loop solution form for the attacker's guidance law. Wei et al. [16] proposed a cooperative defense strategy based on behavioral mode switching of the pursuer. Using differential game theory, they modeled the pursuer's behavior modes as either attacking or evading. By integrating these mode identification results with linear quadratic optimization theory, a bidirectional cooperative optimization strategy was developed for defenders. Sinha et al. [17] introduced a time-constrained mechanism for designing nonlinear feedback control laws. Distinct from differential games, optimal control, and other heuristic approaches, this method enables the evader to avoid the pursuer through precise impact-time control. The cooperative strategy was designed with dual modes: offensive and defensive strategies. Ultimately, a fixed-time sliding mode variable structure control strategy was employed to develop the cooperative guidance law for defenders. While most of these strategies are based on linearized kinematic models to simplify guidance design, they exhibit significant errors and adaptability limitations in scenarios involving highly maneuvering target engagement and multi-constraint cooperative operations.

As a typical process constraint, NFZ avoidance can generally be categorized into offline trajectory design methods [18–20] and online replanning approaches [21–23]. When considering cooperation among multiple aircraft, which inherently introduces additional constraints, numerous research-worthy problems emerge. Huang et al. [24] addressed the cooperative penetration trajectory optimization problem for aircraft formations in complex threat environments by proposing a two-stage rapid optimization strategy combining the rapidly exploring random tree (RRT) method with the Gauss pseudospectral method. They formulated the objective function by weighting constraints including NFZ, mission points, and flight duration to solve for optimal multi-aircraft trajectories. Yu et al. [25] derived a high-precision analytical formula accounting for Earth's rotation effects to determine the entry-phase initiation time for each vehicle through offline trajectory prediction. By backward-calculating launch times based on boost-phase guidance processes, they achieved simultaneous arrival under heterogeneous initial conditions, thereby solving the cooperative time-of-arrival problem for multiple hypersonic glide vehicles subject to multiple NFZ constraints. Li et al. [26] proposed a Transformer-based real-time cooperative guidance strategy to address the coordinated attack problem for multiple hypersonic vehicles under irregular NFZ constraints. Wang et al. [27] investigated the Pursuer as the research subject within a combined linear-quadratic and norm-bounded differential game framework. By introducing switching times to divide the engagement into four distinct phases, they developed phase-specific guidance strategies and derived a composite guidance law incorporating obstacle avoidance through dead-zone functions. This ensures the Pursuer can successfully evade the Defender's interception and strike the Evader while maintaining obstacle avoidance capability. From

the Defenders' perspective, the core focus of cooperative guidance lies in achieving successful escape, which constitutes a terminal constraint. Meanwhile, the incorporation of NFZ avoidance during the guidance process represents a process constraint. Currently, few studies have integrated these two types of mission requirements.

The intensifying adversarial engagements between opposing aircraft factions have created an urgent demand for enhanced intelligent and autonomous capabilities in guidance systems. Conventional guidance approaches, typically reliant on predefined trajectory planning and static decision rules, demonstrate insufficient adaptability and flexibility when confronted with dynamically complex battlefield environments. The advancement of deep learning has catalyzed the emergence of deep reinforcement learning (DRL), which demonstrates remarkable advantages in addressing complex decision-making problems within high-dimensional, dynamic environments. This approach holds particularly promising application prospects in the domain of guidance and control systems. Shalumov et al. [28] formulated the optimal launch timing for the defender in the TAD model and the target's maneuver control before/after launch as a switched-system optimization problem, which they resolved through deep reinforcement learning to address the mixed-integer nonlinear programming challenge. Chen et al. [29] proposed a distributed cooperative guidance law based on the DDPG algorithm to achieve simultaneous attack on stationary targets by multiple aircraft under different initial conditions. The time-to-go estimation abandons the small-angle assumption in traditional linear models and instead derives a more accurate nonlinear estimation formula. English et al. [30] employed deep reinforcement learning to investigate cooperative game outcomes based on regions where attackers cannot successfully intercept targets derived from differential game theory. This approach enabled agents to learn interception strategies that break through conventional success rate limitations, thereby expanding the methodology for target interception guidance. Gong et al. [31] integrated traditional differential game linearization analysis with deep reinforcement learning. Through adversarial reward function design, they derived both attack strategies and proactive defense strategies applicable to target-defender scenarios. Ni et al. [32] addressed the cooperative engagement of hypersonic aircraft and defender against multiple pursuers by decomposing the complex task into progressively difficult subtasks through curriculum learning, thereby accelerating training. This approach enhanced the escape probability for weakly maneuvering and slow-responding agents, though the generated command signals (specifically lateral acceleration) exhibited chattering phenomena.

All the aforementioned studies employed a single-agent policy network to generate action commands for different aircraft. This method essentially remains a centralized decision-making architecture where all aircraft actions are output by the same network. This architecture fails to enable the autonomous decision-making intelligence of individual aircraft. Moreover, it gives rise to issues such as a sharp increase in network complexity when the number of agents increases, and an inability to respond to local dynamic changes in the environment. Therefore, it is imperative to extend the framework to multi-agent reinforcement learning (MARL), which endows each aircraft with independent decision-making capability while ensuring coordinated group behavior. However, current research on cooperative guidance using MARL remains limited, with most studies still focusing on single-agent reinforcement learning approaches.

Current research in cooperative game guidance predominantly focuses on single-dimensional constraint optimization, such as terminal time coordination or formation configuration maintenance, while lacking comprehensive consideration of terminal constraints, process constraints, and dynamic task switching. Moreover, existing approaches often simplify aircraft nonlinear kinematic models to linear formulations, which reduces

computational complexity at the expense of modeling accuracy. Furthermore, single-agent reinforcement learning struggles to decouple conflicting objectives due to policy coupling and action space dimensionality explosion, resulting in challenging reward function design and non-convergent policies. This is because single-agent strategies require optimizing multiple objectives within the same network, but there may be resource or logical conflicts between objectives. For example, NFZ avoidance may require detours, which could potentially undermine the constraints of cooperative guidance. In contrast, multi-agent reinforcement learning fundamentally addresses this through its distributed optimization paradigm, enabling specialized policy learning for distinct constraints while demonstrating inherent advantages in complex, tightly coupled multi-constraint cooperative guidance scenarios.

In response to the aforementioned challenges, the principal contributions of this study are as follows:

(1) To address the critical challenges of long-term decision-making dependence on historical behaviors and non-stationarity in multi-agent training environments, this study innovatively integrates LSTM modules into the Actor–Critic framework, developing a LSTM-based enhanced MADDPG algorithm. This method endows each agent with memory and reasoning capabilities for interactive historical states, significantly improving the intelligent cooperation between Evader and Defender in dynamic adversarial scenarios to effectively evade interception.

(2) To address the challenges of inefficient experience utilization and convergence difficulties caused by uniform sampling, we propose a priority-driven adaptive sampling mechanism. Specifically, we design and implement a prioritized experience replay method based on the SumTree data structure, incorporating bias correction techniques to optimize the sampling process. The proposed method dynamically evaluates the importance of agent experience samples and allocates sampling probabilities accordingly, enabling focused learning of globally high-priority samples. This approach significantly accelerates the training process and enhances convergence stability of the policy network.

(3) To address the complex scenario involving coupled cooperative game guidance and NFZ avoidance tasks with cross-task constraint propagation, we innovatively developed a role-specific cooperative reward function for heterogeneous agents. By integrating reward shaping techniques with cooperatively designed reward terms, this approach explicitly models behavioral interdependencies among agents. The method not only effectively mitigates sparse reward challenges stemming from strongly coupled objectives and temporal delays, but also enhances overall training efficiency and final policy performance by directing agents' attention toward cooperative benefits and global mission objectives.

The remainder of this paper is organized as follows: Section 2 formulates the guidance problem under complex constraints and presents the constraint modeling methodology. Section 3 introduces the theoretical framework of multi-agent reinforcement learning. Section 4 details the proposed TP-MADDPG algorithm and guidance law design. Section 5 comprehensively presents the simulation results. Section 6 concludes the research with a summary of key contributions. The overall framework of this study is illustrated in Figure 1.

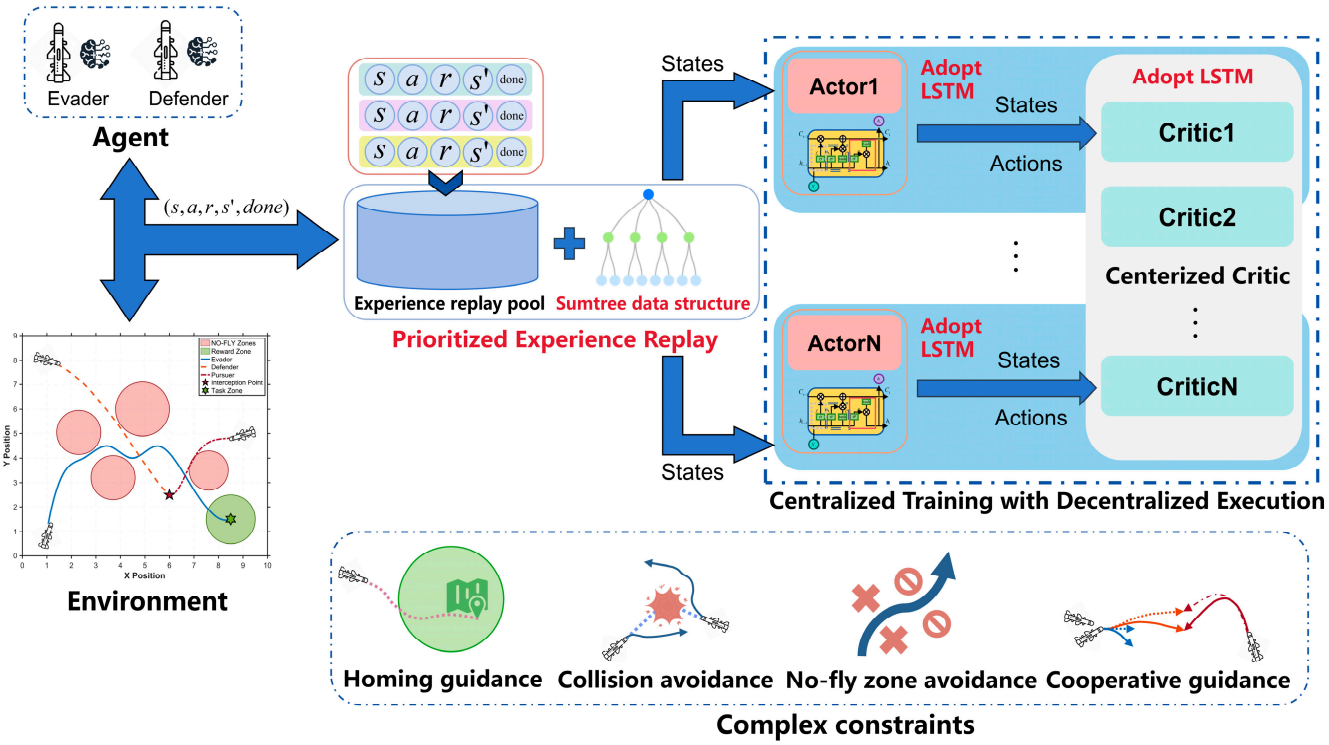


Figure 1. Overall technical roadmap.

2. Problem Modeling

A complex constraint scenario refers to a complex situation where multiple restrictive conditions, elements, or requirements must be taken into account. In this paper, it refers to the complex scenario involving multiple decision-making processes that aircraft face during mission completion, including the threat of interception by enemy aircraft, as well as restrictions imposed by NFZ and target areas.

This section formally defines the planar engagement model, constructed under the following fundamental assumptions:

(1) The Pursuer, Evader, and Defender are all regarded as mass points and move in a two-dimensional plane coordinate system.
(2) During the movement, the magnitudes of the velocities of the three remain constant.
(3) Pursuer intercepts Evader using Augmented Proportional Navigation (APN).

2.1. Dynamics and Kinematics Models

Suppose there are two aircraft on our side, Evader and Defender, and one aircraft Pursuer on the opponent's side. The engagement scenario in a two-dimensional plane is shown in Figure 2. The dynamics and kinematics equations are established in the inertial coordinate system. Among them, a, v, r, q, and γ represent normal acceleration, relative distance, line-of-sight angle, and heading angle respectively. x and y represent the current position coordinates. The subscripts E, D, and P represent the Evader, the Defender, and the Pursuer respectively. PE and PD represent the relationships between the interceptor and the target, and between the interceptor and the defender, respectively.

