mathematics

MDPI

Article

A Multi-Population Mean-Field Game Approach for Large-Scale Agents Cooperative Attack-Defense Evolution in High-Dimensional Environments †

Guofang Wang [1,2,3], Ziming Li [1,2], Wang Yao [2,3,4,*] and Sikai Xia [1,2]

1 School of Mathematical Sciences, Beihang University, Beijing 100191, China
2 Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beijing Advanced Innovation Center for Big Data and Brain Computing, Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beihang University, Beijing 100191, China
3 Peng Cheng Laboratory , Shenzhen 518055, China
4 Institute of Artificial Intelligence, Beihang University, Beijing 100191, China
* Correspondence: yaowang@buaa.edu.cn
† This paper is an extended version of our paper published in Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO'22), Association for Computing Machinery, New York, NY, USA, 9–13 July 2022.

Abstract: As one of the important issues of multi-agent collaboration, the large-scale agents' cooperative attack–defense evolution requires a large number of agents to make stress-effective strategies to achieve their goals in complex environments. Multi-agent attack and defense in high-dimensional environments (3D obstacle scenarios) present the challenge of being able to accurately control high-dimensional state quantities. Moreover, the large scale makes the dynamic interactions in the attack and defense problems increase dramatically, which, using traditional optimal control techniques, can cause a dimensional explosion. How to model and solve the cooperative attack–defense evolution problem of large-scale agents in high-dimensional environments have become a challenge. We jointly considered energy consumption, inter-group attack and defense, intra-group collision avoidance, and obstacle avoidance in their cost functions. Meanwhile, the high-dimensional state dynamics were used to describe the motion of agents under environmental interference. Then, we formulated the cooperative attack–defense evolution of large-scale agents in high-dimensional environments as a multi-population high-dimensional stochastic mean-field game (MPHD-MFG), which significantly reduced the communication frequency and computational complexity. We tractably solved the MPHD-MFG with a generative-adversarial-network (GAN)-based method using the MFGs' underlying variational primal–dual structure. Based on our approach, we carried out an integrative experiment in which we analytically showed the fast convergence of our cooperative attack–defense evolution algorithm by the convergence of the Hamilton–Jacobi–Bellman equation's residual errors. The experiment also showed that a large number of drones can avoid obstacles and smoothly evolve their attack and defense behaviors while minimizing their energy consumption. In addition, the comparison with the baseline methods showed that our approach is advanced.

Keywords: large-scale agents; attack and defense; multi-population mean-field game; high-dimensional solution space; neural networks

MSC: 91A16; 93-10; 49N80

Citation: Wang, G.; Li, Z.; Yao, W.; Xia, S. A Multi-Population Mean-Field Game Approach for Large-Scale Agents Cooperative Attack-Defense Evolution in High-Dimensional Environments. *Mathematics* **2022**, *10*, 4075. https://doi.org/10.3390/math10214075

Academic Editor: Ioannis G. Tsoulos

Received: 28 September 2022
Accepted: 31 October 2022
Published: 2 November 2022

1. Introduction

Cooperative control among multiple agent has always been an important research topic in swarm intelligence [1]. Game theory, as a branch of modern mathematics, studies optimal decision-making problems under conflicting adversarial conditions and can provide a

theoretical basis for multi-agent cooperative decision-making problems. This paper is oriented toward the large-scale agents' cooperative attack–defense evolution, one of the important issues of multi-agent collaboration, which requires a large number of agents to make stress-effective strategies to achieve their goals in complex environments [2]. The multi-player continuous attack–defense game, as the mainstream research method for multi-agent attack–defense evolution, is a differential game between two adversarial teams of cooperative players playing in an area with targets. The attacker attempts to reach the set destination. The goal of the defender is to delay or stop the attacker by catching it [3]. The attack–defense game problem can be described as an optimal decision-making problem under complex multi-constraint conditions [4].

The classic attack–defense games have long been studied in multi-agent cooperative control. The multiple-pursuer-one-evader problem has been well documented. In [5], the Voronoi diagram construct was used for the capture of an evader within a bounded domain. Based on differential game theory and optimal control theory, differential game models have been established to solve optimal strategies by setting some rules and assumptions. In [6], based on the geometric relationship between two pursuers and an evader, a differential game model was established through coordinate transformation to solve the optimal cooperative strategy. Paper [7] studied the optimal guidance law of two missiles intercepting a single target based on the hypothesis of the missile hit sequence. Reference [8] proposed an online decision-making technique based on deep reinforcement learning (DRL) for solving the environmental sensing and decision-making problems in the chase and escape games. For the case of N-pursuers and M-evaders, more complex interactions need to be analyzed. Paper [9] studied a multiplayer border-defense problem and extended classical differential game theory to simultaneously address weapon assignments and multiplayer pursuit–evasion scenarios. Papers [10,11] proposed some methods based on linear programming, and applied deep reinforcement learning methods to deal with a simplified version of the RoboFlag competition [4]. An approach to the task allocation of many agents was proposed in [12], where Bakolas and Tsiotras used the Voronoi diagram construct to control the system. To solve general attack–defense games, the Hamilton–Jacobi–Isaacs (HJI) approach is ideal when the game is low-dimensional. However, because its complexity increases exponentially with the number of agents, the HJI approach is only tractable for the two-player game [13]. However, the above methods cannot be directly applied to the large-scale attack–defense game we are concerned with, because these traditional optimization and control technologies deal with the dynamic interactions between individuals separately. Moreover, to conduct more accurate real-time control for agents, the state variables used to characterize their kinematics are usually high-dimensional. Thus, with the increase in the agents' number, the modeling process of cooperative attack–defense problems tends to be complex, and the difficulty of solving the optimal strategy will increase significantly.

To solve the communication and calculation difficulties caused by agents' interactions on a large scale, mean-field games (MFGs) were proposed by Lasry and Lions [14–16] and Huang, Malhame, and Caines [17–19] independently. MFGs have been widely used in industrial engineering [20–22], crowd motion [23–26], swarm robotics [27,28], epidemic modeling [29,30], and data science [31–33]. In the MFG, each agent can obtain the evolution of the global or macroscopic information (mean-field) by solving the Fokker–Planck–Kolmogorov (FPK) equation. The optimal strategy of each agent is found by solving the Hamilton–Jacobi–Bellman (HJB) equation [34]. Recently, a machine-learning-based method, named APAC-net, has been proposed to solve high-dimensional stochastic MFGs [35]. Intuitively, the large-scale attack–defense problems are more consistent with the multi-population model. The multi-population mean-field game is a critical subclass of mean-field games (MFGs). It is a theoretically feasible multi-agent model for simulating and analyzing the game between multiple heterogeneous populations of interacting massive agents. We proposed a numerical solution method (CA-Net) for multi-population high-dimensional stochastic MFGs in [36], and studied the large-scale attack–defense problem in a 3D blank scenario based on CA-Net

in [37]. In this paper, we focus on a 3D obstacle scene. The presence of multiple obstacles in the 3D space brings qualitative changes to the attack and defense decisions of large-scale agents, i.e., a "diversion" phenomenon. How to model and solve the cooperative attack–defense evolution problem of large-scale agents in 3D obstacle scenarios has become a challenge.

Inspired by the above-mentioned cutting-edge works, in this paper, we jointly considered energy consumption, inter-group attack and defense, intra-group collision avoidance, and obstacle avoidance in their cost functions. Meanwhile, the high-dimensional state dynamics were used to describe the motion of agents under environmental interference. Then, we made the following main contributions:

- We formulated the cooperative attack–defense evolution of large-scale agents in high-dimensional environments as a multi-population high-dimensional stochastic mean-field game (MPHD-MFG), which significantly reduced the communication frequency and computational complexity.
- We propose ECA-Net, an extended nonlinear coupled alternating neural network composed of multiple generators and multiple discriminators. We tractably solved the MPHD-MFG with the ECA-Net algorithm using MFGs' underlying variational primal–dual structure.
- We carried out an integrative experiment in which we analytically showed the fast convergence of our cooperative attack–defense evolution algorithm by the convergence of the Hamilton–Jacobi–Bellman equation's residual errors. The experiment also showed that a large number of drones can avoid obstacles and smoothly evolve their attack and defense behaviors while minimizing their energy consumption. The comparison with the baseline methods showed that our approach is advanced.

Our approach accomplishes a breakthrough from few-to-few to mass-to-mass in terms of attack–defense game theory in 3D obstacle scenarios.

In Section 2, we model the cooperative attack–defense evolution of large-scale agents in a 3D obstacle scene and formulate it as an MPHD-MFG. In Section 3, we propose ECA-Net to tractably solve the MPHD-MFG. Section 4 shows the performance of our algorithm with numerical results, and in Section 5, we draw our conclusions.

2. Modeling and Formulating

The system model consists of the objective function and the kinematics equation, which describe how large-scale agents evolve paths through attack–defense relationships. We considered a 3D obstacle attack–defense scenario with N agents, as shown in Figure 1. The blue side $\mathbb{N}_1$ as the attacker with N_1 agents hopes to break through the red side's interception and successfully reach the destination; the red side $\mathbb{N}_2$ as the defender with N_2 agents hopes to complete the interception against the blue side in the given area to prevent the blue side from penetrating. The state of agent i is denoted by $\mathbf{x}_i^p(t) \in \mathbb{R}^n$; $p = 1$ represents the blue side; $p = 2$ represents the red side, $i \in \mathbb{N}_1$ or $\mathbb{N}_2$, and $N_1 + N_2 = N$.

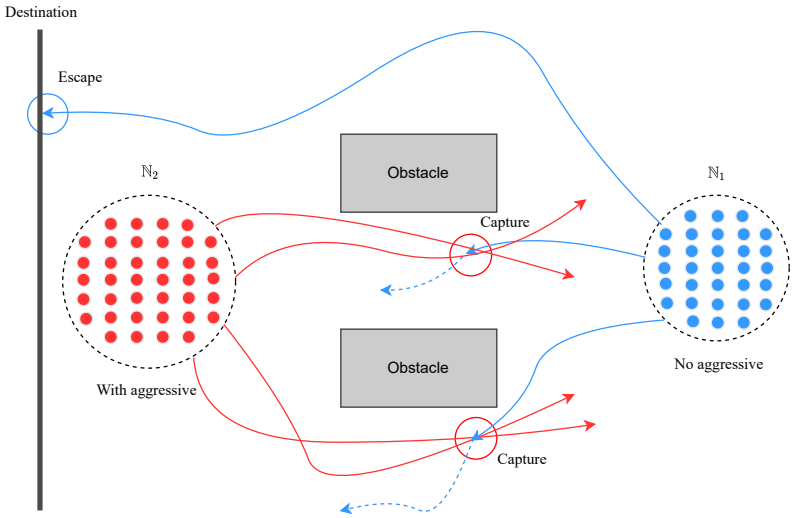


Figure 1. Vertical view of large-scale agents attacking and defending in 3D obstacle scene.