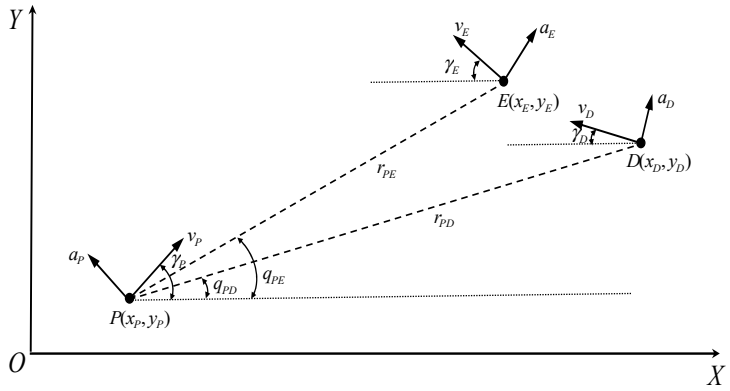


Figure 2. Planar engagement geometry.

The dynamic equations of Evader, Defender and Pursuer are respectively

$$\begin{cases} \ddot{x}_E = a_E \sin\gamma_E \\ \ddot{y}_E = a_E \cos\gamma_E \\ \dot{\gamma}_E = a_E / v_E \end{cases} \tag{1}$$

$$\begin{cases} \ddot{x}_D = a_D \sin\gamma_D \\ \ddot{y}_D = a_D \cos\gamma_D \\ \dot{\gamma}_D = a_D / v_D \end{cases} \tag{2}$$

$$\begin{cases} \ddot{x}_P = -a_P \sin\gamma_P \\ \ddot{y}_P = a_P \cos\gamma_P \\ \dot{\gamma}_P = a_P / v_P \end{cases} \tag{3}$$

where $\ddot{x}_E$, $\ddot{y}_E$ and $\dot{\gamma}_E$ are respectively the acceleration components of Evader in the X-axis and Y-axis directions and the rate of the heading angle. The meanings of variables represented by other subscripts can be deduced by analogy.

The relative kinematic equations of the nonlinear Interceptor and the Target are established as follows:

$$\begin{cases} \dot{r}_{PE} = -v_E \cos(\gamma_E + q_{PE}) - v_P \cos(\gamma_P - q_{PE}) \\ r_{PE}\dot{q}_{PE} = v_E \sin(\gamma_E + q_{PE}) - v_P \sin(\gamma_P - q_{PE}) \end{cases} \tag{4}$$

where $\dot{r}_{PE}$ is the relative velocity between the Interceptor and the Target, $\dot{q}_{PE}$ is the line-of-sight angle rate between the Interceptor and the Target. By differentiating with respect to time again, we can obtain

$$\ddot{q}_{PE} = -2\frac{\dot{r}_{PE}}{r_{PE}}\dot{q}_{PE} + \frac{a'_E - a'_P}{r_{PE}} \tag{5}$$

where $\ddot{q}_{PE}$ is the second-order derivative of the line-of-sight angle. a'_E and a'_P are respectively the components of a_E and a_P in the line-of-sight direction.

The calculation formula for the zero-effort miss (ZEM) corresponding to the nonlinear model is

$$ZEM = \frac{r_{PE}^2 \dot{q}_{PE}}{\sqrt{\dot{r}_{PE}^2 + r_{PE}^2 \dot{q}_{PE}^2}} \tag{6}$$

Pursuer adopts APN guidance, and the expression of its command acceleration is

$$a_P = NV_{PE}\dot{q}_{PE} + \frac{K}{2}a_E \quad |a_P| \le a_{P\max} \tag{7}$$

where N, V_{PE}, K and $a_{P\max}$ are respectively the navigation parameter, relative velocity, correction coefficient, and overload limit during the Interceptor's guidance process.

2.2. Modeling of No-Fly Zones and Target Areas

A no-fly zone refers to an area where aircraft are prohibited from entering or overflying due to terrain, geopolitical, or military factors, which imposes constraints on flight paths. The quantity and spatial complexity of such zones significantly influence trajectory planning difficulty.

The target zone refers to the final destination area that the aircraft must reach during mission execution. This zone meets all operational requirements and provides a foundation for successfully completing subsequent tasks such as reconnaissance, patrol, or attack missions.

In this study, both no-fly zones and target zones are modeled as circular areas, with the mathematical formulation established as follows:

$$(x_i - x_j)^2 + (y_i - y_j)^2 \leq R_j^2 \tag{8}$$

where $(x_i, y_i), i \in \{E, D\}$ represents the coordinates of the aircraft's current position, $(x_j, y_j), j \in \{N, T\}$ denotes the center coordinates of either the NFZ or target zone, and $R_j, j \in \{N, T\}$ indicates the radius of the respective zone.

The circular zones vary in size and exhibit maximally complex spatial distribution patterns. These zones maintain fixed positions throughout the operation, representing static NFZ and target areas. A partial schematic of these zones is illustrated in Figure 3.

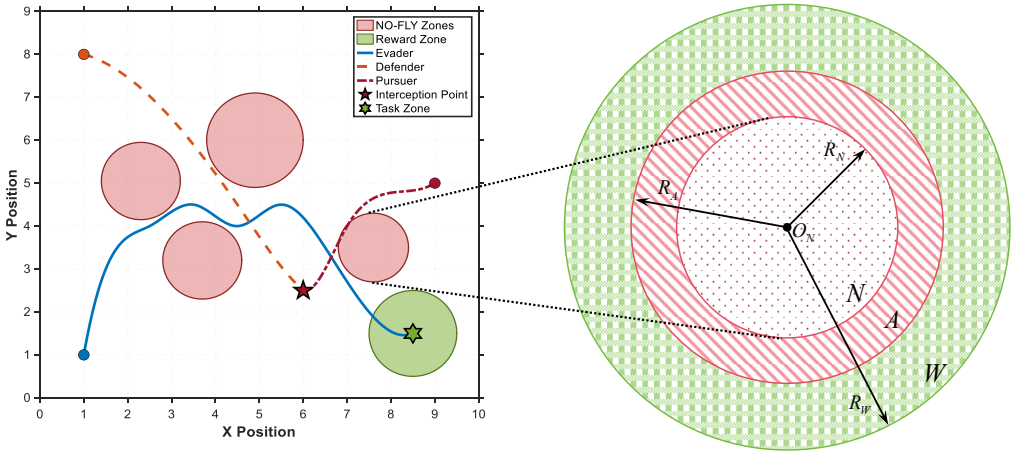


Figure 3. Schematic diagram of NFZ and target zones.

For the target zone, the original model remains unchanged since mission completion is achieved upon the aircraft's entry. In contrast, the NFZ model incorporates two additional annular regions surrounding the original zone, creating a three-tiered structure. This modification is necessary because the aircraft must first detect the NFZ before executing avoidance maneuvers. The zones are arranged concentrically from outermost to innermost as: the warning zone, the avoidance zone, and the NFZ.

Using the maximum radius as the collision avoidance criterion, the Euclidean distance between the NFZ and the agent is expressed as

$$d \triangleq d(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{p}^o\| \tag{9}$$

where $\boldsymbol{p}$ represents the position vector of the aircraft, and $\boldsymbol{p}^o$ represents the position vector of the center of the NFZ.

For the warning area, when an aircraft enters this area, it should detect the NFZ and take corresponding evasive actions. Define its radius as R_W, and the expression of this area is

$$W = \left\{ R_A \leq d \triangleq \|\boldsymbol{p} - \boldsymbol{p}^o\| < R_W \right\} \tag{10}$$

The avoidance area can be defined as a buffer region where aircraft must perform mandatory collision avoidance maneuvers if they fail to complete evasive actions within the warning area, thereby guaranteeing system safety compliance. Defined with radius R_A, this zone is mathematically expressed as

$$A = \left\{ R_N \leq d \triangleq \|\boldsymbol{p} - \boldsymbol{p}^o\| < R_A \right\} \tag{11}$$

For the NFZ, define its radius as R_N, and the expression of the zone is

$$N = \left\{ d \triangleq \|\boldsymbol{p} - \boldsymbol{p}^o\| < R_N \right\} \tag{12}$$

Therefore, the guidance problem in complex constrained scenarios can be regarded as a sequential decision-making process. The entire mission process can be described as follows: during the process of Evader reaching the final target area, it needs to avoid the NFZ and cooperate with Defender to evade Pursuer's attack. At the same time, while evading the NFZ, Defender needs to cooperate with Evader's timely maneuvers to intercept Pursuer, increase Evader's survival probability, and lay the foundation for the completion of subsequent tasks.

2.3. Modeling of Complex Constraints

The complex constraints in this study can be categorized into four aspects, which are elaborated below.

2.3.1. Homing Guidance Constraints

For Evader, its constraint is to reach the mission area at the final moment after maneuvering. Therefore, its constraint is

$$(x_E(t_f) - x_T)^2 + (y_E(t_f) - y_T)^2 \leq R_T^2 \tag{13}$$

where $x_E(t_f)$ and $y_E(t_f)$ are the X-coordinate and Y-coordinate of Evader at the final moment respectively.

For both Pursuer and Defender, they need to complete the guidance mission. Therefore, they need to satisfy the relative-distance and ZEM constraints as follows:

$$\begin{cases} \lim\limits_{t \to t_f'} \|ZEM(t)\| = 0 \\ \lim\limits_{t \to t_f'} \|r(t)\| = 0 \end{cases} \tag{14}$$

where t_f' is the termination time of the guidance mission.

2.3.2. No-Fly Zone Avoidance Constraints

During the flight, Evader and Defender need to detect and avoid multiple NFZ, and satisfy the following constraints:

$$\sqrt{(x_i - x_N)^2 + (y_i - y_N)^2} > R_N \tag{15}$$

For the Pursuer, it does not need to consider the NFZ constraints and can choose the optimal path to attack directly by passing through the NFZ. This increases the penetration difficulty of our aircraft.

2.3.3. Collision Avoidance Constraints

During the flight, the distance between Evader and Defender needs to be kept within the safe threshold all the time, and the following collision avoidance constraint should be satisfied:

$$\sqrt{(x_E - x_D)^2 + (y_E - y_D)^2} \geq d_{safe} \tag{16}$$

where d_{safe} represents the minimum safe-distance threshold.

2.3.4. Cooperative Guidance Constraints

For Evader and Defender, they need to cooperate while satisfying the first three constraint conditions to achieve the goal of Defender successfully intercepting Pursuer. Through the timely maneuvers of Evader, Pursuer can be made to move towards a position where Defender can intercept more easily, reducing the overall overload of Defender and increasing the success rate of guidance. This involves information interaction between Evader and Defender and the cooperative design of acceleration.

Define the desired relative position vector $\boldsymbol{r}_{PD}^{des}$. At this time, it is required that the maneuvers of Evader can force the trajectory of Pursuer to satisfy the following constraints:

$$\min_{a_E} \left\| \boldsymbol{r}_{PD}(t) - \boldsymbol{r}_{PD}^{des} \right\|^2 \tag{17}$$

where $\boldsymbol{r}_{PD} = \boldsymbol{r}_P - \boldsymbol{r}_D$ is the position vector of Pursuer relative to Defender.

At the same time, it is required that Defender completes the interception within the time window created by Evader, and the following constraints need to be satisfied:

$$t_f^D \leq t_f^E + \Delta t_{tol} \tag{18}$$

where t_f^D is the estimated time for Defender to intercept Pursuer, t_f^E is the time for Evader to force Pursuer to reach the preset position, and Δt_{tol} is the allowable time error.

3. Multi-Agent Reinforcement Learning Framework

First, introduce the basic concepts of deep reinforcement learning and select the Actor–Critic structure as the training framework. Secondly, by comparing with the single-agent model algorithm, expound the reasons for choosing MARL. Finally, introduce the basic principles of the MADDPG algorithm, laying the foundation for subsequent algorithm improvements.

3.1. Deep Reinforcement Learning

Reinforcement learning is an important branch of machine learning, independent of the two modes of supervised learning and unsupervised learning. It is usually used to solve sequential decision-making problems. The key to such problems is that an agent solves a policy in the interaction with the environment to generate corresponding actions in different states, with the aim of maximizing the cumulative long-term reward. However, due to the complex temporal relationships involved, especially for the long-distance and long-time-series decision-making problems in this paper, the memory function of the agent and the strong correlation between data must be considered. At this time, the concept of Markov Decision Process (MDP) needs to be introduced.

As the basic framework of reinforcement learning, the MDP consists of (S, A, P, R, γ) elements, including the environmental state S, the agent's action A, the state-transition probability P, the reward function R, and the discount factor γ for future rewards in sequence. Its core feature is that the state and reward of the environment at the next moment only depend on the state at the current moment and the action of the agent. The task flow of reinforcement learning is shown in Figure 4.