2.1. Kinematics Equation

Each agent $i \in \mathbb{N} = \mathbb{N}_1 \cup \mathbb{N}_2$ belongs to a population $p(i) \in \mathbb{Q} = \{1,2\}$ and is characterized by a state $\mathbf{x}_i^p(t) \in \mathbb{R}^n$ at time $t \in [0,T]$. We considered that the continuous-time state $\mathbf{x}_i^p(t)$ of player i of population p has the dynamic–kinematic equation of the following form

$$\mathrm{d}\mathbf{x}_i^p = \mathbf{h}_i^p(\mathbf{x}_i^p, \mathbf{x}_i^{p^-}, \mathbf{x}_i^{-p^-}, \mathbf{u}_i^p)\mathrm{d}t, \tag{1}$$

where $\mathbf{u}_i^p : [0,T] \to \mathcal{U}_p \subseteq \mathbb{R}^m$ is the control input (strategy) implemented by agent i in view of the state $\mathbf{x}_i^p$ at time t, $\mathbf{x}_i^{p^-} : \mathbb{R}^n \times [0,T] \to \mathcal{P}_2(\mathbb{R}^n)$ is the states of all other intra-group agents, $\mathbf{x}_i^{-p^-}$ describes the states of all other inter-group agents, and $\mathbf{h}_i^p : \mathbb{R}^n \times \mathcal{P}_2(\mathbb{R}^n)^2 \times \mathcal{U}_p \to \mathbb{R}^n$ is the nonlinear evolution function, $p = 1,2$.

2.2. Objective Function

For player i of population p, $p = 1,2$, we considered the cost function of the following form:

$$\begin{aligned} J_i^p(\mathbf{x}_i^p, \mathbf{x}_i^{p^-}, \mathbf{x}_i^{-p^-}, \mathbf{u}_i^p) = \int_0^T L_i^p(\mathbf{x}_i^p(t), \mathbf{u}_i^p(t)) + F_i^p(\mathbf{x}_i^p(t), \mathbf{x}_i^{p^-}(\mathbf{x}_i^p(t), t), \mathbf{x}_i^{-p^-}(\mathbf{x}_i^{-p}(t), t))\mathrm{d}t \\ + G_i^p(\mathbf{x}_i^p(T), \mathbf{x}_i^{p^-}(\mathbf{x}_i^p(T), T), \mathbf{x}_i^{-p^-}(\mathbf{x}_i^{-p}(T), T)), \end{aligned} \tag{2}$$

where $L_i^p : \mathbb{R}^n \times \mathcal{U}_p \to \mathbb{R}$ is the running cost incurred by agent i based solely on its actions, $F_i^p : \mathbb{R}^n \times \mathcal{P}_2(\mathbb{R}^n)^2 \to \mathbb{R}$ is the running cost incurred by agent i based on its interactions with the rest of the same population and the other population, and $G_i^p : \mathbb{R}^n \times \mathcal{P}_2(\mathbb{R}^n)^2 \to \mathbb{R}$ is a terminal cost incurred by agent i based on its final state and the final states of the whole of the related populations.

2.2.1. Blue-Side Control Problem

In the 3D obstacle space, the blue side's agents traveling from a common source hope to break through the red side's interception while avoiding obstacles and successfully reach the destination, where a straight line segment in the area above the red side's initial distribution is set as the destination. At time $t \geq 0$, the i-th agent controls its strategy $\mathbf{u}_i^1$ to minimize its: (1) motion energy, (2) red side capture, (3) blue side inter-agent collision,

(4) collision of the blue side with the obstacles, and (5) travel time during the remaining travel to the destination. The running cost L_i^1 is given by

$$L_i^1 = \underbrace{c_1 \|\mathbf{u}_i^1(t)\|^2}_{\text{(1) motion energy minimization}}, \tag{3}$$

where c_1 is a constant. The running cost L_i^1 denotes the control effort implemented by agent i. The interaction cost F_i^1 is given by

$$\begin{aligned} F_i^1 = & \underbrace{c_2\left(-\frac{1}{N_2}\sum_{k=1}^{N_2}\left\|\mathbf{p}_i^1(t) - (\mathbf{p}_k^2(t) + e_r\mathbf{e})\right\|\right)}_{\text{(2) red side capture minimization}} + \underbrace{c_3\left(\frac{1}{N_1}\sum_{j\neq i}^{N_1}\mathbf{1}_{\|\mathbf{p}_i^1(t)-\mathbf{p}_j^1(t)\|\leq e_0}\right)}_{\text{(3) blue side inter-agent collision avoidance}} \\ & + \underbrace{c_4\left(\frac{1}{N_1}\sum_{i=1}^{N_1}\begin{cases}\sum_{a=1}^{A}\gamma_{obs,a}\frac{1}{Q_a(\mathbf{p}_i,t)} & \text{if } \mathbf{p}_i \in \Omega_{obs,trn} \\ 0 & \text{otherwise}\end{cases}\right)}_{\text{(4) blue side obstacle collision avoidance}}, \end{aligned} \tag{4}$$

$$\Omega_{obs,trn} = \bigcup_{a=1}^{A}\Omega_{obs,trn,a} \tag{5}$$

$$\Omega_{obs,trn,a}: \quad Q_a(x,y,z) = \frac{1}{3v_{1,a}^2}(x-x_{0,a})^2 + \frac{1}{3v_{2,a}^2}(y-y_{0,a})^2 + \frac{1}{3v_{3,a}^2}(z-z_{0,a})^2 \leq 1.1, \quad a = 1,\ldots,A, \tag{6}$$

$$\Omega_{obs,a} = \{(x,y,z)||x-x_{0,a}| \leq v_{1,a}, |y-y_{0,a}| \leq v_{2,a}, |z-z_{0,a}| \leq v_{3,a}\}, \quad a = 1,\ldots,A. \tag{7}$$

where $\mathbf{p} = (x,y,z)$ is the agent's usual Euclidean spatial coordinate position, $\mathbf{e}$ is the unit vector of the spherical coordinate system, e_r is the capture radius of the red side agent, e_0 is the safe distance between agents, c_2, c_3, c_4 are constants, $\gamma_{obs,a}$, $a = 1,\ldots,A$, is the repulsive force gain coefficient between the agent i and other obstacles, $(x_{0,a}, y_{0,a}, z_{0,a})$ is the center of the corresponding obstacle, and $(\pm v_{1,a}, \pm v_{2,a}, \pm v_{3,a})$ are its vertices, which are parallel with the x, y, z axes, respectively. The interaction cost F_i^1 denotes the sum of the trajectory collision loss of the intra-group, inter-group, and with obstacles about agent i.

Remark 1. *In our 3D obstacle scene, obstacles were rectangular solids Ω_{obs}. For training, the cuboid obstacle repulsion is formulated as a differentiable unit ellipsoid repulsion function $\frac{1}{Q}$ [38]. To better reduce the collision between agents and obstacles, the radius of $\Omega_{obs,trn}$ is 10% larger than that of the unit circumscribed ellipsoid of Ω_{obs}. Our algorithm was trained by using ellipsoidal repulsion, which can produce gradient information smoothly in obstacles, stimulating the model to learn trajectories to avoid obstacles [39].*

The terminal cost G_i^1 is given by

$$G_i^1 = \underbrace{c_5\|\mathbf{x}_i^1(T) - \mathbf{x}_0\|_2}_{\text{(5) travel time minimization}}, \tag{8}$$

where c_5 is a constant, $\mathbf{x}_i^1(T)$ is the final state of agent i, and $\mathbf{x}_0$ is the target state in which we want the agents to reach the objective. The terminal cost G_i^1 denotes the distance between agent i's terminal state and the desired state.

In summary, by defining the cost function as (2), the optimal control problem faced by agent i of the blue side is given by

$$\begin{aligned} &\inf_{\mathbf{u}_i^1 \in \mathcal{U}_1} \quad J_i^1(\mathbf{x}_i^1, \mathbf{x}_i^{1^-}, \mathbf{x}_i^{2^-}, \mathbf{u}_i^1) \\ &\text{s.t.} \quad \mathrm{d}\mathbf{x}_i^1 = \mathbf{h}_i^1(\mathbf{x}_i^1, \mathbf{x}_i^{1^-}, \mathbf{x}_i^{2^-}, \mathbf{u}_i^1)\mathrm{d}t. \end{aligned} \tag{9}$$

2.2.2. Red Side Control Problem

The red side's agents traveling from a common source avoid obstacles and hope to complete the interception against the blue side to prevent the blue side from penetrating. At time $t \geq 0$, the i-th agent controls its strategy $\mathbf{u}_i^2$, to minimize its: (1) motion energy, (2) distance to the blue side, (3) red side inter-agent collision, and (4) the collision of the red side with the obstacles during the remaining travel time. The running cost L_i^2 is given by

$$L_i^2 = \underbrace{l_1 \|\mathbf{u}_i^2(t)\|^2}_{\text{(1) motion energy minimization}}, \tag{10}$$

where l_1 is a constant. The running cost L_i^2 denotes the control effort implemented by agent i. The interaction cost F_i^2 is given by

$$\begin{aligned} F_i^2 = & \underbrace{l_2\left(\frac{1}{N_1}\sum_{k=1}^{N_1}\left\|\mathbf{p}_i^2(t)-\mathbf{p}_k^1(t)\right\|\right)}_{\text{(2) distance to blue side minimization}} + \underbrace{l_3\left(\frac{1}{N_2}\sum_{j\neq i}^{N_2}\mathbf{1}_{\|\mathbf{p}_i^2(t)-\mathbf{p}_j^2(t)\|\leq e_0}\right)}_{\text{(3) red side inter-agent collision avoidance}} \\ & \underbrace{+ l_4\left(\frac{1}{N_2}\sum_{i=1}^{N_2}\begin{cases}\sum_{a=1}^{A}\gamma_{obs,a}\frac{1}{Q_a(\mathbf{p}_i,t)} & \text{if } \mathbf{p}_i \in \Omega_{obs,trn} \\ 0 & \text{otherwise}\end{cases}\right)}_{\text{(4) red side obstacle collision avoidance}}, \end{aligned} \tag{11}$$

where l_2, l_3, l_4 are constants. The interaction cost F_i^2 denotes the sum of the trajectory collision loss of the intra-group, inter-group, and with obstacles about agent i.