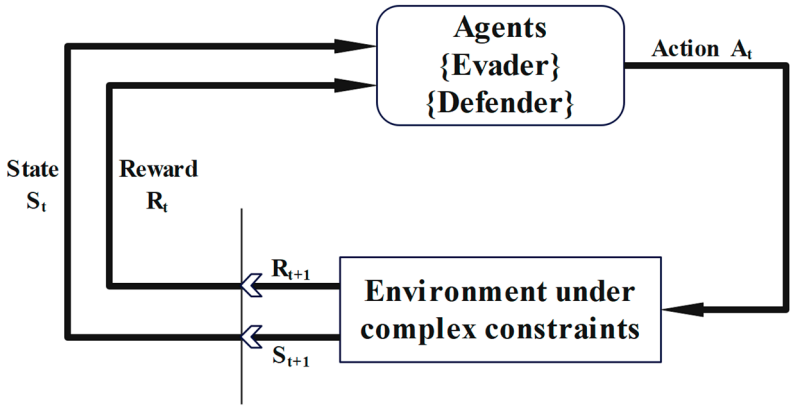


Figure 4. Schematic diagram of reinforcement learning process.

3.2. Multi-Agent Reinforcement Learning

Single-agent models typically assume that the environment remains stable, independent of the agent's own learning process. However, in many practical scenarios, agents do not exist in isolation. Their behaviors not only affect the environment, but more importantly, their decision-making spaces and reward forms are significantly influenced by the strategies and behaviors of other independent agents in the environment. This means that single-agent models cannot depict the dynamic interaction process in which the behaviors of multiple decision-making entities affect each other, and are even less capable of describing scenario tasks with cooperative, competitive, or mixed relationships.

In complex systems or task collaborations, multiple agents are required to work together. Each agent has an independent policy network and generates actions according to different states. The important difference between them lies in the non-stationarity of the environment, that is, the action selection of one agent will directly affect the rewards and states of other agents, and the environment is constantly changing dynamically. Multi-agent reinforcement learning can model and solve various complex interaction patterns, and at the same time support decentralized decision-making, thereby improving the overall robustness of the system.

In complex constraint scenarios, there is a cooperation problem between Evader and Defender. Moreover, the choice of one party can influence the state and actions of the other. This interaction requires agents to take into account the behaviors of other agents when making decisions. In this paper, Evader and Defender need to cooperate to complete the task, and how to achieve the common goal through collaboration has become a crucial issue. Therefore, by choosing the multi-agent reinforcement learning approach, the synergy between them can be fully explored, making the completion of the entire task more rational and efficient.

3.3. Multi-Agent Deep Deterministic Policy Gradient Algorithm

The Deep Deterministic Policy Gradient (DDPG) algorithm builds upon Deep Q-learning (DQN) by adopting an Actor–Critic architecture. This framework simultaneously learns approximate optimal Q-functions and optimal actions, effectively addressing chal-

lenges in both continuous state space representation and continuous action space evaluation. The algorithm maintains training stability and sample diversity through continued use of experience replay and target network techniques.

For the Critic component, its primary function is to evaluate state-action pair values. Specifically, it employs neural network $Q_\varphi(s,a)$ to approximate the Bellman equation $Q^*(s,a)$ of the optimal state-action value function. This enables accurate Q-value outputs for given state-action pairs (s,a), expressed as

$$Q^*(s,a) = \underset{s'\sim P}{E}[r(s,a) + \gamma \max_{a'} Q^*(s',a')] \tag{19}$$

where $s' \sim P$ refers to the state at the next moment, which is sampled from the environmental model $P(\cdot|s,a)$.

The deviation between them is typically measured by mean squared error and adopted as the loss function. The training objective is to find the optimal network parameters ϕ that minimize the final loss, with the specific expression given by

$$\min_{\varphi} Loss = \min_{\varphi} \underset{(s,a,r,s',f)\sim B}{E}[(Q_\varphi(s,a) - (r + \gamma(1-f)Q_\varphi(s',a')))^2] \tag{20}$$

where B is the experience replay pool, in which the stored data is (s,a,r,s',f), representing the current state, the action taken, the environmental reward, the next state, and the training stop flag respectively. $\gamma \in [0,1]$ is the discount factor.

The optimal solution can be obtained by applying gradient descent to the loss function. However, directly updating the network parameters in Equation (20) after each gradient step may lead to training instability. To address this, DDPG employs dual networks: an online network that updates parameters in real-time following gradient directions, and a target network that synchronizes parameters with the online network at fixed intervals. Specifically, DDPG adopts a soft update approach that combines parameters from both networks, expressed as

$$\widetilde{\varphi} = \tau\varphi + (1-\tau)\widetilde{\varphi} \tag{21}$$

where $\widetilde{\varphi}$ represents the target network parameters, ϕ represents the online network parameters, and τ represents the update factor.

The Actor component functions to output deterministic actions based on environmental states, ensuring the input to Critic yields maximum Q-values. Its training methodology involves:

$$\max_{\theta} \underset{s\sim B}{E}[Q_\varphi(s,a)] \overset{a=\mu_\theta(s)}{=\!=} \max_{\theta} \underset{s\sim B}{E}[Q_\varphi(s,\mu_\theta(s))] \tag{22}$$

where $\mu_\theta(s)$ is the strategy for generating actions and θ is the network parameter.

However, DDPG exhibits limitations in multi-agent environments primarily because each agent must learn stable policies while other agents' strategies continuously evolve. To address this, MADDPG extends DDPG by adopting the Centralized Training with Decentralized Execution (CTDE) paradigm, effectively expanding multi-agent scenarios to accommodate mixed cooperative–competitive environments.

Centralized training refers to leveraging global information during the learning process, where the Critic network incorporates state and action data from all agents. Decentralized execution means each agent maintains an independent Actor network that makes decisions solely based on its local observations and policy network. The schematic diagram of the training method of the MADDPG algorithm is shown in Figure 5.

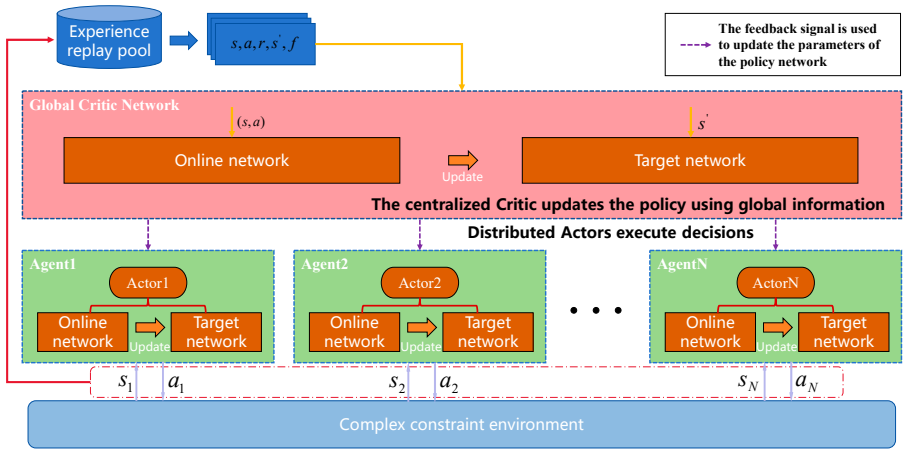


Figure 5. Schematic diagram of the training method of the MADDPG algorithm.

4. Multi-Agent Reinforcement Learning-Based Intelligent Game Guidance Method

In this section, the design method of the intelligent game guidance law for complex constraint scenarios is introduced in detail. The proposed framework addresses temporal dependencies in mission-critical decision-making through sequential memory mechanisms while solving the credit assignment problem in multi-agent reinforcement learning via prioritized experience replay algorithms. The cooperative architecture features a constrained state-action space formulation integrated with neural networks containing temporal feature extraction modules, along with specially designed cooperative game reward functions for adversarial interactions.

4.1. TP-MADDPG Algorithm Model

4.1.1. LSTM-Based Enhanced MADDPG Algorithm

Long Short-Term Memory, a variant of recurrent neural network, is used to process and predict time-series data. It addresses the problems of gradient vanishing and gradient explosion in RNN. Its main structures include the forget gate, input gate, and output gate, etc. The specific network structure is shown in Figure 6.

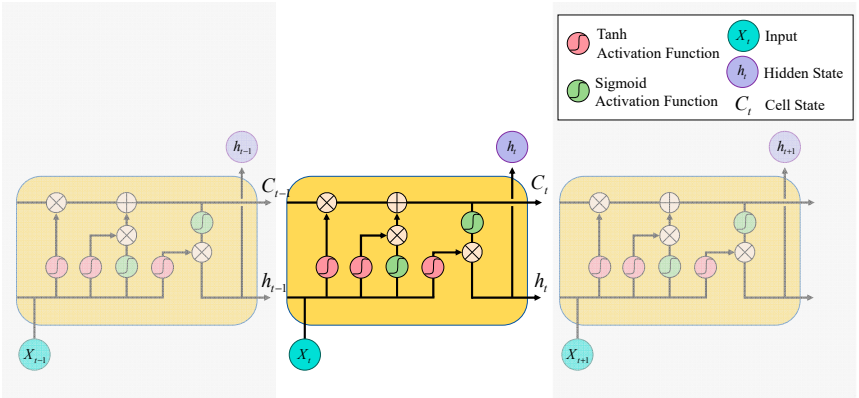


Figure 6. Schematic diagram of LSTM network structure.

The schematic diagram of the gating structure is shown in Figure 7. X_t is the input of the time series, which is the state space of the agent in this task; h_t is the output of the time series, and t refers to the current time. The data flow is as follows. The output h_{t-1} from the previous time step is concatenated with the input X_t at the current time step, and then they pass through the forget gate, input gate, and output gate in sequence. This results in the

forget weight and output weight for storing the information C_{t-1} from the previous time step. Based on different weights, some of the information in C_{t-1} is selectively discarded, and the critical state $\widetilde{C}_t$ information is selectively stored, thereby updating the storage unit C_t. The following are the update formulas according to different types of gating units:

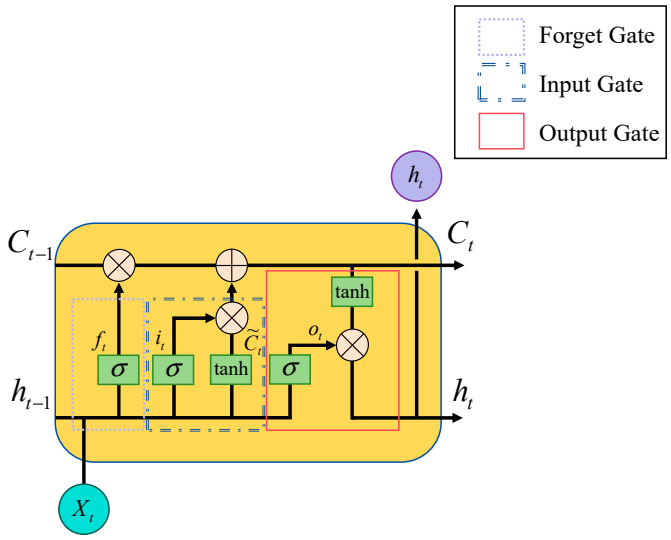


Figure 7. Schematic diagram of gating mechanisms in LSTM.

Forget gate:

$$f_t = \sigma(W_f \cdot [h_{t-1}, X_t] + b_f) \tag{23}$$

where f_t is the process vector, $\sigma(\cdot)$ represents the sigmoid activation function, $[,]$ represents the concatenation operation, W_f is the weight matrix, and b_f is the bias.

Input gate:

$$\widetilde{C}_t = \tanh(W_C \cdot [h_{t-1}, X_t] + b_C) \tag{24}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, X_t] + b_i) \tag{25}$$

where i_t is the process vector, W_C and W_i are weight matrices, and b_C and b_i are biases.

Output gate:

$$o_t = \sigma(W_o \cdot [h_{t-1}, X_t] + b_o) \tag{26}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{27}$$

where o_t is the process vector, W_o is the weight matrix, and b_o is the bias.

The LSTM network has significant advantages in processing sequential data and capturing temporal dependencies. The memory cells within it can dynamically update and store information, leveraging crucial information from past time steps to better infer the current hidden state. As a result, it has the ability to model the features of long-sequence inputs.

For the complex constraint scenarios in this paper, the decisions of each agent are often influenced by historical behaviors and environmental states. If the dynamically changing constraints are regarded as a hidden state, agents can utilize the accumulated historical information to mine relevant information and improve decision-making accuracy. Coupled with the non-stationary multi-agent environment, by interacting with historical information, agents can better adapt to the training environment even in a non-stationary environment, promoting cooperation and cooperation among agents.

The LSTM network can be combined with either the Actor network or the Critic network, as both are capable of handling time-series data. For the former, the key to introducing the LSTM network lies in dealing with cross-temporal reasoning tasks, enabling the

policy to be more continuous and dynamically adjusting the action distribution according to environmental changes. For the latter, the purpose of introducing the LSTM network is to explore the dynamic change characteristics of the state-action, so as to better estimate the value of the policy and avoid short-sighted Q-value estimation. This task not only relies on the hidden information in previous actions but also requires the evaluation of the long-term value of state-action pairs. However, the traditional Actor–Critic architecture only depends on the current state and cannot establish the causal relationship between historical states and future rewards. Therefore, the LSTM network is introduced into both the Actor and the Critic. The algorithm flow chart after integrating the LSTM network is shown in Figure 8.

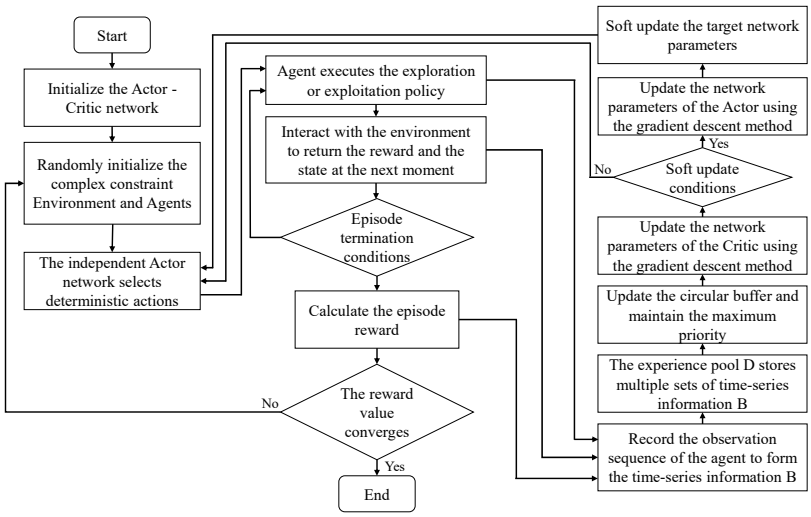


Figure 8. LSTM-based enhanced MADDPG algorithm flowchart.

4.1.2. Priority-Driven Adaptive Sampling Mechanism

The traditional experience replay mechanism based on uniform sampling has drawbacks, leading to issues such as low training efficiency and imbalance in the number of positive and negative samples. This paper focuses on complex constraint scenarios. During the training process, a large number of low-value samples can reduce the quality of experience replay. The priority mechanism enables model training to focus on high-value samples, thus accelerating training and enabling the model to converge more quickly.

Based on this requirement, a priority-driven adaptive sampling mechanism is proposed. The calculation method for the priority of each sample is as follows:

$$P(i) = \frac{(|\delta_i| + \varepsilon)^{\alpha}}{\sum_j (|\delta_j| + \varepsilon)^{\alpha}} \tag{28}$$

where $P(i)$ is the priority of the sample; δ is the temporal difference error; ε is the minimum priority, usually a very small constant to ensure that all samples have a chance to be sampled; and α is the priority exponent, which is used to control the priority distribution.

At the same time, the bias-corrected optimization learning process is introduced as follows:

$$w_i = \left[\frac{1}{N \cdot P(i)}\right]^{\beta} \tag{29}$$

where w_i is the importance weight, which is ultimately used as the weight for gradient update. N is the buffer size, that is, the capacity of the experience pool. β is the importance sampling exponent, and it also serves as a bias correction coefficient to adjust the entire optimization learning process.

Since there is a bias in the priority of samples in the initial stage, in order not to overly rely on the importance sampling correction in the early stage of training, a method of dynamically updating β is adopted. As the training progresses, the correction intensity is gradually increased, ultimately achieving full correction to balance the convergence speed and stability.

$$\beta_{t+1} = \min(1.0, \beta_t + \Delta\beta) \tag{30}$$

where β_t is the value of the bias correction coefficient at the current time, and $\Delta\beta$ is the increment per step.

Store the priorities of different samples according to the SumTree data structure, and maintain an independent "priority tree" for each agent. The SumTree data structure is shown in Figure 9. Based on the global average TD-error, priority synchronization among different agents is achieved to ensure the accuracy of experience replay. Compared with the traditional uniform sampling, the improved experience replay mechanism demonstrates significant advantages in key indicators such as convergence speed, sparse reward learning, and long-term return optimization. It solves the learning efficiency challenges caused by the traditional method's neglect of differences in experience value.

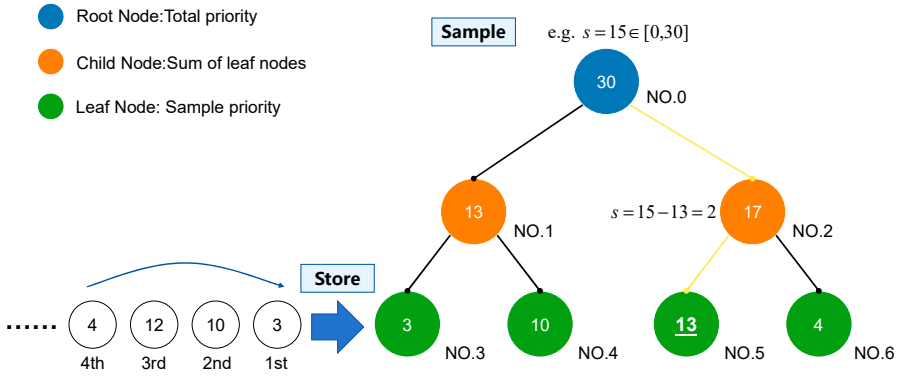


Figure 9. Schematic diagram of SumTree data structure.

4.2. Multi-Agent Cooperative Design Based on TP-MADDPG

Evader and Defender play different roles and perform different tasks in the terminal guidance game. However, their common task is to cooperate with each other so that Evader can evade Pursuer, avoid NFZ, and reach the target area where it is convenient to carry out tasks. As two independent agents, they each have their own policy networks.

$$agentE : \mu_{\theta E}(\cdot)\ agentD : \mu_{\theta D}(\cdot) \tag{31}$$

4.2.1. State Space and Action Space

In this task, it is assumed that communication is always maintained between them, that is, they can share each other's state information. The design of the global state space is as follows:

$$\begin{aligned} S &= [s_E, s_D, s_P] \\ &\begin{cases} s_E = [p_E, v_E, \gamma_E] \\ s_D = [p_D, v_D, \gamma_D] \\ s_P = [p_P, v_P, \gamma_P] \end{cases} \end{aligned} \tag{32}$$

where s_E, s_D, s_P represents the state space information of Evader, Defender, and Pursuer respectively. The state of each aircraft is composed of its own position vector, velocity, and heading angle.

The action space can generally be divided into discrete actions, continuous actions, and hybrid actions. In the terminal guidance process, continuous control is required to achieve

complex cooperation. Compared with discrete actions, the continuous action space can adjust actions more smoothly in response to environmental changes. Therefore, this paper selects continuous actions. The action space dimension of both agents is one-dimensional, and the normal acceleration is used as the action command.

$$\begin{aligned} A &= \{a_E, a_D\} \\ a_i &\in [a_{i\min}, a_{i\max}], i = E, D \end{aligned} \tag{33}$$

where a_E, a_D are the normal accelerations of the Evader and the Defender respectively. $a_{i\min}, a_{i\max}$ are the minimum and maximum values for generating actions respectively.

4.2.2. Neural Network Structure Integrating Temporal Feature Extraction

By introducing the LSTM network into the original neural network structure, it is possible to more fully explore the time-related content in both actions and states. At the same time, it is necessary to specify the length of the time series to be focused on, and splice the action and state information of the agent into a network aligned to the specified length.

In the Actor–Critic architecture, there are a target network and an online network. Both have the same network structure, as shown in Figure 10.

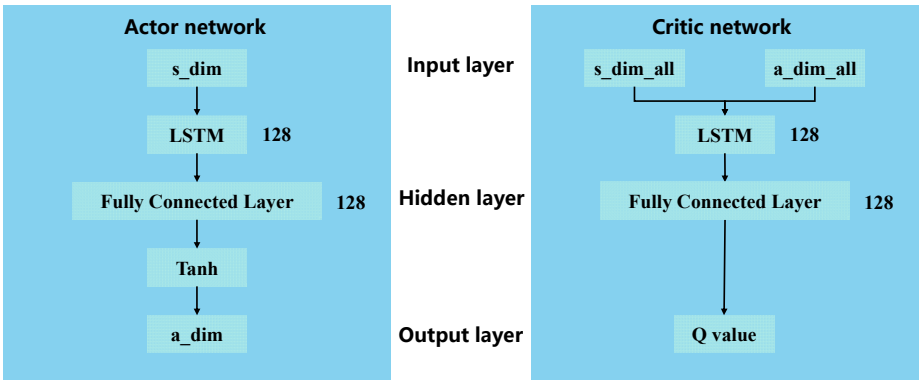


Figure 10. Schematic diagram of the neural network structure in the Actor–Critic architecture.

For the Actor network, each agent takes its 12-dimensional observation information s_dim, trains it through an LSTM network and a fully connected network, both with 128 neurons. Then it outputs 1-dimensional original action information, which is amplitude-limited by the Tanh activation function to obtain the final action information. It should be emphasized that Evader and Defender have their own independent Actor networks.

For the Critic network, due to its CTDE structure, dimension unification is required first. This involves concatenating the observation information s_dim_all and action information a_dim_all of all agents. After selecting the specified time-series length, the observation and action information are jointly concatenated. Then, they are trained successively through an LSTM network and a fully connected network, both with 128 neurons, and finally, a 1-dimensional Q-value is output.

4.2.3. Reward Function for Cooperative Game with Heterogeneous Roles

Agent E and Agent D have a fully cooperative relationship. They each have an independent reward function, but the reward value in the final round is the sum of the rewards of each agent. If only the terminal reward is used as the reward function, that is, a reward is given only when the task is successful, it will lead to low exploration efficiency of the agents.

Reward Shaping mainly addresses the issues of sparse rewards and credit assignment during the training process. By introducing some prior knowledge, a dense additional

reward function is designed to guide agents to learn efficiently. Therefore, for the game guidance task in the complex constraint scenario of this paper, two types of rewards, namely terminal reward and process reward, are designed as follows:

$$R = R_{ter} + R_{pro} \tag{34}$$

where R_{ter}, R_{pro} represent the terminal reward and the process reward respectively.

The terminal reward includes a time penalty r_{time}, a reward r_{goal} for Evader reaching the target area, a reward r_{win} for Defender intercepting Pursuer, and a penalty r_{hit} for Pursuer intercepting Evader. Its form is as follows:

$$R_{ter} = r_{time} + r_{goal} + r_{win} + r_{hit} \tag{35}$$

When the number of training steps in each episode exceeds the predefined time, a penalty will be uniformly imposed to encourage the agents to explore the optimal strategy as soon as possible. When Evader or Defender accomplishes their respective tasks, a positive reward will be given. If Evader is intercepted by Pursuer, the training will stop immediately and a large penalty value will be given to all agents.

The process reward is designed separately for different agents. The design of the general NFZ penalty term r_0 is as follows:

$$r_0(x) = f(x) \bullet \frac{K_f}{d - R_N}$$
$$f(x) = \begin{cases} 0, & d > R_W \\ c_1 + c_1 \cos\left(\pi \frac{d^2 - R_A^2}{R_W^2 - R_A^2}\right), & R_A < d \le R_W \\ c_2 + c_3 \cos\left(\pi \frac{d^2 - R_N^2}{R_A^2 - R_N^2}\right), & R_N < d \le R_A \\ 1, & d \le R_N \end{cases} \tag{36}$$
$$\begin{cases} c_2 + c_3 = 1, \\ c_2 - c_3 = 2c_1 \end{cases}$$

Construct $f(x)$ in the form of a piece-wise function of the distance between the aircraft and the NFZ. Appropriate penalties are given for different regions where the aircraft is located. The closer to the NFZ, the stronger the penalty. By introducing the cosine function, the penalty intensity changes smoothly at the safety boundary. K_f, c_1, c_2, c_3 in it serves as a reward adjustment coefficient.