In summary, by defining the cost function as (2), the optimal control problem faced by agent i of the red side is given by

$$\begin{aligned} \inf_{u_i^2 \in \mathcal{U}_2} \quad & J_i^2(\mathbf{x}_i^2, \mathbf{x}_i^{2^-}, \mathbf{x}_i^{1^-}, \mathbf{u}_i^2) \\ \text{s.t.} \quad & \mathrm{d}\mathbf{x}_i^2 = \mathbf{h}_i^2(\mathbf{x}_i^2, \mathbf{x}_i^{2^-}, \mathbf{x}_i^{1^-}, \mathbf{u}_i^2)\mathrm{d}t. \end{aligned} \tag{12}$$

To this end, the cooperative attack–defense evolution for large-scale agents in the 3D obstacle scene is formulated as a non-cooperative differential game. Thus, an N-player non-cooperative game is formed. Its obvious solution is the Nash equilibrium, that is no agent can unilaterally reduce its cost under this control decision [16].

2.3. Multi-Population High-Dimensional Mean-Field Game

With the increase of the number N of game participants, the complexity of solving differential games will increase significantly. For the current agent in the cooperative attack–defense problem (16), the neighborhood interactions of intra-group collision avoidance, the inter-group interactions of attack and defense, and the ecological interactions of obstacle avoidance will lead to the need for a large amount of communication and computing resources. Traditional optimization and control methods usually deal with the increasing interactions separately, leading to the dimension explosion problem. The mean-field game is able to overcome the communication and computational difficulties associated with a large scale, and its core technology is to inscribe a large number of interacting swarm intelligence problems as coupled sets of partial differential equations (PDEs). Therefore, we propose the multi-population high-dimensional stochastic MFG (MPHD-MFG) reformulation of the cooperative attack–defense problem (16). Under the framework of the MPHD-MFG, a generic player only reacts to the collective behaviors (mean field) of all players instead of the behavior of each player, which greatly reduces the amount of communication and computation. Here, "mean-field" means the states' probability distribution. Now, we can drop the index i since players are indistinguishable within each population of the MPHD-MFG. Let $\rho^p(\mathbf{x}^p, t) : \mathbb{R}^n \times [0, T] \to \mathcal{P}_2(\mathbb{R}^n)$ denote the probability density function of state $\mathbf{x}^p$ at time t, then the cost function (2) is transformed into

$$
\begin{aligned}
&J^p(\mathbf{x}^p, \rho^p, \rho^{-p}, \mathbf{u}^p) = \\
&\mathbb{E}\Big[\int_0^T L^p(\mathbf{x}^p(t), \mathbf{u}^p(t)) + F^p(\mathbf{x}^p(t), \rho^p(\mathbf{x}^p(t), t), \rho^{-p}(\mathbf{x}^{-p}(t), t))\mathrm{d}t + G^p(\mathbf{x}^p(T), \rho^p(\mathbf{x}^p(T), T), \rho^{-p}(\mathbf{x}^{-p}(T), T))\Big] \\
&= \int_0^T \Big\{\int_\Omega L^p(\mathbf{x}^p(t), \mathbf{u}^p(t))\rho^p(\mathbf{x}^p, t)\mathrm{d}\mathbf{x}\Big\}\mathrm{d}t + \int_0^T \Big\{\int_\Omega F^p(\mathbf{x}^p(t), \rho^p(\mathbf{x}^p(t), t), \rho^{-p}(\mathbf{x}^{-p}(t), t))\rho^p(\mathbf{x}^p, t)\mathrm{d}\mathbf{x}\Big\}\mathrm{d}t \\
&\qquad + \int_\Omega G^p(\mathbf{x}^p(T), \rho^p(\mathbf{x}^p(T), T), \rho^{-p}(\mathbf{x}^{-p}(T), T))\rho^p(\mathbf{x}^p, t)\mathrm{d}\mathbf{x} \\
&= \int_0^T \Big\{\int_\Omega L^p(\mathbf{x}^p(t), \mathbf{u}^p(t))\rho^p(\mathbf{x}^p, t)\mathrm{d}\mathbf{x}\Big\}\mathrm{d}t + \int_0^T \mathcal{F}^p(\mathbf{x}^p(t), \rho^p(\mathbf{x}^p(t), t), \rho^{-p}(\mathbf{x}^{-p}(t), t))\mathrm{d}t \\
&\qquad + \mathcal{G}^p(\mathbf{x}^p(T), \rho^p(\mathbf{x}^p(T), T), \rho^{-p}(\mathbf{x}^{-p}(T), T)),
\end{aligned}
\tag{13}
$$

where $\Omega \in \mathbb{R}^n$ is the state space containing all possible states of the generic agent and variational derivatives of functionals $\mathcal{F}, \mathcal{G}$ for ρ are the interaction and terminal costs F and G, respectively. Meanwhile, the state dynamic–kinematic equation in (1) is transformed into

$$
\mathrm{d}\mathbf{x}^p = \mathbf{h}^p(\mathbf{x}^p, \rho^p, \rho^{-p}, \mathbf{u}^p)\mathrm{d}t + \sigma^p \mathrm{d}\mathbf{W}_t^p, \tag{14}
$$

where $\sigma^p \in \mathbb{R}^{n\times m}$ denotes a fixed coefficient matrix of a population-dependent volatility term and $\mathbf{W}^p$ means an m-dimensional Wiener process springs from the environment, in which each component $\mathbf{W}_k^p$ is independent of $\mathbf{W}_l^p$ for all $k \neq l$, $p = 1, 2$. According to Ito's lemma [40], (14) can be expressed in terms of the mean field $\rho^p(\mathbf{x}^p, t)$ and then will be equivalent to the Fokker–Planck (FPK) equation given by

$$
\partial_t \rho^p - \frac{{\sigma^p}^2}{2}\Delta\rho^p + \nabla \cdot (\rho^p \mathbf{h}^p) = 0. \tag{15}
$$

With cost function (13) and FPK Equation (15), the multi-population high-dimensional MFG, which describes the cooperative attack–defense evolution of a large number of agents, is now summarized as

$$
\begin{aligned}
&\inf_{\rho^p, \mathbf{u}^p} J^p(\mathbf{x}^p, \rho^p, \rho^{-p}, \mathbf{u}^p) \\
&\text{s.t. } \partial_t \rho^p - \frac{{\sigma^p}^2}{2}\Delta\rho^p + \nabla \cdot (\rho^p \mathbf{h}^p) = 0, \quad \rho^p(\mathbf{x}^p, 0) = \rho_0^p(\mathbf{x}^p) \\
&\qquad \partial_t \rho^{-p} - \frac{{\sigma^{-p}}^2}{2}\Delta\rho^{-p} + \nabla \cdot (\rho^{-p} \mathbf{h}^{-p}) = 0, \quad \rho^{-p}(\mathbf{x}^{-p}, 0) = \rho_0^{-p}(\mathbf{x}^{-p}) \\
&\qquad p = 1, 2,
\end{aligned}
\tag{16}
$$

where ρ_0^p is the initial probability distribution of population p's agents. To this end, the cooperative attack–defense evolution for large-scale agents in the 3D obstacle scene is formulated as a multi-population high-dimensional MFG. For every population $p = 1, 2$, each agent i of population $p(i)$ forecasts a distribution $\{\rho^p(\cdot, t)\}_{t=0}^T$ and aims at minimizing its cost, which eventually reaches the Nash equilibrium, where no agent can decrease its individual cost by changing its control strategy unilaterally. The formulaic representation, for every $\mathbf{x}^p \in \mathbb{R}^n$:

$$
J^p(\mathbf{x}^p, \rho^p, \rho^{-p}, \hat{\mathbf{u}^p}) \leq J^p(\mathbf{x}^p, \rho^p, \rho^{-p}, \mathbf{u}^p), \quad \forall \mathbf{u}^p : [0, T] \to \mathcal{U}_p, \tag{17}
$$

where $\hat{\mathbf{u}^p}$ is the agent's equilibrium strategy at state $\mathbf{x}^p$. Here, we assumed that the agent is small, and its unilateral actions will not change the density ρ^p. Reference [41] provides a sufficient condition for the solution to the multi-population MFG PDEs, and Reference [42] provides the necessary one.

Remark 2. *Under appropriate assumptions, the MFG's solution will offer an approximate Nash equilibrium (ϵ-NE) for the corresponding game with a large, but finite number of agents [41].*

3. GAN-Based Approach for MPHD-MFG

In this section, we put forward a generative-adversarial-network (GAN)-based method for solving the multi-population high-dimensional MFG in (16). Inspired by Wasserstein GANs [43], APAC-Net [35], and CA-Net [36,37], we formulated (16) as a convex-concave saddle-point problem using MFG's variational primal–dual structure. Then, we propose an extended coupled alternating neural network (ECA-Net) algorithm to solve the MPHD-MFG in (16).