For Agent E, its process reward consists of the NFZ penalty r_0, angle reward r_{ang}^E, distance reward r_{dis}^E, and cooperation reward r_{col}^E, in the following form:

$$r_{pro}^E = r_0 + r_{ang}^E + r_{dis}^E + r_{col}^E \tag{37}$$

The angle reward is obtained by calculating the difference between the heading angle γ_E of Evader at each moment and the standard heading angle γ_E^{des}, so as to adjust the action strategy and avoid the situation where the velocity direction of Evader deviates too much from the direction pointing to the target area. The standard heading angle is calculated from the line connecting the position of Evader and the center of the target area. Its form is as follows:

$$r_{ang}^E = \alpha \bullet \cos(\gamma_E^{des} - \gamma_E) \tag{38}$$

The distance reward is aimed at the distance d' from Evader to the target area, with the expectation of reaching the target area along the shortest path. Its form is as follows:

$$r_{dis}^{E} = \lambda \bullet \frac{1}{d'} \tag{39}$$

The core of the design of the cooperation reward is to establish a connection between the strategies of the two agents, so as to better reflect their cooperative cooperation. This connection is achieved through the position estimation of Pursuer. Its form is as follows:

$$r_{col}^{E} = \eta \bullet \|\boldsymbol{p}_P - \hat{\boldsymbol{p}}_P\|^2 + \omega \tag{40}$$

where $\boldsymbol{p}_P$ is the true future position of Pursuer, $\hat{\boldsymbol{p}}_P$ is Defender's estimation of Pursuer's future position, and η, ω are all adjustment coefficients.

The specific way for Defender to estimate the future position of Pursuer can be derived from Equation (4) as the following formula:

$$\begin{gathered} \gamma_P = \arcsin\left(\frac{v_E \sin(\gamma_E + q_{PE}) - r_{PE}\dot{q}_{PE}}{v_P}\right) + q_{PE} \\ \boldsymbol{v}(t) = v_P \begin{bmatrix} \cos\gamma_P(t) \\ \sin\gamma_P(t) \end{bmatrix} \\ \boldsymbol{r}(t+\Delta t) = \boldsymbol{r}(t) + \int_t^{t+\Delta t} \boldsymbol{v}(\tau) d\tau \end{gathered} \tag{41}$$

The position of Evader at the next moment can be determined by its action at the current moment. Meanwhile, based on the position and heading angle information of Pursuer at the current moment, its position at the next moment can also be obtained. The focus of the game between the two sides is to obtain the opponent's maneuvering strategy. Based on this, Evader can force Pursuer to move to a position where Defender can easily intercept. Also, since Evader cannot obtain the real-time acceleration information of Pursuer, it ultimately reflects in the change in the heading angle caused by the acceleration. The key to obtaining the heading angle lies in the change in the line-of-sight angle. In this paper, the change rate of the line-of-sight angle is specified to estimate the heading angle of Pursuer, and then the future position of Pursuer is inferred. Furthermore, by using the error between the predicted position and the real position, the agent adjusts its strategy through the reward function to adapt to the specified change rate of the line-of-sight angle.

For Agent D, its process reward consists of the NFZ penalty r_0 and the cooperation reward r_{col}^{D}, and is in the following form:

$$r_{pro}^{D} = r_0 + r_{col}^{D} \tag{42}$$

The cooperation reward is designed in the following form:

$$r_{col}^{D} = \frac{1}{\|\boldsymbol{r}_P - \boldsymbol{r}_D\|^2} \tag{43}$$

This means that the closer Defender is to Pursuer, the greater the reward value. By setting the cooperation reward item, the reward functions of Defender and Evader are no longer simply summed, but a deeper level of cooperation is achieved. Evader influences the prediction quality of Defender through its own actions, thus improving Defender's success rate in intercepting Pursuer. Moreover, by adjusting the coefficients in the reward function, the learning difficulty of individual goals and team goals can be balanced.

Therefore, the process reward is the sum of the process rewards of all agents, and its expression is as follows:

$$R_{pro} = r_{pro}^{E} + r_{pro}^{D} \tag{44}$$

5. Model Training and Simulation Verification

First, explicitly define all parameter settings in both the simulation environment and the improved algorithm. Second, demonstrate the superior performance of the proposed algorithm through comparative experiments with other mainstream reinforcement learning methods. Finally, visually present the training effectiveness using engagement trajectory plots, relative distance curves, acceleration profiles, and miss distance statistics, while discussing the robustness and generalization capability of the trained model.

5.1. Simulation Scenario and Algorithm Parameter Settings

Regarding the setting of simulation scenario parameters, to ensure that the trained model has good generalization, it is necessary to train in a random environment. Random initialization is mainly carried out for two parameters: position coordinates and heading angles, specifically, the parameters are randomly generated within a given range. The initial coordinates of Pursuer are (50, 50) m, the initial coordinates of Evader are (6450, 1150) m, and the initial coordinates of Defender are (6400, 1100) m. The initial positions of Evader, Defender, and Pursuer are all randomly generated within their respective rectangular ranges, and the initial heading angles are also randomly selected according to the ranges given in Table 1. The time step of the simulation is set to 0.01 s. When the relative distance between the missile and the target is less than 5 m and the ZEM distance is less than 2 m, it is regarded as a successful interception or defense. Otherwise, it is regarded as an interception or defense failure.

Table 1. Initial setting of simulation parameters.

Parameter	Pursuer	Evader	Defender
Position coordinates (m)	(50, 50) $\pm$ 50	(6450, 1150) $\pm$ 100	(6400, 1100) $\pm$ 100
Speed (m·s^{-1})	650	300	350
Heading angle (°)	25 $\pm$ 5	60 $\pm$ 5	20 $\pm$ 5
Maximum overload (m·s^{-2})	400	350	400

The parameter design in the reward function serves as a balancing mechanism to normalize reward magnitudes and ensure proper reward allocation. Well-tuned parameters can significantly enhance training efficiency while minimizing computational overhead during the learning process. The relevant parameter settings are shown in Table 2. The parameters for the NFZ in the environment and the target zone are shown in Table 3, while the hyperparameters in the algorithm are presented in Table 4.

Table 2. Simulation coefficient setting.

Parameter	Value
N	5
K	6
K_f	-1
c_1	1/8
c_2	5/8
c_3	3/8
α	1
λ	1
η	-1×10^3
ω	4

Table 3. Parameter settings for NFZ and target zone.

Zone Name	Center Coordinates (m)	Radius (m)
NFZ 1	(5650, 1000)	200
NFZ 2	(4800, −100)	350
NFZ 3	(2600, −400)	550
NFZ 4	(3800, 1800)	400
NFZ 5	(5050, 2300)	450
NFZ 6	(6700, 600)	350
Target zone	(2000, 1100)	150

Table 4. Parameter settings of the TP-MADDPG algorithm.

Parameter	Value
ε	1×10^{-6}
α	0.6
β_0	0.4
$\Delta\beta$	1×10^{-3}
Hidden layer dimension of LSTM network	128
Length of the sampled sequence	10
Learning rate of the Actor network	4×10^{-4}
Learning rate of the Critic network	5×10^{-4}
Reward discount factor	0.99
Training batch size	1024
Capacity of the experience replay buffer	1×10^{6}
Initial standard deviation of action noise	0.3
Minimum standard deviation of action noise	0.02
Exploring noise	0.3
Policy noise	0.2

The specific configuration of the experimental environment is as follows: Windows 11, CPU is Intel i5-14600KF, memory is approximately 48 GB, and GPU is NVIDIA GeForce RTX 4070.

5.2. Algorithm Training Process

During the training process, the total number of training steps is 1×10^6, and the maximum number of training steps in each episode is 1000. Meanwhile, comparative experiments are conducted with other mainstream multi-agent reinforcement learning algorithms. Each algorithm is trained with 3 different random seeds. The selection of random seeds among different algorithms remains consistent. The obtained reward values are smoothed and confidence intervals are added. The reward curves of different algorithms are shown in Figure 11.

The results demonstrate that only the improved TP-MADDPG algorithm achieves convergence to the target value within the specified training steps while maintaining policy stability in later stages, as evidenced by steady reward convergence. In contrast, the conventional MADDPG algorithm underperforms on time-dependent decision tasks, exhibiting significant reward oscillations during mid-training due to abrupt policy adjustments that lead to action–environment mismatch. The MAPPO algorithm initially shows positive rewards but suffers from policy divergence as agents increase environmental exploration, ultimately failing to form a stable policy network. Meanwhile, the MATD3 algorithm consistently yields suboptimal rewards without reaching the expected convergence threshold.

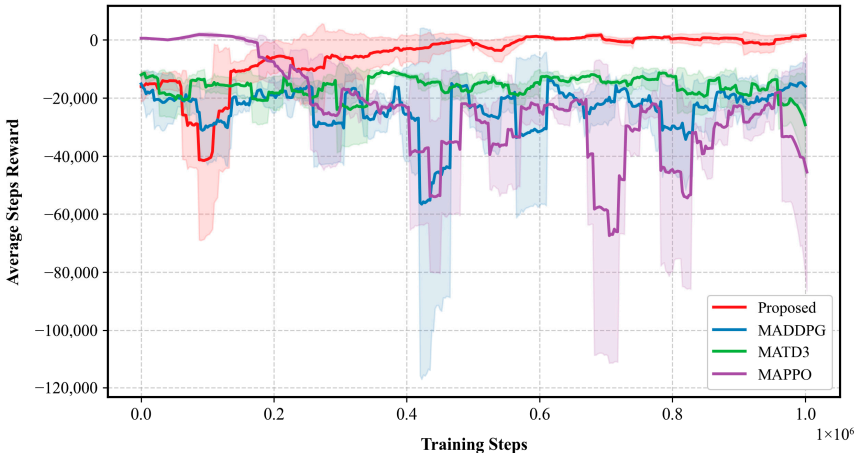


Figure 11. Comparison of reward curves of different algorithms.

Further analysis shows that due to the random initialization of the environment and the inherent instability of the multi-agent environment, all algorithms exhibit varying degrees of oscillation during the training process. The setting of the reward function in this paper places more emphasis on the penalty terms. The dominant factor of the positive reward is whether the final task is successful, that is, whether Defender successfully intercepts Pursuer and whether Evader reaches the target area. Therefore, the reward values are mostly in the negative range. In particular, the penalty for entering the NFZ is severe, which leads to the generation of large negative rewards during the training process of each algorithm.

Compared with the algorithm proposed in this paper, other multi-agent reinforcement learning algorithms have a slower convergence rate, and the reward values remain at a relatively low level throughout the training process. On the one hand, the multi-aircraft terminal guidance game problem in complex constraint scenarios depends on the sequential decision-making information of agents. That is, the previous actions of an agent can affect its own state and actions, as well as those of cooperative agents and the interceptor. Therefore, it has sequential dependence. It is necessary to introduce an LSTM network to emphatically establish this dependence and correlation. On the other hand, for the MADDPG and MATD3 algorithms, they do not have an experience prioritized replay mechanism. During the sampling process, the number of negative samples is so large that positive samples cannot be captured. At the same time, instability of the policy network may occur during training.

To verify the effectiveness of the core components within the TP-MADDPG algorithmic framework, this paper designs ablation experiments comparing the full TP-MADDPG algorithm with two of its key variants. The resulting reward curves are shown in Figure 12. The complete TP-MADDPG algorithm significantly outperforms the ablation variants in terms of convergence speed, final performance, and stability. Its reward curve consistently remains at the highest level throughout training, exhibiting smoothness and narrow confidence bands, indicating stable learning and low policy variance. The incorporation of the LSTM module enables agents to leverage historical information for forecasting the evolution of adversarial situations, thereby facilitating smooth decision-making during both cooperative guidance and homing phases. Meanwhile, the prioritized experience replay (PER) mechanism enhances training efficiency and sample utilization by prioritizing high-value experiences. Together, these components contribute to the acquisition of high-performance, robust policies.

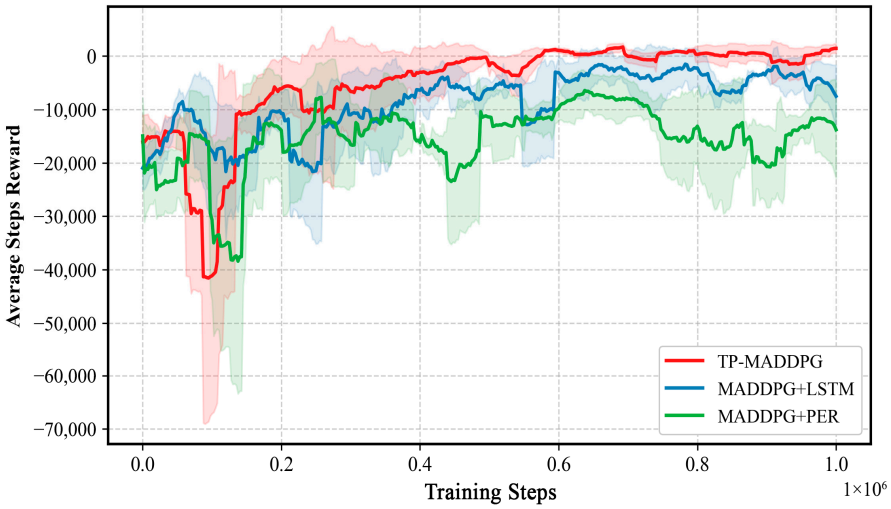
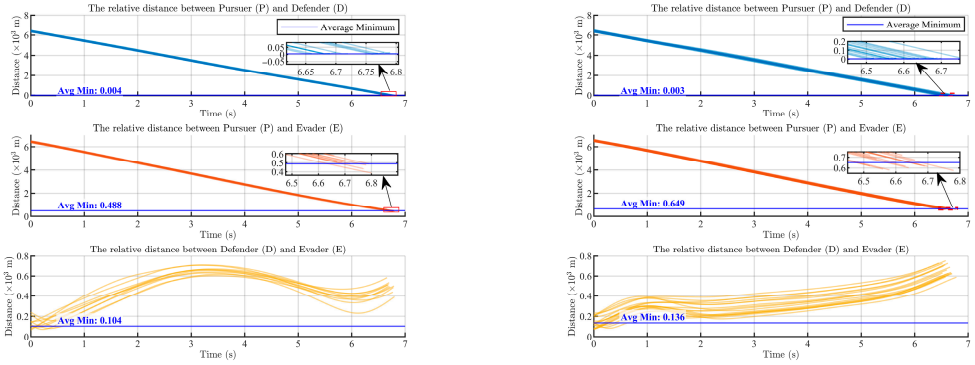


Figure 12. Comparison of reward curves among algorithm variants in the ablation study.

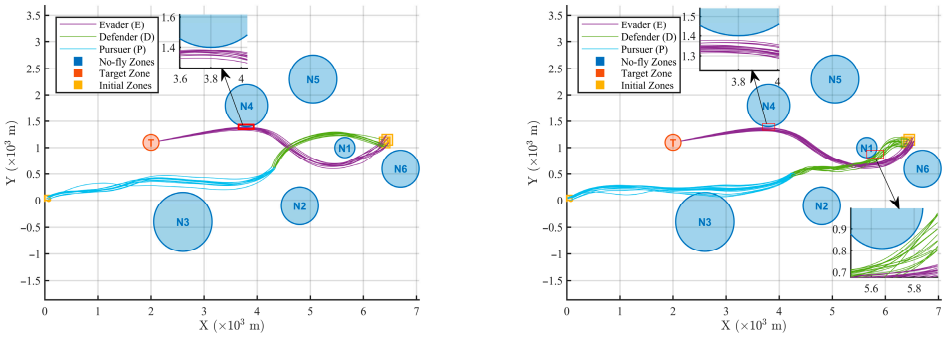
5.3. Analysis of Simulation Experimental Results

Based on the training results of the TP-MADDPG algorithm, the models with the highest reward values among all random seeds are selected for testing. A total of 2 representative models are chosen for testing, and the testing environment still maintains random initialization. Each model is tested 10 times, and the simulation results of different models are shown in Figures 13–16.



(**a**) (**b**)

Figure 13. The relative distances of aircraft in different models. (**a**) The relative distance in Model 1; (**b**) The relative distance in Model 2.



(**a**) (**b**)

Figure 14. Trajectories of aircraft in different models. (**a**) The aircraft trajectories in Model 1; (**b**) The aircraft trajectories in Model 2.

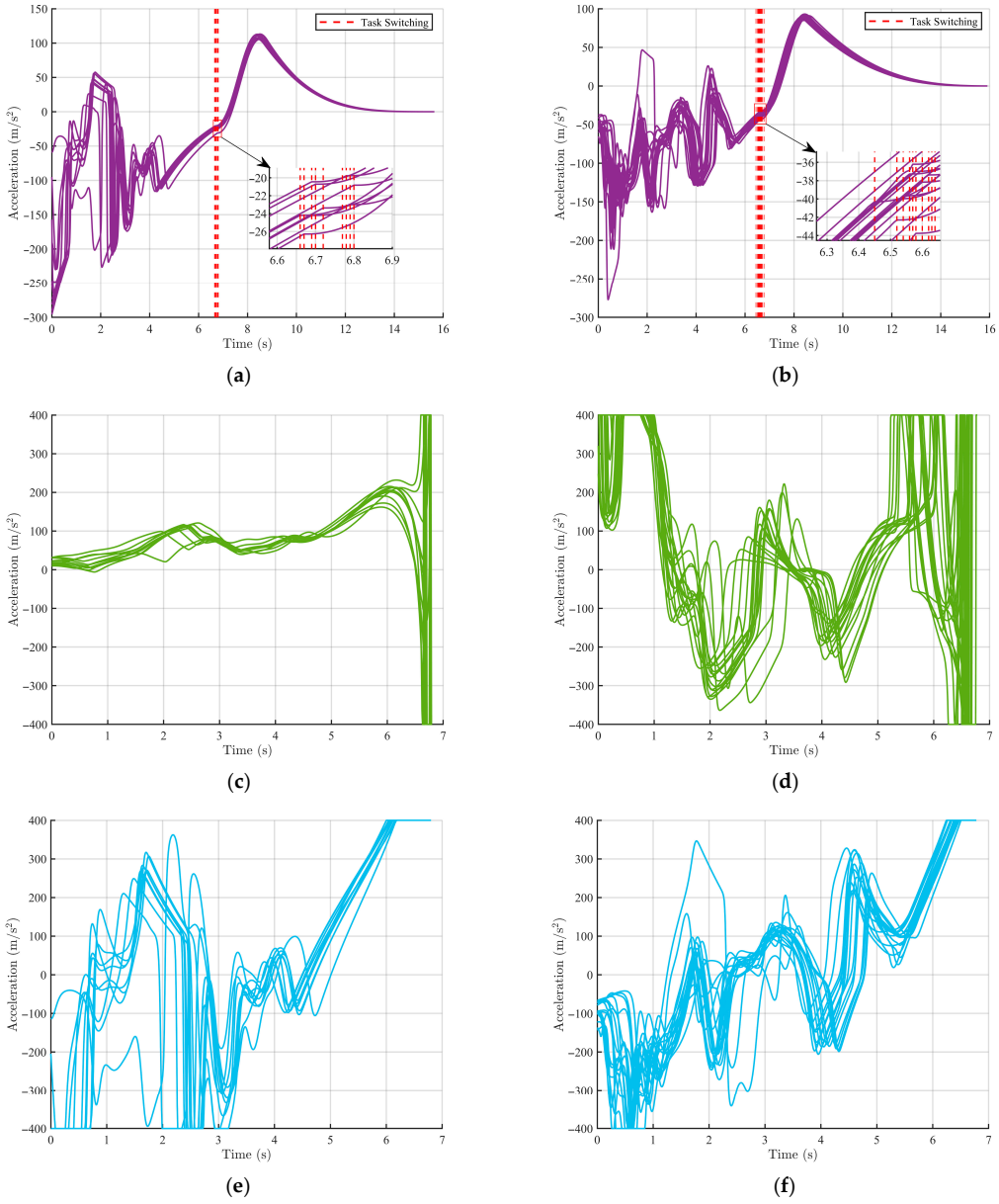


Figure 15. The overloads of aircraft in different models. (**a**) The overload of Evader in Model 1; (**b**) The overload of Evader in Model 2; (**c**) The overload of Defender in Model 1; (**d**) The overload of Defender in Model 2; (**e**) The overload of Pursuer in Model 1; (**f**) The overload of Pursuer in Model 2.

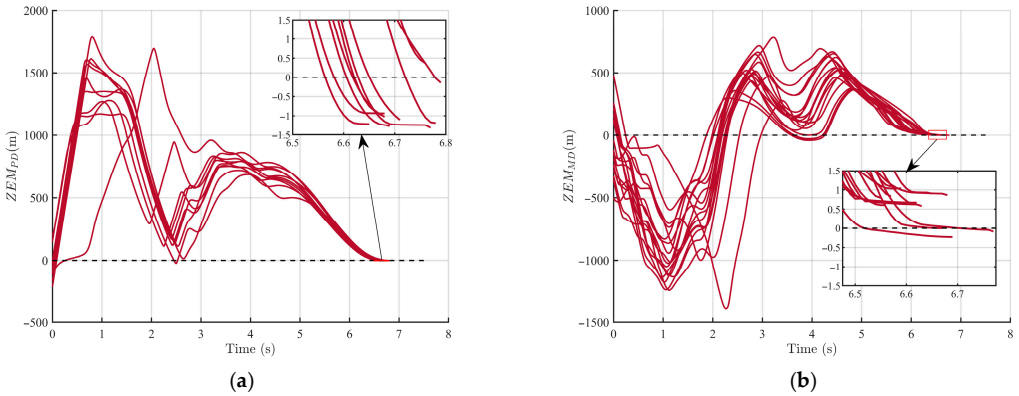


Figure 16. The zero-effort miss between Defender and Pursuer in different models. (**a**) The ZEM between Defender and Pursuer in Model 1; (**b**) The ZEM between Defender and Pursuer in Model 2.

The relative distance comparison chart is shown in Figure 13. Each chart consists of three sub-charts. From top to bottom, they are the relative distance between Pursuer and Defender, the relative distance between Pursuer and Evader, and the relative distance between Evader and Defender. In each sub-chart, the average value of the minimum values in all test rounds is calculated, and an indicator line is drawn to mark the specific value. As can be seen from the figure, the final relative distance between Pursuer and Defender is less than the set 5 m threshold, and the average value is shown as the blue indicator line in the figure. Moreover, when Defender successfully intercepts Pursuer, the relative distances between Evader and Pursuer and between Evader and Defender are both within the safe range.

The overall trajectory diagram is shown in Figure 14. By analyzing the test results of the two models, it can be seen that Evader and Defender can cooperate with each other on the premise of avoiding the NFZ, enabling Defender to successfully intercept Pursuer. Moreover, after Evader completes the cooperative guidance mission, it can continue to navigate to the target area. In addition, the agents have chosen different strategies, which is particularly evident in the trajectory diagram. Defender passes above and below the NFZ 1 respectively, and cooperates with Evader to finally successfully intercept Pursuer. In order to increase the interception success rate of Defender, Evader shows obvious maneuvering behavior. By maneuvering, Evader forces Pursuer to respond, which is shown in the figure as a change in the trajectories of the two towards the positive Y-axis. This reduces the error between the actual and predicted future positions of Pursuer, making it easier for Defender to intercept Pursuer.

For Model 1, in the early stage, Defender moves by slowly changing the direction of its initial velocity with a small overload. In the middle stage, it gradually increases the overload to change its moving direction and moves towards the predicted position of Pursuer. In the later stage, Defender cooperates with Evader, precisely adjusts its own overload, gradually achieves accurate prediction of the future position of Pursuer, and finally realizes a successful interception.

For Model 2, in the early stage, Defender changes the initial moving direction with a large overload. In the middle stage, while maintaining a safe distance from Evader and avoiding the NFZ, it gradually adjusts its moving direction. In the later stage, it successfully intercepts Pursuer. As for Evader, since it adopts a different luring maneuver strategy from Model 1 during the cooperative guidance mission phase, it needs a large overload to adjust its trajectory during the homing guidance mission. Nevertheless, the final result is still that it successfully avoids the NFZ and reaches the target area.

Through the comparative analysis of the two trajectory diagrams, it can be concluded that the agents have explored and learned relatively general strategies during training. For example, when dealing with the same NFZ, instead of avoiding it in a single direction or along a single path, they can simultaneously consider cooperation with another agent while evading. At the same time, when dealing with tasks with different time spans and constraint conditions, they can still exhibit good generalization ability and robustness. For instance, agent E can smoothly switch between the cooperative guidance task and the homing guidance task, that is, when performing the previous task, it needs to take into account the constraint conditions and success rate of the subsequent task.

Figure 15 are the overload figures of Evader, Defender, and Pursuer under different models. Evader needs to carry out the cooperative guidance task and the target homing guidance task, and the red dotted line indicates the switching moment between the two tasks. As can be seen from the figures, the overload of Evader triggers a response from Pursuer, which is a prerequisite for Evader to force Pursuer to reach the desired position. Through the restricted overload maneuver in the later stage, Evader causes the overload of

Pursuer to increase in a short time, and Pursuer maneuvers with the maximum overload. At the same time, in the early stage, Defender moves with a small overload by cooperating with Evader, and in the later stage, it maneuvers to the predicted position of Pursuer with the maximum overload to complete the cooperative guidance task.