3.1. Variational Primal–Dual Structure of MPHD-MFG

We reveal the underlying primal–dual structure of the MPHD-MFG, then deduce the convex–concave saddle-point problem equivalent to (16). Denote $\Phi_p : \Phi_p(\mathbf{x}^p, t) = \inf_{\mathbf{u}^p} J^p(\mathbf{x}^p, \rho^p, \rho^{-p}, \mathbf{u}^p)$ as the Lagrange multiplier; we can add the differential constraint (15) (FPK equation) into the cost function (13) to obtain the extended cost function:

$$\begin{aligned}\sup_{\Phi_p}\inf_{\rho^p,\mathbf{u}^p}\Big\{&\int_0^T\int_\Omega L^p(\mathbf{x}^p(t),\mathbf{u}^p(t))\rho^p(\mathbf{x}^p,t)\mathrm{d}\mathbf{x}\mathrm{d}t+\int_0^T\mathcal{F}^p(\mathbf{x}^p(t),\rho^p(\mathbf{x}^p(t),t),\rho^{-p}(\mathbf{x}^{-p}(t),t))\mathrm{d}t\\&+\mathcal{G}^p(\mathbf{x}^p(T),\rho^p(\mathbf{x}^p(T),T),\rho^{-p}(\mathbf{x}^{-p}(T),T))\\&-\int_0^T\int_\Omega\Phi_p(\mathbf{x}^p,t)(\partial_t\rho^p-\frac{\sigma^{p2}}{2}\Delta\rho^p+\nabla\cdot(\rho^p(\mathbf{x}^p,t)\mathbf{h}^p(\mathbf{x}^p,\rho^p,\rho^{-p},\mathbf{u}^p)))\mathrm{d}\mathbf{x}^p\mathrm{d}t\\&-\int_0^T\int_\Omega\Phi_{-p}(\mathbf{x}^{-p},t)(\partial_t\rho^{-p}-\frac{\sigma^{-p2}}{2}\Delta\rho^{-p}+\nabla\cdot(\rho^{-p}(\mathbf{x}^{-p},t)\mathbf{h}^{-p}(\mathbf{x}^{-p},\rho^{-p},\rho^p,\mathbf{u}^{-p})))\mathrm{d}\mathbf{x}^{-p}\mathrm{d}t\Big\}.\end{aligned}\tag{18}$$

The Hamiltonian $H_p : \mathbb{R}^n \times \mathcal{P}_2(\mathbb{R}^n)^2 \times \mathbb{R}^n \to \mathbb{R}$, $p = 1, 2$ is defined as

$$H_p(\mathbf{x}^p,\rho^p,\rho^{-p},z^p)=\sup_{\mathbf{u}^p}\{-L_p(\mathbf{x}^p,\mathbf{u}^p)-z^{p\top}\mathbf{h}^p(\mathbf{x}^p,\rho^p,\rho^{-p},\mathbf{u}^p)\}.\tag{19}$$

Utilizing (19) and integrating by parts, we can rewrite (18) as

$$\begin{aligned}\inf_{\rho^p}\sup_{\Phi_p}\Big\{&\int_0^T\int_\Omega(\partial_t\Phi_p+\frac{\sigma^{p2}}{2}\Delta\Phi_p-H_p(\mathbf{x}^p,\nabla\Phi_p))\rho^p(\mathbf{x}^p,t)\mathrm{d}\mathbf{x}^p\mathrm{d}t+\int_0^T\mathcal{F}^p(\mathbf{x}^p(t),\rho^p(\mathbf{x}^p(t),t),\rho^{-p}(\mathbf{x}^{-p}(t),t))\mathrm{d}t\\&+\mathcal{G}^p(\mathbf{x}^p(T),\rho^p(\mathbf{x}^p(T),T),\rho^{-p}(\mathbf{x}^{-p}(T),T))+\int_\Omega\Phi_p(\mathbf{x}^p,0)\rho_0^p(\mathbf{x}^p)\mathrm{d}\mathbf{x}^p-\int_\Omega\Phi_p(\mathbf{x}^p,T)\rho^p(\mathbf{x}^p,T)\mathrm{d}\mathbf{x}^p\\&+\int_0^T\int_\Omega(\partial_t\Phi_{-p}+\frac{\sigma^{-p2}}{2}\Delta\Phi_{-p}+\nabla\Phi_{-p}\cdot\mathbf{h}^{-p})\rho^{-p}(\mathbf{x}^{-p},t)\mathrm{d}\mathbf{x}^{-p}\mathrm{d}t+\int_\Omega\Phi_{-p}(\mathbf{x}^{-p},0)\rho_0^{-p}(\mathbf{x}^{-p})\mathrm{d}\mathbf{x}^{-p}\\&-\int_\Omega\Phi_{-p}(\mathbf{x}^{-p},T)\rho^{-p}(\mathbf{x}^{-p},T)\mathrm{d}\mathbf{x}^{-p}\Big\}.\end{aligned}\tag{20}$$

Here, our derivation path follows that of [44–46]. The formulation (20) is the cornerstone of our approach.

3.2. ECA-Net for Cooperative Attack–Defense Evolution

We solved (20) by training a GAN-based neural network. The solving network of the multi-population MFG in the 3D obstacle scene is an extended nonlinear coupled alternating neural network formed by multiple generators and multiple discriminators, named ECA-Net. We coupled the obstacle avoidance loss term in the design of the loss function of the ECA-Net algorithm. The structure and training process of ECA-Net are shown in Figure 2.

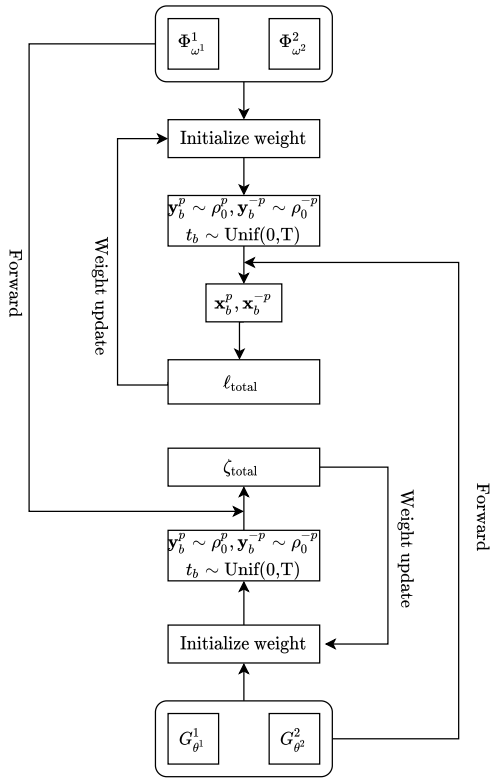


Figure 2. Visualization of the structure and training process of ECA-Net. Its training process is divided into two pairs of coupled alternating training parts—generators and discriminators.

First, we initialized two pairs of neural networks $N^p_{\omega^p}(\mathbf{x}^p, t)$ and $N^p_{\theta^p}(\mathbf{y}^p, t)$, $p = 1, 2$. Let

$$\Phi^p_{\omega^p}(\mathbf{x}^p, t) = (1-t)N^p_{\omega^p}(\mathbf{x}^p, t) + tG^p(\mathbf{x}^p), \quad \text{Ç}^p_{\theta^p}(\mathbf{y}^p, t) = (1-t)\mathbf{y}^p + tN^p_{\theta^p}(\mathbf{y}^p, t), \tag{21}$$

where $\mathbf{y}^p \sim \rho_0^p$ is a sample drawn from the initial distribution. The formulation of $\Phi^p_{\omega^p}$ (the value function for the generic agent of population p) and $\text{Ç}^p_{\theta^p}$ (the density distribution of population p) in (21) automatically encodes the terminal condition $G^p(\cdot, T)$ and initial distribution $\rho_0^p(\cdot)$, respectively. Moreover, ECA-Net encodes the underlying structure of the MPHD-MFG by (20) and (21), exempting the neural network from learning the entire game solution from scratch.

Our approach for training this neural network includes parallel–alternate training of two pairs of $\text{Ç}^p_{\theta^p}$ and $\Phi^p_{\omega^p}$, $p = 1, 2$. Intuitively, to gain the equilibrium of the MPHD-MFG of the cooperative attack–defense problem, we trained an extended coupled alternating neural network (ECA-Net) about multi-group distributions and agent controls. Specifically, we trained $\Phi^p_{\omega^p}$ by first sampling a batch $\{\mathbf{y}_b^p\}_{b=1}^B$ from given initial distribution ρ_0^p, another batch $\{\mathbf{y}_b^{-p}\}_{b=1}^B$ from given initial density ρ_0^{-p}, and $\{t_b\}_{b=1}^B$ uniformly from $[0, T]$. Then, we computed the push-forward states $\mathbf{x}_b^p = \text{Ç}^p_{\theta^p}(\mathbf{y}_b^p, t_b)$, $\mathbf{x}_b^{-p} = \text{Ç}^{-p}_{\theta^{-p}}(\mathbf{y}_b^{-p}, t_b)$ for $b = 1, \ldots, B$. The main loss term for training the discriminator $\Phi^p_{\omega^p}$ is given by

$$\text{loss}_{\Phi^p} = \frac{1}{B}\sum_{b=1}^{B}\Phi^p_{\omega^p}(\mathbf{x}_b^p, 0) + \frac{1}{B}\sum_{b=1}^{B}\left\{\partial_t\Phi^p_{\omega^p}(\mathbf{x}_b^p, t_b) + \frac{\sigma^{p2}}{2}\Delta\Phi^p_{\omega^p}(\mathbf{x}_b^p, t_b) - H_p(\nabla_{\mathbf{x}^p}\Phi^p_{\omega^p}(\mathbf{x}_b^p, t_b))\right\}. \tag{22}$$

We need to consider adding a dual term for the distribution of the neighboring population:

$$\text{penalty}_{\text{neighbors}} = \eta\left\{\frac{1}{B}\sum_{b=1}^{B}\left[\partial_t\Phi^{-p}_{\omega^{-p}}(\mathbf{x}_b^{-p}, t_b) + \frac{\sigma^{-p2}}{2}\Delta\Phi^{-p}_{\omega^{-p}}(\mathbf{x}_b^{-p}, t_b) + \nabla_{\mathbf{x}^{-p}}\Phi^{-p}_{\omega^{-p}}(\mathbf{x}_b^{-p}, t_b)\cdot\mathbf{h}^{-p}(\mathbf{x}_b^{-p}, \mathbf{x}_b^p, t_b)\right]\right\} \tag{23}$$

to correct for the density update of the other population (which tends to obey the FPK equation) [36]. We can optionally add a regularization term:

$$\text{penalty}_{\text{HJB}} = \lambda \left\{ \frac{1}{B} \sum_{b=1}^{B} \| \partial_t \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b) + \frac{\sigma^{p2}}{2} \Delta \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b) - H_p(\nabla_{\mathbf{x}^p} \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b)) + F^p(\mathbf{x}_b^p, \mathbf{x}_b^{-p}, t_b) \| \right\} \tag{24}$$

to penalize deviations from the HJB equations. Finally, we backpropagated the total loss ℓ_{total} to update the weights of the discriminator $\Phi_{\omega^p}^p$.