By comparing the overload figures of the two models, the similarities are as follows. In the early stage, both Evader and Defender assign a relatively high weight to the constraint of avoiding the NFZ in terms of overload generation. In the middle stage, Evader conducts multiple small-scale maneuvers to dynamically adjust the error between the real and predicted positions of Pursuer. At the same time, it induces decision-making oscillations of Pursuer, accumulates its trajectory error, and reduces the probability of being intercepted. Defender, on the other hand, cooperates with Evader. During the process of continuously adjusting the overload, it gradually reduces the relative distance between itself and Pursuer. In the later stage, in order to increase the interception success rate of Defender against Pursuer, Evader forces Pursuer to maneuver with the maximum overload through its own maneuvers. At this time, it creates more suitable hitting conditions for Defender to achieve the goal of meeting the expected miss distance.

The difference between the two models lies in the variance of maneuvering methods resulting from the different overload directions generated by Defender. This is particularly evident in the overload graphs of Defender and Pursuer.

For the cooperative guidance mission, the average overloads of Evader, Defender, and Pursuer are statistically presented in Table 5. Regarding Model 1, Evader and Defender cooperate with each other and complete the mission with relatively small overloads, and the sum of their average overloads is less than the average overload of Pursuer. For Model 2, Evader adopts a smaller average overload compared to Model 1, but Defender needs to exert a greater overload to complete the cooperative guidance mission.

Table 5. The average overloads of aircraft in different models.

Aircraft	Average Overload ($m \cdot s^{-2}$)
Evader 1	83.012
Defender 1	87.391
Pursuer 1	194.111
Evader 2	75.271
Defender 2	166.835
Pursuer 2	150.672

The comparison chart of ZEM is shown in Figure 16. The test results indicate that the ZEM all meet the set 2 m threshold and are all below 1.5 m. For the training results of Model 1, the direct reason for the change in the ZEM in the 0–3 second interval is that Defender's avoidance of the NFZ reduces the relative velocity between Pursuer and Defender, thus increasing the ZEM. After Defender completes the avoidance, it gradually adjusts the overload to make the angle between the velocity directions of the two and the line-of-sight smaller, thereby increasing the relative velocity and decreasing the ZEM. Finally, with the cooperation of Evader, it precisely adjusts the overload to hit Pursuer and achieve the minimum ZEM. For the training results of Model 2, the change in the ZEM in the early stage is due to the constraint that Defender needs to avoid the NFZ, and the change in the middle and late stages is due to the maneuver adjustment in cooperation with Evader to ensure that the ZEM converges to the minimum value.

The model was tested 100 times, and the ZEM between Pursuer and Defender were statistically recorded to form the scatter plot shown in Figure 17. As can be seen from the figure, all ZEM are below 1.5 m, and the vast majority of them are below 0.8 m. This

conclusively demonstrates that the strategy of agent D achieves successful interception of Pursuer, confirming the proposed guidance law delivers superior guidance performance.

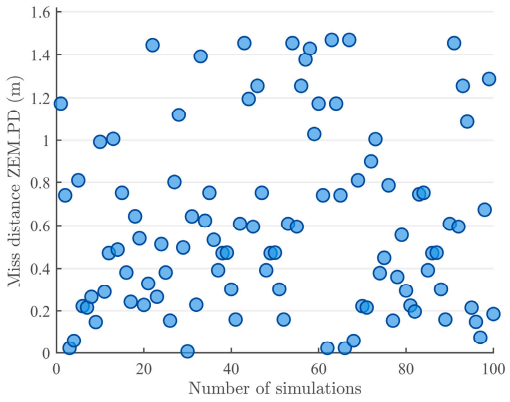


Figure 17. Scatter plot of ZEM statistics.

To further validate the superiority of the proposed TP-MADDPG algorithm in cooperative game guidance scenarios involving complex constraints, this section introduces conventional guidance methods as comparative benchmarks. Two classic approaches—Proportional Navigation (PN) and Model Predictive Control (MPC)—are selected to construct traditional guidance schemes tailored to the three-body adversarial scenario with multiple constraints considered in this paper. Specifically, the Pursuer continues to adopt Augmented Proportional Navigation (APN) to maintain consistency in guidance strategy; the Evader employs the Artificial Potential Field (APF) method to achieve NFZ avoidance and target-area seeking; and the Defender adopts PN fused with APF or MPC fused with APF to fulfill NFZ avoidance while coordinating interception of Pursuer. Comparative simulations between these benchmark schemes and the proposed TP-MADDPG algorithm are conducted under identical initial conditions and constraint settings.

Figure 18 shows the simulation results of cooperative game guidance using the conventional PN and MPC methods. In Figure 18a–d, the left column presents the simulation outputs of the PN fused with APF approach, while the right column shows those of the MPC fused with APF approach. Figure 18e,f display the acceleration profiles of Evader and Pursuer under the two traditional methods.

From the perspective of aircraft trajectories, the hybrid PN and MPC methods can basically achieve fundamental NFZ avoidance. However, evident shortcomings appear in the three-body cooperative adversarial scenario. With the PN–APF method, the Defender's trajectory adjustments lack dynamic adaptability to the Pursuer's maneuvers, resulting in low precision in capturing the interception window and ultimately failing to achieve effective interception of Pursuer. Although the MPC–APF method improves trajectory prediction for Pursuer through model-based forecasting, the coupling of multiple constraints expands the dimensionality of the prediction space, increasing computational latency and reducing the real-time performance of trajectory planning. Consequently, Defender's interception actions exhibit significant delays, and the interception task likewise fails. As a result, Pursuer successfully accomplishes effective interception of Evader.

Analysis of the acceleration characteristics reveals that, under the traditional methods, Evader's acceleration output is designed solely around APF-based obstacle avoidance and homing guidance, without accounting for cooperative guidance coupling with Defender. This leads to frequent abrupt changes in acceleration during the NFZ avoidance phase, with overload variation rates far exceeding those generated by the proposed TP-MADDPG algorithm.

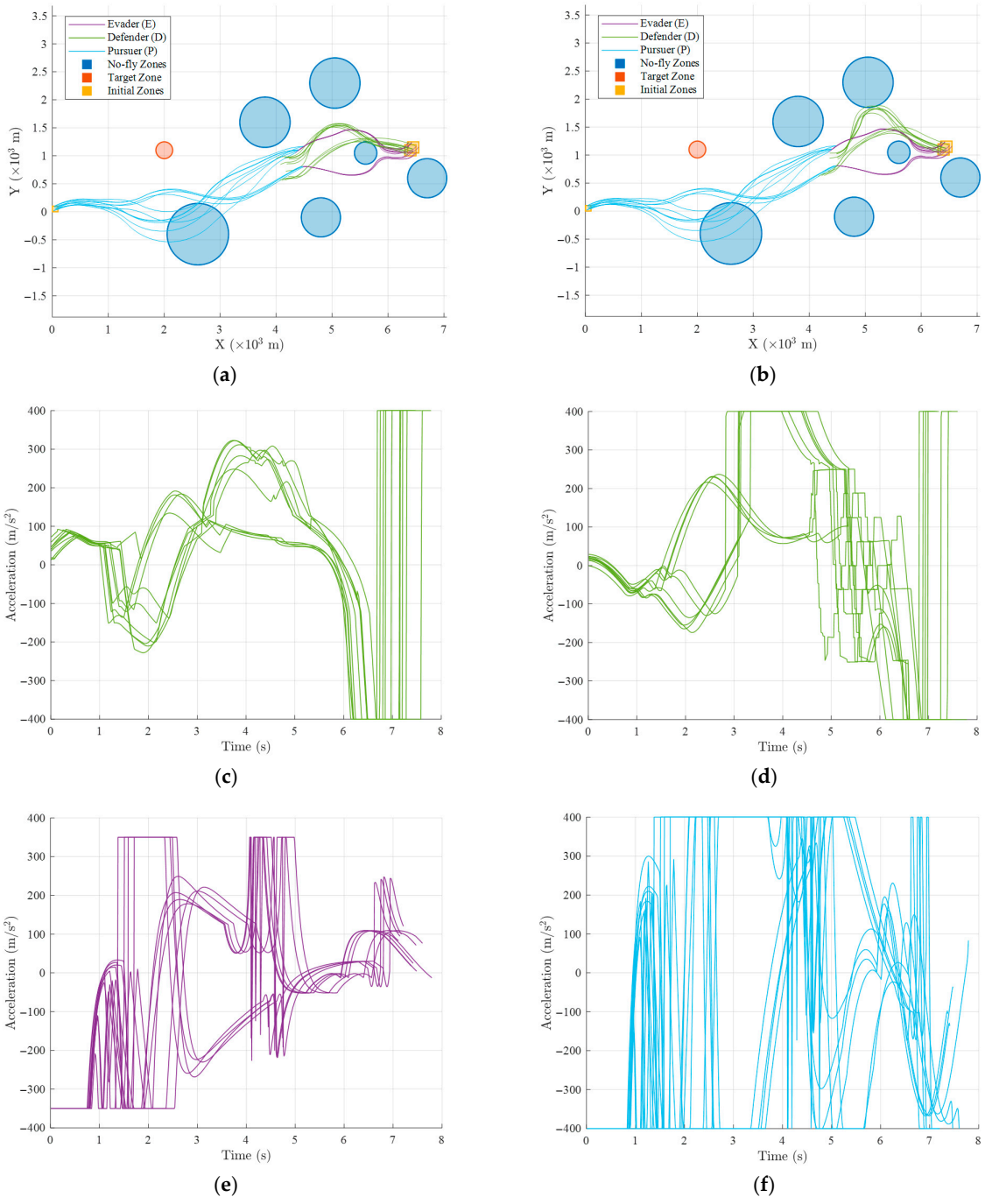


Figure 18. Performance comparison of conventional control methods. (**a**) Cooperative game trajectories under PN guidance law; (**b**) Cooperative game trajectories under MPC strategy; (**c**) Acceleration commands generated by the PN guidance law for the Defender; (**d**) Acceleration commands generated by the MPC Strategy for the Defender; (**e**) Acceleration commands of the Evader; (**f**) Acceleration commands of the Pursuer.

The above comparative results indicate that, although conventional PN and MPC methods can accomplish singular tasks such as obstacle avoidance or target seeking, they exhibit inherent deficiencies—including poor dynamic adaptability, insufficient real-time performance, and lack of cooperative strategies—when applied to complex scenarios characterized by three-body dynamic engagements, tightly coupled multi-constraints, and cross-task coordination. Consequently, these methods cannot effectively

balance the multi-objective requirements of NFZ avoidance, cooperative interception, and homing guidance.

5.4. Analysis of Parameter Sensitivity

To quantitatively evaluate the robustness and generalization capability of the proposed TP-MADDPG controller, this section presents a parameter sensitivity analysis, as illustrated in Figure 19. While keeping the trained policy fixed, we designed 12 test scenarios involving multi-dimensional parameter perturbations, including: system-wide velocity perturbation, perturbation of the proportional navigation gain of the Pursuer, variation in the maximum overload limit of the aircraft, geometric size and spatial position disturbances of the NFZ, and fully randomized parameter perturbation scenarios. All perturbations were applied within a $\pm 20\%$ range, with each parameter independently and uniformly sampled from the specified interval.

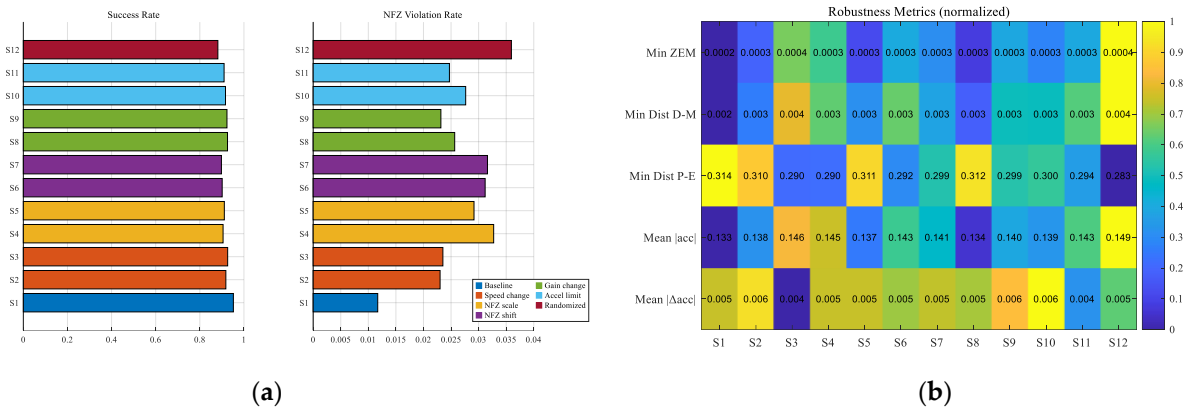


Figure 19. Parameter sensitivity analysis and generalization performance evaluation. (**a**) Task success rate and NFZ failure rate under different parameter configurations; (**b**) Multidimensional robustness metric heatmap analysis.