To train the generator, we again sampled $\{\mathbf{y}_b^p\}_{b=1}^B$, $\{\mathbf{y}_b^{-p}\}_{b=1}^B$ and $\{t_b\}_{b=1}^B$, $p = 1, 2$ as before and computed

$$\text{loss}_{\mathcal{G}^p} = \frac{1}{B} \sum_{b=1}^{B} \left\{ \partial_t \Phi_{\omega^p}^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), t_b) + \frac{\sigma^{p2}}{2} \Delta \Phi_{\omega^p}^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), t_b) - H_p(\nabla_{\mathbf{x}^p} \Phi_{\omega^p}^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), t_b)) + F^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), \mathcal{G}_{\theta^{-p}}^{-p}(\mathbf{y}_b^{-p}), t_b) \right\}. \tag{25}$$

Finally, we backpropagated the total loss ζ_{total} to update the weights of the generator $\mathcal{G}_{\theta^p}^p$.

In conclusion, in each time slot $t \in [0, T]$, the ECA-Net will be trained. The generator $\mathcal{G}_{\theta^p}^p$ will generate the state distribution of population p at time t, and the discriminator $\Phi_{\omega^p}^p$ will obtain the result of the value function of population p at time t, $p = 1, 2$. Please refer to Algorithm 1 for the detailed operation flow.

Algorithm 1 ECA-Net for cooperative attack–defense evolution.

Require: σ^p diffusion parameter, G^p terminal cost, H_p Hamiltonian, F^p interaction term, $p = 1, 2$.
Require: Initialize neural networks $N_{\omega^p}^p$ and $N_{\theta^p}^p$, batch size B.
Require: Set $\Phi_{\omega^p}^p$ and $\mathcal{G}_{\theta^p}^p$ as in (21).
While not converged, **do**

train $\Phi_{\omega^p}^p$:

Sample batch $\{(\mathbf{y}_b^p, t_b)\}_{b=1}^B$, $\{(\mathbf{y}_b^{-p}, t_b)\}_{b=1}^B$, where $\mathbf{y}_b^p \sim \rho_0^p$, $\mathbf{y}_b^{-p} \sim \rho_0^{-p}$ and $t_b \sim \text{Unif}(0, T)$.
$\mathbf{x}_b^p \leftarrow \mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p, t_b)$, $\mathbf{x}_b^{-p} \leftarrow \mathcal{G}_{\theta^{-p}}^{-p}(\mathbf{y}_b^{-p}, t_b)$ for $b = 1, \ldots, B$.
$\ell_0^p \leftarrow \frac{1}{B} \sum_{b=1}^B \Phi_{\omega^p}^p(\mathbf{x}_b^p, 0)$
$\ell_t^p \leftarrow \frac{1}{B} \sum_{b=1}^B \left\{ \partial_t \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b) + \frac{\sigma^{p2}}{2} \Delta \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b) - H_p(\nabla_{\mathbf{x}^p} \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b)) \right\}$
$\ell_{\text{n}}^p \leftarrow \eta \left\{ \frac{1}{B} \sum_{b=1}^B \left[\partial_t \Phi_{\omega^{-p}}^{-p}(\mathbf{x}_b^{-p}, t_b) + \frac{\sigma^{-p2}}{2} \Delta \Phi_{\omega^{-p}}^{-p}(\mathbf{x}_b^{-p}, t_b) + \nabla_{\mathbf{x}^{-p}} \Phi_{\omega^{-p}}^{-p}(\mathbf{x}_b^{-p}, t_b) \cdot \mathbf{h}^{-p}(\mathbf{x}_b^{-p}, \mathbf{x}_b^p, t_b) \right] \right\}$
$\ell_{\text{HJB}}^p \leftarrow \lambda \left\{ \frac{1}{B} \sum_{b=1}^B \| \partial_t \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b) + \frac{\sigma^{p2}}{2} \Delta \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b) - H_p(\nabla_{\mathbf{x}^p} \Phi_{\omega^p}^p(\mathbf{x}_b^p, t_b)) + F^p(\mathbf{x}_b^p, \mathbf{x}_b^{-p}, t_b) \| \right\}$
$\ell_{\text{total}}^p \leftarrow \ell_0^p + \ell_t^p + \ell_{\text{n}}^p + \ell_{\text{HJB}}^p$
Backpropagate total loss $\ell_{\text{total}} = \sum_p \ell_{\text{total}}^p$ to $\omega = (\omega^p)_{p=1,2}$ weights.

train $\mathcal{G}_{\theta^p}^p$:

Sample batch $\{(\mathbf{y}_b^p, t_b)\}_{b=1}^B$, $\{(\mathbf{y}_b^{-p}, t_b)\}_{b=1}^B$, where $\mathbf{y}_b^p \sim \rho_0^p$, $\mathbf{y}_b^{-p} \sim \rho_0^{-p}$ and $t_b \sim \text{Unif}(0, T)$.
$\zeta_t^p \leftarrow \frac{1}{B} \sum_{b=1}^B \left\{ \partial_t \Phi_{\omega^p}^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), t_b) + \frac{\sigma^{p2}}{2} \Delta \Phi_{\omega^p}^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), t_b) - H_p(\nabla_{\mathbf{x}^p} \Phi_{\omega^p}^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), t_b)) \right.$
$\left. + F^p(\mathcal{G}_{\theta^p}^p(\mathbf{y}_b^p), \mathcal{G}_{\theta^{-p}}^{-p}(\mathbf{y}_b^{-p}), t_b) \right\}$
Backpropagate total loss $\zeta_{\text{total}} = \sum_p \zeta_t^p$ to $\theta = (\theta^p)_{p=1,2}$ weights.
end while

4. Simulation Results

In this section, we carry out an integrative experiment based on Algorithm 1 and demonstrate the feasibility and effectiveness of our approach via the following numerical simulation results. To prove the advanced nature of our approach, we finally compare its performance with that of baseline methods.

4.1. Experimental Setup

For the attack–defense obstacle scenario in Figure 3, the initial density of the two populations is discrete uniform distributions ρ_0^p, $p = 1, 2$. The initial spatial coordinates (x, y, z) of the blue and red sides are located in the cuboid areas $((-5, 5), (-9, -7), (-9, -7))$ and $((-5, 5), (7, 9), (-9, -7))$, respectively. Note that we set all other initial coordinates to zero—initial angular position, initial velocity, and initial angular velocity were all set to zero. The coordinates of the terminal line were set to $(\cdot, 8, 8)$. Two obstacles were placed between the attacking and defending groups. The obstacles were represented by cuboids, specified by the coordinates of their vertices. The first obstacle was placed at $((-3, -1), (-2, 2), (-10, 10))$, and the second obstacle was placed at $((1, 3), (-2, 2), (-10, 10))$.

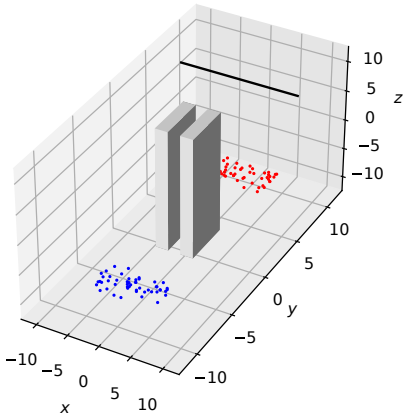


Figure 3. The 3D attack and defense experimental scenario.

We examined with this high-dimensional scene where the dynamics was that of quadrotor crafts. The dynamic–kinematic equation of the generic quadrotor craft i of population p, $p = 1, 2$, is given by

$$\begin{cases} \dot{x}_p = v_{x_p} \\ \dot{y}_p = v_{y_p} \\ \dot{z}_p = v_{z_p} \\ \dot{\psi}_p = v_{\psi_p} \\ \dot{\theta}_p = v_{\theta_p} \\ \dot{\phi}_p = v_{\phi_p} \\ \dot{v}_{x_p} = \frac{u_p}{m_p}(\sin(\phi_p)\sin(\psi_p) + \cos(\phi_p)\cos(\psi_p)\sin(\theta_p)) \\ \dot{v}_{y_p} = \frac{u_p}{m_p}(-\cos(\psi_p)\sin(\phi_p) + \cos(\phi_p)\sin(\theta_p)\sin(\psi_p)) \\ \dot{v}_{z_p} = \frac{u_p}{m_p}(\cos(\theta_p)\cos(\phi_p)) - g \\ \dot{v}_{\psi_p} = \tilde{\tau}_{\psi_p} \\ \dot{v}_{\theta_p} = \tilde{\tau}_{\theta_p} \\ \dot{v}_{\phi_p} = \tilde{\tau}_{\phi_p} \end{cases}, \tag{26}$$

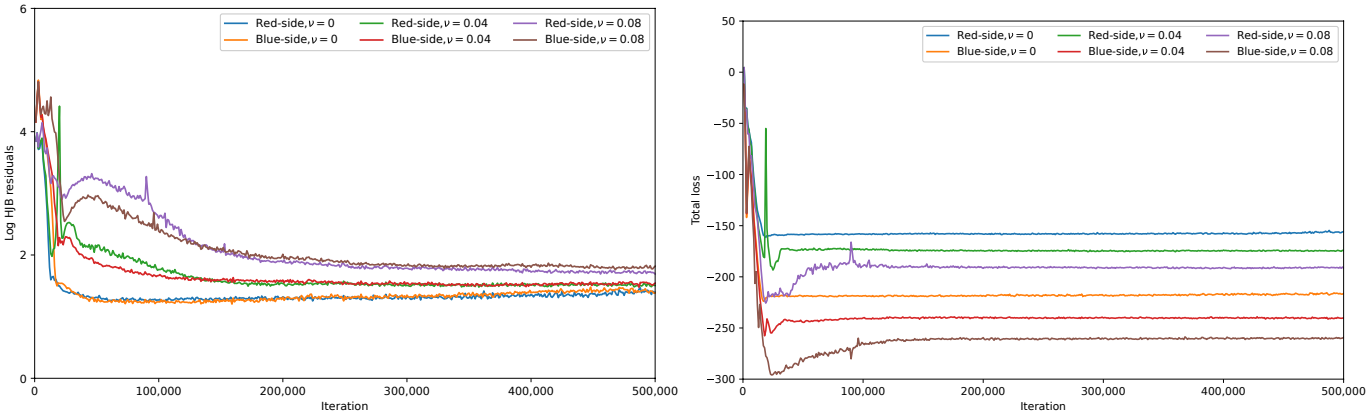
which we compactly denote as $\dot{\mathbf{x}^p} = \mathbf{h}^p(\mathbf{x}^p, \mathbf{u}^p)$, where $\mathbf{h}^p$ is a 12-dimensional vector function in the right-hand side of (26), $\mathbf{x}^p = [x_p, y_p, z_p, \psi_p, \theta_p, \phi_p, v_{x_p}, v_{y_p}, v_{z_p}, v_{\psi_p}, v_{\theta_p}, v_{\phi_p}]^\top \in \mathbb{R}^{12}$ is the state with velocities $\mathbf{v}^p = [v_{x_p}, v_{y_p}, v_{z_p}, v_{\psi_p}, v_{\theta_p}, v_{\phi_p}]^\top \in \mathbb{R}^6$, and $\mathbf{u}^p = [u_p, \tilde{\tau}_{\psi_p}, \tilde{\tau}_{\theta_p}, \tilde{\tau}_{\phi_p}]^\top \in \mathbb{R}^4$ is the control. In the stochastic case, we added a noise term to the dynamics: $\mathrm{d}\mathbf{x}^p = \mathbf{h}^p(\mathbf{x}^p, \mathbf{u}^p)\mathrm{d}t + \sigma^p \mathrm{d}\mathbf{W}_t^p$, where $\mathbf{W}$ means a standard Brownian motion, meaning the quadcopter suffers from noisy measurements. The cost functions of the blue side and red side are given in Section 2.2.

For the model hyperparameters, we set $c_1 = 0.5$ (in (3)), $c_2 = 1, c_3 = 20, c_4 = 5$ (in (4)), $c_5 = 5$ (in (8)), $l_1 = 0.5$ (in (10)), and $l_2 = 1, l_3 = 20, l_4 = 5$ (in (11)). For ECA-Net, both networks have three linear hidden layers with 100 hidden units in each layer. Residual

neural networks (ResNets) were used for both networks, with a skip connection weight of 0.5. The tanh activation function was used in $\Phi^p_{\omega^p}$, while the ReLU activation function was used in $\zeta^p_{\theta^p}$. For training, we used ADAM with $\beta = (0.5, 0.9)$, learning rate $2e-4$ for $\Phi^p_{\omega^p}$, learning rate $5e-5$ for $\zeta^p_{\theta^p}$, weight decay of $1e-4$ for both networks, batch size 50, $\lambda = 2$ (the HJB penalty coefficient), and $\eta = 2.5e-3$ (the neighbors' penalty coefficient) in Algorithm 1. As in standard machine learning methods, all the plots in Section 4.3 and Appendices A.1 and A.2 were generated using validation data (data not used in training), to ensure the general adaptability of ECA-Net.

4.2. Convergence Analysis

The convergence of the MPHD-MFG method and the ECA-Net algorithm can be observed by checking the convergence of the HJB residual errors, along with the convergence of the total loss. In Figure 4a, we plot the HJB residual errors of the red and blue sides, i.e., ℓ^p_{HJB} in Algorithm 1, which measures the deviation from the objective function (13) and shows the convergence of the theoretical model, the MPHD-MFG. Without an efficient strategy control, the HJB residuals under different stochasticity parameters ($\nu = \frac{\sigma^2}{2} = 0$, 0.04, 0.08) were relatively high. The HJB residuals dropped fast after we applied a series of controls. After around 2×10^5 iterations, the error curves tended to be bounded and stable when we obtained the optimal control for the drones. In Figure 4b, we plot the total loss of the red and blue sides, i.e., ℓ^p_{total} in Algorithm 1. After around 2×10^5 iterations, the total loss value curves under different stochasticity parameters ($\nu = \frac{\sigma^2}{2} = 0$, 0.04, 0.08) tended to be bounded and stable, which means that the weight update of the neural network was completed, proving the good convergence of the solution algorithm, ECA-Net.



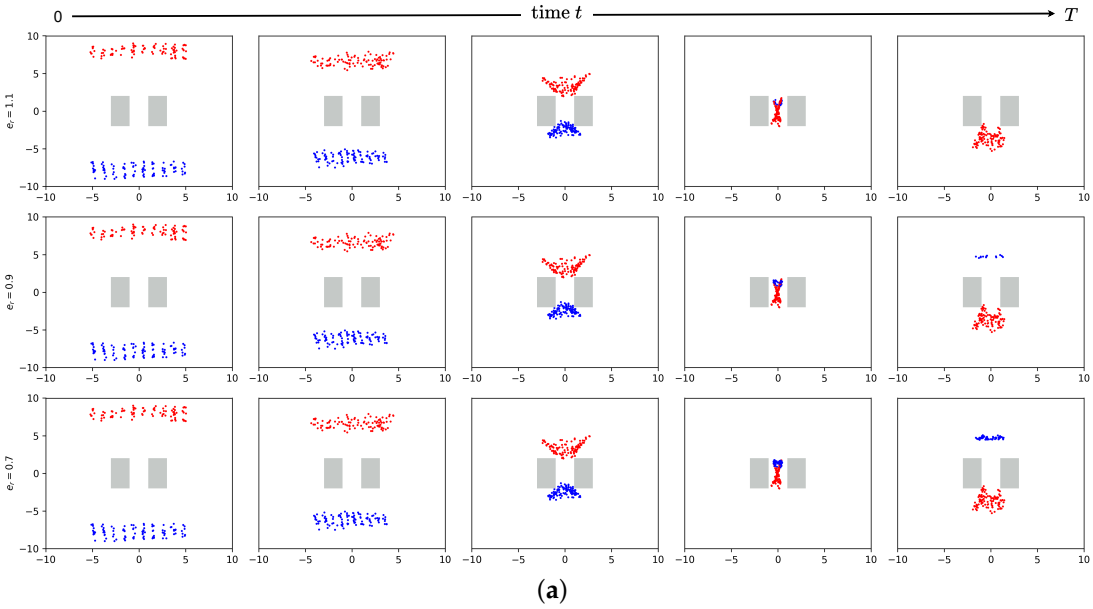
(**a**) (**b**)

Figure 4. Convergence analysis. (**a**) Convergence of HJB residual; (**b**) Convergence of total loss.

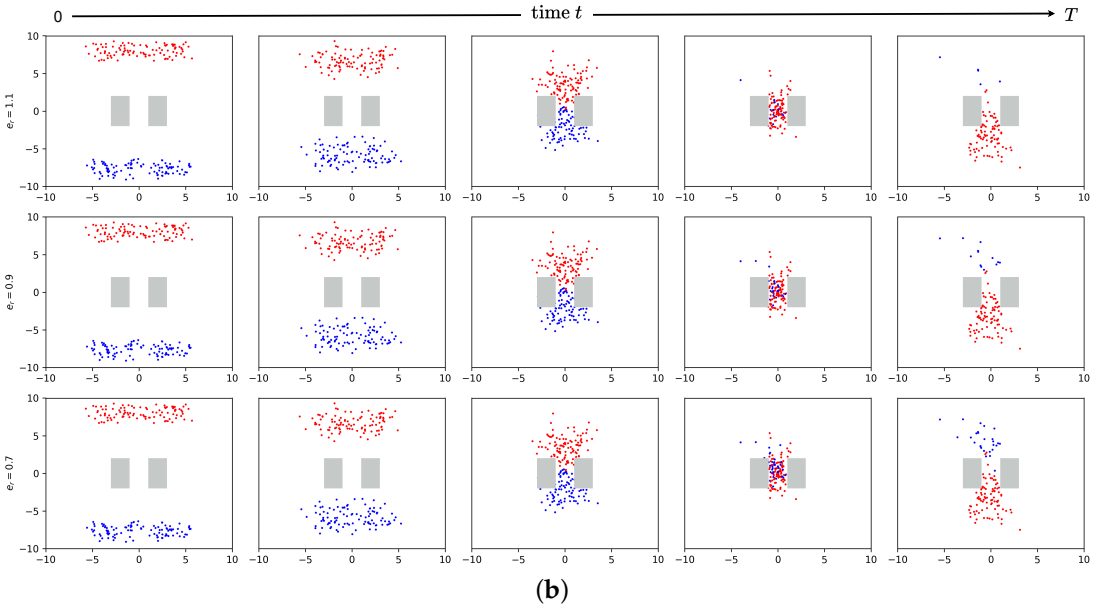
4.3. Performance Analysis

We set the same number for the blue side and red side, $N_1 = N_2 = 100$. Now, we obtained the model from the above training and predicted the trajectories of the red and blue sides. Assuming that the blue side is not aggressive and the red side is aggressive, if the blue UAV is surrounded by two or more red UAVs within e_r, the blue individual will be captured. We give the evolutionary trajectories of the red and blue sides under different stochasticity parameters ($\nu = \frac{\sigma^2}{2} = 0$, 0.04, 0.08) and different capture radii ($e_r = 0.7$, 0.9, 1.1), as shown in Figure 5. When stochasticity parameter ν was fixed, observing Figure 5a–c over time t, respectively, the UAVs successfully avoided obstacles while minimizing their energy consumption; the smaller the capture radius of the red side, the higher the survival rate of the blue side is. At the same capture radius e_r and at the same moment t, the larger the stochasticity parameter, the more scattered the distribution of the UAVs is, which increases the difficulty for the red side to completely capture the blue side. For example, in

the case of $e_r = 1.1$, comparing the first line of Figure 5a–c, the larger ν is, the higher the survival rate of the blue side. In particular, Figure 5c shows the diversion phenomenon. At $\nu = 0.08$, the third moment, the UAVs are flying through the obstacles. In order to avoid collision with the obstacles, the UAVs choose three different paths, i.e., a diversion occurs, which demonstrates the adaptive nature of the UAV. In addition, the corresponding 3D run diagram of Figure 5 is placed in Appendix A.1. Appendix A.2 shows and analyzes the offensive and defensive effects of the UAV swarms in the asymmetric case.