Figure 19a shows that the model maintains a high task success rate and an extremely low NFZ failure rate across the vast majority of configurations. This indicates that the policy is not overfitted to a specific set of training parameters, but has instead learned generalizable capabilities to cope with diverse adversarial situations and constraint configurations. Furthermore, Figure 19b provides a quantitative assessment across multiple key performance dimensions. The heatmap demonstrates that all metrics remain stable under different configurations. Collectively, these results confirm that the policy generated by the controller is not only effective, but also exhibits strong stability and adaptability, with a high degree of resilience to uncertainties in the dynamic model, variations in environmental parameters, and fluctuations in actuator constraints.

6. Conclusions

This paper proposes and validates a cooperative game guidance framework based on multi-agent reinforcement learning for terminal guidance of multiple aircraft under complex constraints. The complex constraints are characterized by four aspects: homing guidance, NFZ avoidance, collision prevention, and cooperative guidance. The core contributions of this research lies in two key aspects: First, it integrates the highly coupled cooperative game guidance task with NFZ avoidance, formulating a novel complex cooperative guidance mission while addressing smooth task-switching between cooperative guidance and homing guidance phases, thereby achieving cross-task constraint propagation. This integration significantly increases mission complexity and decision-making difficulty. Second, departing from conventional approaches that treat heterogeneous-role

aircraft as a single agent, our work models them as independent agents with differentiated policy networks, effectively capturing and optimizing emergent cooperative strategies in complex adversarial scenarios. The autonomous cooperative decision-making enabled by these heterogeneous policy networks substantially enhances mission adaptability under complex constraints.

The simulation results demonstrate that under varying initial conditions, Defender can successfully intercept Pursuer with minimal miss distance through cooperation with Evader, while maintaining overall lower acceleration demands than Pursuer. Meanwhile, Evader smoothly transitions and accomplishes its homing guidance mission after completing the cooperative guidance task. These findings validate that the proposed intelligent cooperative game guidance law exhibits remarkable performance robustness, generalization capability, and environmental adaptability. Under complex constraints, Evader and Defender autonomously develop cooperative behaviors based on learned intelligent policies to achieve their respective mission objectives, demonstrating emergent intelligent effects. This confirms the effectiveness and advancement of the proposed method in addressing highly dynamic, multi-constrained terminal guidance game problems.

This study is based on simplified dynamical assumptions of two-dimensional planar motion with constant velocities, focusing on validating the core algorithmic mechanisms of multi-agent cooperative game guidance. However, differences exist compared with real-world aerospace adversarial scenarios characterized by highly coupled three-dimensional nonlinear dynamics. Factors such as velocity degradation under high-overload maneuvers, actuator dynamic response delays, and aerodynamic constraints—common in practical engineering applications—have not been considered, which constitutes the primary limitation of this work. Specifically, velocity attenuation induced by high-overload maneuvers alters the kinematic characteristics of the aircraft, causing the evolution of relative distances and line-of-sight angles to deviate from constant-speed features. This increases the difficulty of trajectory prediction for the pursuer. However, the velocity variation follows intrinsic laws that can be fitted via linear or nonlinear models, allowing agents to capture these regularities through temporal feature learning and adaptively adjust their strategies. Regarding three-dimensional dynamics, gravity components and altitude-varying atmospheric density introduce additional dimensions to the state space, thereby increasing the complexity of policy learning. Moreover, variable atmospheric density necessitates the incorporation of corresponding aerodynamic constraints to ensure consistency with the actual maneuverability of the aircraft.

Future research will take integrated guidance-control design as the central direction, fully incorporating engineering factors such as three-dimensional nonlinear dynamics, aerodynamic constraints, actuator dynamics, and high-overload-induced velocity decay into the multi-agent reinforcement learning framework. Moreover, the multi-agent cooperative game guidance problem under such complex constraints will be treated as a key research node within the integrated guidance-control system design, with further exploration of its integration with aircraft autonomous decision-making, situational awareness, and other functional modules.

Author Contributions: Conceptualization, Y.G. and F.L.; methodology, F.L.; software, F.L.; validation, F.L.; formal analysis, Z.L.; investigation, J.W.; resources, S.W.; data curation, J.W.; writing—original draft preparation, F.L.; writing—review and editing, S.W.; visualization, F.L.; supervision, Y.G.; project administration, Y.G.; funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by National Natural Science Foundation (NNSF) of China under Grant 62473375, 62073335, 62303484, the Science fund for Distinguished Young People in Shaanxi Province under grant no. 2022JC-42.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

Abbreviations

The following abbreviations are used in this manuscript:

NFZ	No-fly Zones
LSTM	Long Short-Term Memory
ZEM	Zero-effort Miss

References

1. Chen, Z.Y.; Yu, J.L.; Dong, X.W.; Ren, Z. Three-dimensional cooperative guidance strategy and guidance law for intercepting highly maneuvering target. *Chin. J. Aeronaut.* **2021**, *34*, 485–495. [CrossRef]
2. Guo, Y.; Zhang, S.; Wang, S.B.; Wang, S.C.; Liu, Z.G. Defending an interceptor with lure role scenario: A finite-time cooperative guidance method. *Proc. Inst. Mech. Eng. Part G-J. Aerosp. Eng.* **2024**, *238*, 1393–1405. [CrossRef]
3. Yu, J.L.; Dong, X.W.; Li, Q.D.; Ren, Z.; Lv, J.H. Cooperative guidance strategy for multiple hypersonic gliding vehicles system. *Xi'an Chin. J. Aeronaut.* **2020**, *33*, 990–1005. [CrossRef]
4. Lu, J.-H.; Yu, J.-L.; Dong, X.-W. New Progress in Cooperative Guidance for Aircraft Swarm System. *Acta Autom. Sin.* **2025**, *51*, 727–743. [CrossRef]
5. 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]
6. Ratnoo, A.; Ghose, D. Impact Angle Constrained Guidance Against Nonstationary Nonmaneuvering Targets. *J. Guid. Control Dyn.* **2010**, *33*, 269–275. [CrossRef]
7. Harl, N.; Balakrishnan, S.N. Impact Time and Angle Guidance With Sliding Mode Control. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 1436–1449. [CrossRef]
8. Shi, Y.X.; Hu, Q.L.; Li, D.Y.; Lv, M.L. Adaptive Optimal Tracking Control for Spacecraft Formation Flying With Event-Triggered Input. *Ieee Trans. Ind. Inform.* **2023**, *19*, 6418–6428. [CrossRef]
9. Boyell, R.L. Defending a Moving Target Against Missile or Torpedo Attack. *IEEE Trans. Aerosp. Electron. Syst.* **1976**, *AES-12*, 522–526. [CrossRef]
10. Shima, T. Optimal cooperative pursuit and evasion strategies against a homing missile. *J. Guid. Control Dyn.* **2011**, *34*, 414–425. [CrossRef]
11. Rubinsky, S.; Gutman, S. Three-Player Pursuit and Evasion Conflict. *J. Guid. Control Dyn.* **2014**, *37*, 98–110. [CrossRef]
12. Wang, S.; Guo, Y.; Wang, S.; Liu, Z.; Zhang, S. Cooperative optimal guidance method for multi-aircraft with luring role. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 283–297. [CrossRef]
13. Shi, H.; Zhu, J.H.; Kuang, M.C.; Yuan, X.M. Cooperative prediction guidance law in target-attacker-defender scenario. *Sci. China-Inf. Sci.* **2021**, *64*, 149201. [CrossRef]
14. Garcia, E.; Casbeer, D.W.; Pachter, M. Design and Analysis of State-Feedback Optimal Strategies for the Differential Game of Active Defense. *Ieee Trans. Autom. Control* **2019**, *64*, 553–568. [CrossRef]
15. Sun, Q.L.; Shen, M.H.; Gu, X.L.; Hou, K.; Qi, N.M. Evasion-Pursuit Strategy against Defended Aircraft Based on Differential Game Theory. *Int. J. Aerosp. Eng.* **2019**, *2019*, 7980379. [CrossRef]
16. Wei, Y.S.; Liu, T.X.; Wei, C. Mode-switching cooperative defense strategy for the orbit pursuit-evasion-defense game. *Def. Technol.* **2025**, *44*, 272–286. [CrossRef]
17. Sinha, A.; Kumar, S.R.; Mukherjee, D. Three-agent Time-constrained Cooperative Pursuit-Evasion. *J. Intell. Robot. Syst.* **2022**, *104*, 28. [CrossRef]
18. Wang, K.; Mu, C.X. Learning-Based Control With Decentralized Dynamic Event-Triggering for Vehicle Systems. *Ieee Trans. Ind. Inform.* **2023**, *19*, 2629–2639. [CrossRef]
19. Wang, Y.; Chao, T.; Wang, S.; Yang, M. Trajectory tracking control of hypersonic vehicle considering modeling uncertainty. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2019**, *233*, 4779–4787. [CrossRef]
20. Xie, Y.; Liu, L.; Tang, G.; Zheng, W. Highly constrained entry trajectory generation. *Acta Astronaut.* **2013**, *88*, 44–60. [CrossRef]
21. Gao, Y.; Cai, G.B.; Yang, X.G.; Hou, M.Z. Improved Tentacle-Based Guidance for Reentry Gliding Hypersonic Vehicle With No-Fly Zone Constraint. *IEEE Access* **2019**, *7*, 119246–119258. [CrossRef]
22. Wang, T.; Zhang, H.B.; Tang, G.J. Predictor-corrector entry guidance with waypoint and no-fly zone constraints. *Acta Astronaut.* **2017**, *138*, 10–18. [CrossRef]

23. Zhao, D.J.; Song, Z.Y. Reentry trajectory optimization with waypoint and no-fly zone constraints using multiphase convex programming. *Acta Astronaut.* **2017**, *137*, 60–69. [CrossRef]
24. Huang, R.; Cui, N.-G.; Wei, C.-Z.; Wang, J.-B.; Huang, P.-X. Rapid trajectory optimization for cooperative penetration of missile formation based on two-stage RRT-GMP strategy. *J. Chin. Inert. Technol.* **2015**, *23*, 356–362. [CrossRef]
25. Yu, W.B.; Chen, W.C.; Jiang, Z.G.; Zhang, W.Q.; Zhao, P.L. Analytical entry guidance for coordinated flight with multiple no-fly-zone constraints. *Aerosp. Sci. Technol.* **2019**, *84*, 273–290. [CrossRef]
26. Li, Q.; Yu, J.; Hua, Y.; Dong, X.; Li, Q.; Lu, J.; Ren, Z. Cooperative Time-Guidance Strategy for Multihypersonic Vehicles With Irregular No-Fly Zones: A Transformer-Based Real-Time Planning Method. *IEEE Trans. Aerosp. Electron. Syst.* **2025**, *61*, 1283–1297. [CrossRef]
27. Wang, X.T.; Yang, M.; Wang, S.Y.; Hou, M.Z.; Chao, T. Linear-quadratic and norm-bounded combined differential game guidance scheme with obstacle avoidance for attacking defended aircraft in three-player engagement. *Def. Technol.* **2024**, *42*, 136–155. [CrossRef]
28. Shalumov, V. Cooperative online Guide-Launch-Guide policy in a target-missile-defender engagement using deep reinforcement learning. *Aerosp. Sci. Technol.* **2020**, *104*, 105996. [CrossRef]
29. Chen, Z.; Wei, W.; Chen, W. Reinforcement Learning-based Intelligent Guidance Law for Cooperative Attack of Multiple Missiles. *Acta Armamentarii* **2021**, *42*, 1638–1647. [CrossRef]
30. English, J.T.; Wilhelm, J.P. Defender-Aware Attacking Guidance Policy for the Target-Attacker-Defender Differential Game. *J. Aerosp. Inf. Syst.* **2021**, *18*, 366–376. [CrossRef]
31. Gong, X.P.; Chen, W.C.; Chen, Z.Y. Intelligent Game Strategies in Target-Missile-Defender Engagement Using Curriculum-Based Deep Reinforcement Learning. *Aerospace* **2023**, *10*, 133. [CrossRef]
32. Ni, W.; Wang, Y.; Xu, C.; Chi, F.; Liang, H. Cooperative game guidance method for hypersonic vehicles based on reinforcement learning. *Acta Aeronaut. Astronaut. Sin.* **2023**, *44*, 729400. [CrossRef]

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.