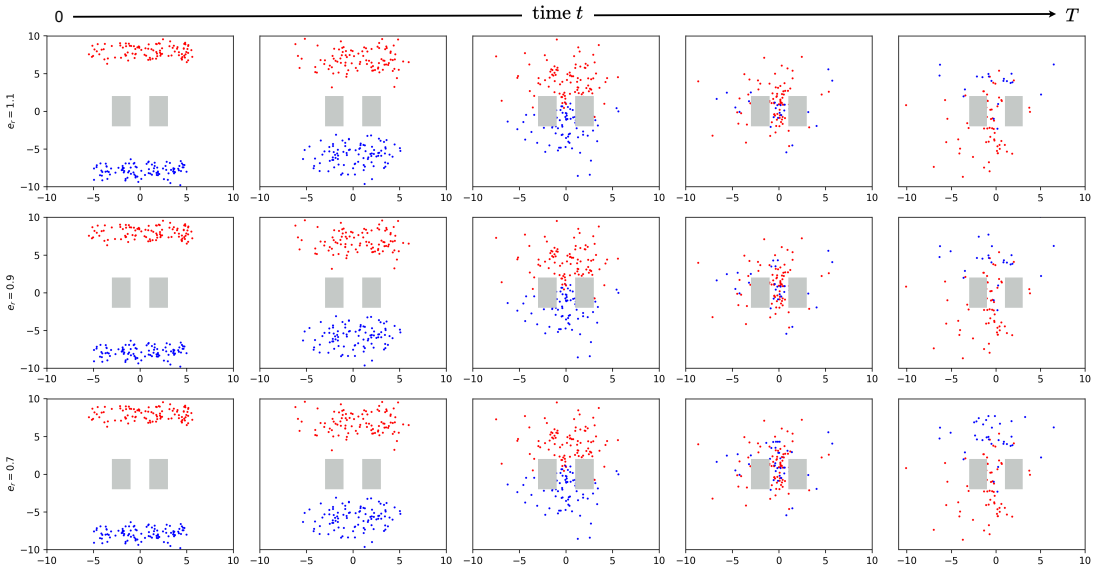


(**a**)



(**b**)

Figure 5. *Cont.*



(**c**)

Figure 5. Vertical view of the large-scale UAV attack and defense behaviors under different stochasticity parameters and different capture radii. (**a**) Snapshots of the distributions of the UAVs' locations under different capture radii when $\nu = 0$; (**b**) Snapshots of the distributions of the UAVs' locations under different capture radii when $\nu = 0.04$; (**c**) Snapshots of the distributions of the UAVs' locations under different capture radii when $\nu = 0.08$.

4.4. Comparison with Baselines

We verified the progressiveness of our approach by comparing its performance with that of some typical baseline methods for attack–defense games in Table 1. From Table 1, it can be seen that our approach handled the most complex application scene and simplified the large-scale communication. For more details about the following baseline methods, please refer to the related References [7–9,37], etc.

Table 1. Comparison with baseline methods for attack–defense games.

Method	Scene	Scale of UAVs	Scene Complexity	Communication
[7]	3D blank scene	Small	0.67 [1]	$\mathcal{O}(N)$ [2]
[8]	2D obstacle scene	Small	0.67	$\mathcal{O}(N)$
[9]	2D blank scene	Large	0.67	$\mathcal{O}(N)$
[37]	3D blank scene	Large	0.83	$\mathcal{O}(1)$
Ours	3D obstacle scene	Large	1	$\mathcal{O}(1)$

[1] The measurement method of scene complexity is as follows: here, it is a scoring system, [2D, 3D] = [$1^{'}$, $2^{'}$]; [blank scene, obstacle scene] = [$1^{'}$, $2^{'}$]; [small, large] = [$1^{'}$, $2^{'}$]. We accumulated the scores for each literature experiment scene according to each item and finally normalized them. [2] $\mathcal{O}()$ is infinitesimal of the same order.

5. Conclusions

In this paper, we formulated the cooperative attack–defense evolution of large-scale agents in high-dimensional environments as a multi-population high-dimensional stochastic mean-field game (MPHD-MFG), which significantly reduced the communication frequency and computational complexity of the swarm intelligence system. Then, we tractably solved the MPHD-MFG with a generative-adversarial-network (GAN)-based method using the MFGs' underlying variational primal–dual structure. Based on our approach, we conducted a comprehensive experiment. The good convergence of the MPHD-MFG method and the ECA-Net algorithm was corroborated by checking the bounded stable convergence of the HJB residual error and the total loss. Through simulations, we saw that a large number of UAVs can avoid obstacles (even showing diversions) and smoothly evolve their

attack and defense behaviors while minimizing their energy consumption.The comparison with the baseline methods showed that our approach is advanced. In the future, we will consider in-depth research in the asymmetric case of 3D obstacle scenarios, for example the evolution of a cooperative attack and defense between multiple (greater than or equal to three) large-scale swarms and the evolution of a cooperative attack and defense under the existence of individual performance (speed, acceleration, turning range) differences between attackers and defenders.

Author Contributions: Conceptualization, G.W. and Z.L.; Formal analysis, G.W.; Investigation, Z.L.; Methodology, G.W. and W.Y.; Software, G.W.; Supervision, W.Y. and S.X.; Validation, W.Y. and S.X.; Visualization, G.W.; Writing—original draft, G.W. and Z.L.; Writing—review and editing, W.Y. and S.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Science and Technology Innovation 2030-Key Project under Grant 2020AAA0108200.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We thank Zhiming Zheng for his fruitful discussions, valuable opinions, and guidance.

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

Abbreviations

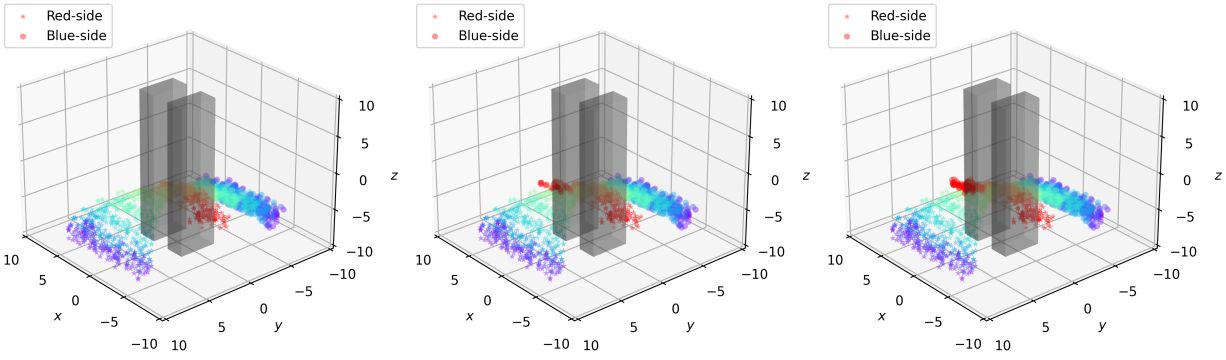
The following abbreviations are used in this manuscript:

MFG	Mean-field game
3D	Three-dimensional
UAV	Unmanned aerial vehicle
GANs	Generative adversarial neural networks
ECA-Net	Extended coupled alternating neural network
HJB	Hamilton–Jacobi–Bellman (partial differential equation)
FPK	Fokker–Planck (equation)

Appendix A. The 3D Renderings of Numerical Results and More Experiments

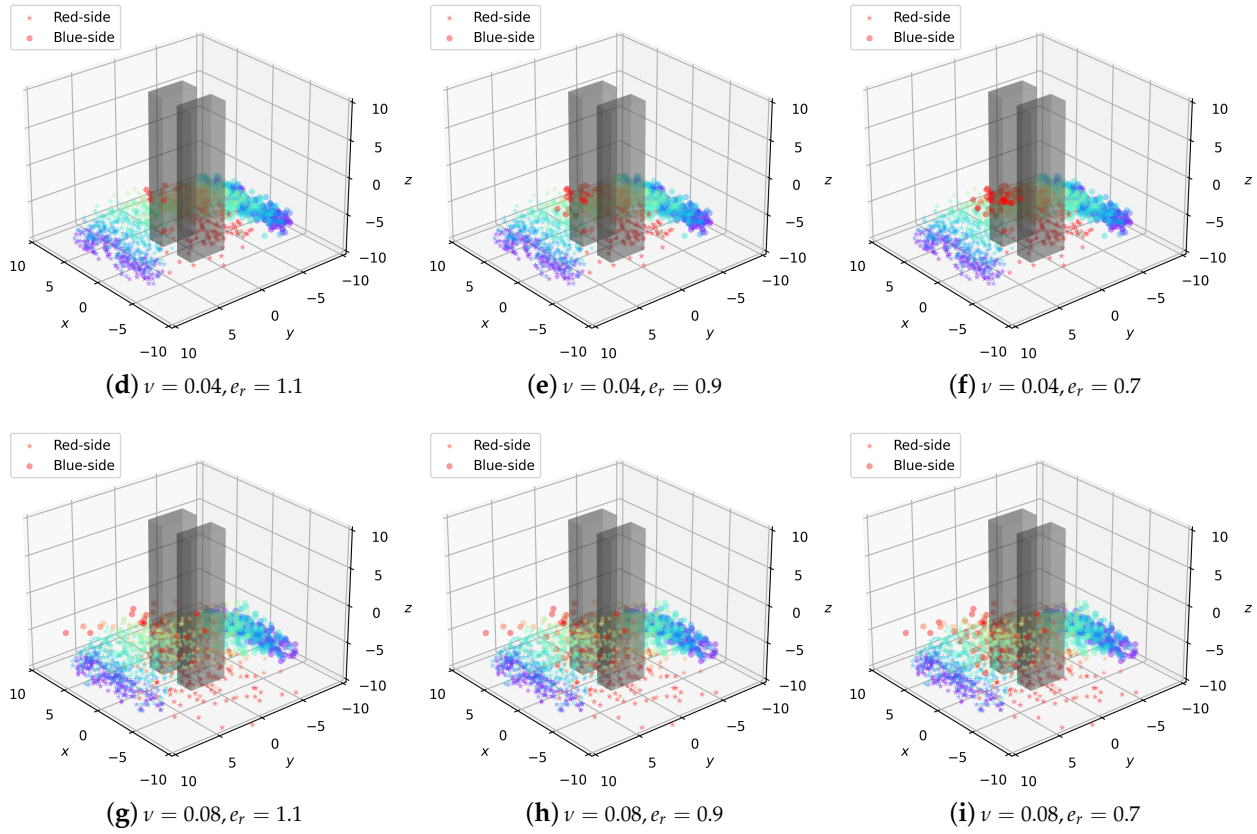
Appendix A.1. The 3D Run Diagram Figure A1 about Figure 5

Here, we give the 3D experimental run diagram Figure A1 about its vertical view Figure 5 for reference. In the following diagrams, time is represented by color. Specifically, purple represents the starting time, red represents the final time, and the intermediate colors represent intermediate times.



(a) $\nu = 0, e_r = 1.1$ **(b)** $\nu = 0, e_r = 0.9$ **(c)** $\nu = 0, e_r = 0.7$

Figure A1. *Cont.*



(d) $\nu = 0.04, e_r = 1.1$ **(e)** $\nu = 0.04, e_r = 0.9$ **(f)** $\nu = 0.04, e_r = 0.7$

(g) $\nu = 0.08, e_r = 1.1$ **(h)** $\nu = 0.08, e_r = 0.9$ **(i)** $\nu = 0.08, e_r = 0.7$

Figure A1. The 3D run diagram of large-scale UAV attack and defense behaviors under different stochasticity parameters and different capture radii.

Appendix A.2. Asymmetric Case Study

Here, we set different numbers of the blue side and red side, $N_1 = 100, N_2 = 60$ or $N_1 = 60, N_2 = 100$. Let $\nu = 0.08, e_r = 1.1$ and the capture conditions be consistent with Section 4.3. Now, we can predict its trajectory, as shown in Figures A2 and A3. Taking the given parameters as an example, with the fixed stochasticity parameter and capture radius, this section shows the deduction of the diversion obstacle avoidance and attack–defense behaviors of the red and blue sides with asymmetric numbers.

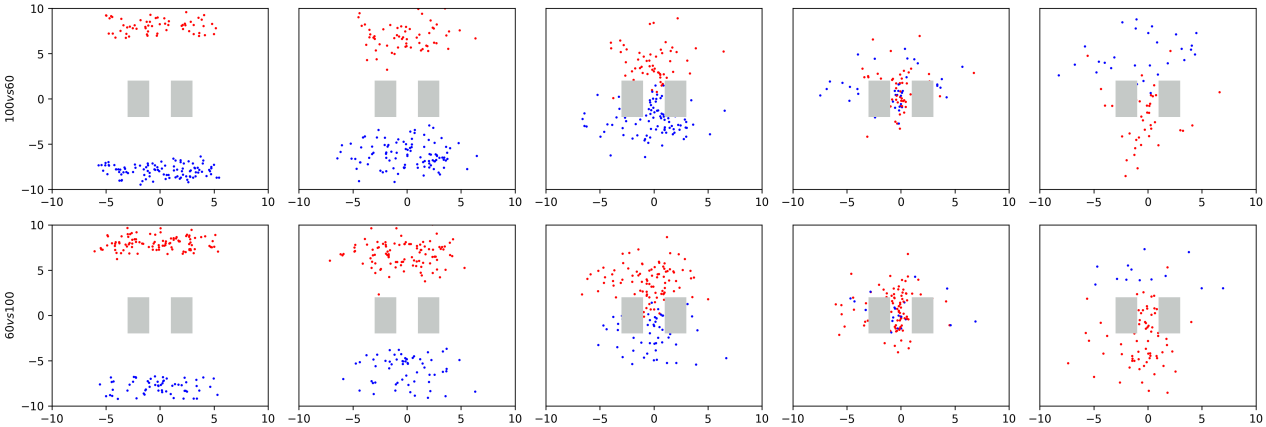


Figure A2. Vertical view of the asymmetric case about large-scale UAV attack and defense behaviors under $\nu = 0.08, e_r = 1.1$.

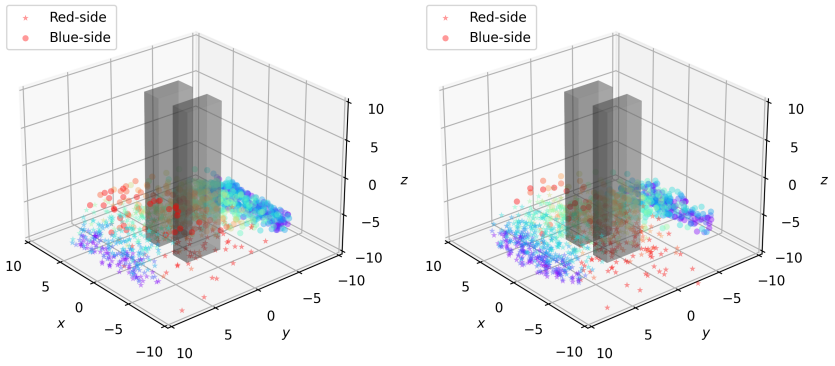


Figure A3. The 3D run diagram of the asymmetric case about large-scale UAV attack and defense behaviors under $\nu = 0.08, e_r = 1.1$. (**a**) Blue side vs. red side: 100 vs. 60; (**b**) Blue side vs. red side: 60 vs. 100.

References

1. Yu, C.; Zhang, M.; Ren, F.; Tan, G. Multiagent Learning of Coordination in Loosely Coupled Multiagent Systems. *IEEE Trans. Cybern.* **2015**, *45*, 2853–2867. [CrossRef]
2. Yang, Y.; Luo, R.; Li, M.; Zhou, M.; Zhang, W.; Wang, J. Mean Field Multi-Agent Reinforcement Learning. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; Volume 80, pp. 5571–5580.
3. Chen, C.; Mo, L.; Zheng, D. Cooperative attack–defense game of multiple UAVs with asymmetric maneuverability. *Acta Aeronaut. Astronaut. Sin.* **2020**, *41*, 324152. [CrossRef]
4. Huang, L.; Fu, M.; Qu, H.; Wang, S.; Hu, S. A deep reinforcement learning-based method applied for solving multi-agent defense and attack problems. *Expert Syst. Appl.* **2021**, *176*, 114896. [CrossRef]
5. Huang, H.; Zhang, W.; Ding, J.; Stipanovic, D.M.; Tomlin, C.J. Guaranteed decentralized pursuit-evasion in the plane with multiple pursuers. In Proceedings of the IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, 12–15 December 2011. [CrossRef]
6. Zha, W.; Chen, J.; Peng, Z.; Gu, D. Construction of Barrier in a Fishing Game With Point Capture. *IEEE Trans. Cybern.* **2017**, *47*, 1409–1422. [CrossRef] [PubMed]
7. Liu, Y.; Qi, N.; Tang, Z. Linear Quadratic Differential Game Strategies with Two-pursuit Versus Single-evader. *Chin. J. Aeronaut.* **2012**, *25*, 896–905. [CrossRef]
8. Wan, K.; Wu, D.; Zhai, Y.; Li, B.; Gao, X.; Hu, Z. An Improved Approach towards Multi-Agent Pursuit–Evasion Game Decision-Making Using Deep Reinforcement Learning. *Entropy* **2021**, *23*, 1433. [CrossRef]
9. Garcia, E.; Casbeer, D.W.; Moll, A.V.; Pachter, M. Multiple Pursuer Multiple Evader Differential Games. *IEEE Trans. Autom. Control* **2021**, *66*, 2345–2350. [CrossRef]
10. Earl, M.; D'Andrea, R. Modeling and control of a multi-agent system using mixed integer linear programming. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002. [CrossRef]
11. Earl, M.; D'Andrea, R. A study in cooperative control: The RoboFlag drill. In Proceedings of the Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301), Anchorage, AK, USA, 8–10 May 2002. [CrossRef]
12. Bakolas, E.; Tsiotras, P. Optimal pursuit of moving targets using dynamic Voronoi diagrams. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010. [CrossRef]
13. Isaacs, R. *Differential Games*; Wiley: Hoboken, NJ, USA, 1967.
14. Lasry, J.M.; Lions, P.L. Jeux à champ moyen. I–Le cas stationnaire. *Comptes Rendus Math.* **2006**, *343*, 619–625. [CrossRef]
15. Lasry, J.M.; Lions, P.L. Jeux à champ moyen. II–Horizon fini et contrôle optimal. *Comptes Rendus Math.* **2006**, *343*, 679–684. [CrossRef]
16. Lasry, J.M.; Lions, P.L. Mean field games. *Jpn. J. Math.* **2007**, *2*, 229–260. [CrossRef]
17. Huang, M.; Caines, P.; Malhame, R. Individual and mass behaviour in large population stochastic wireless power control problems: Centralized and nash equilibrium solutions. In Proceedings of the 42nd IEEE International Conference on Decision and Control (IEEE Cat. No.03CH37475), Maui, HI, USA, 9–12 December 2003. [CrossRef]
18. Caines, P.E.; Huang, M.; Malhamé, R.P. Large population stochastic dynamic games: Closed-loop McKean-Vlasov systems and the Nash certainty equivalence principle. *Commun. Inf. Syst.* **2006**, *6*, 221–252. [CrossRef]
19. Huang, M.; Caines, P.E.; Malhame, R.P. Large-Population Cost-Coupled LQG Problems With Nonuniform Agents: Individual-Mass Behavior and Decentralized ε-Nash Equilibria. *IEEE Trans. Autom. Control* **2007**, *52*, 1560–1571. [CrossRef]
20. Gomes, D.; Saúde, J. A mean-field game approach to price formation in electricity markets. *arXiv* **2018**, arXiv:1807.07088.
21. Kizilkale, A.C.; Salhab, R.; Malhamé, R.P. An integral control formulation of mean field game based large scale coordination of loads in smart grids. *Automatica* **2019**, *100*, 312–322. [CrossRef]

22. Paola, A.D.; Trovato, V.; Angeli, D.; Strbac, G. A Mean Field Game Approach for Distributed Control of Thermostatic Loads Acting in Simultaneous Energy-Frequency Response Markets. *IEEE Trans. Smart Grid* **2019**, *10*, 5987–5999. [CrossRef]
23. Lachapelle, A.; Wolfram, M.T. On a mean field game approach modeling congestion and aversion in pedestrian crowds. *Transp. Res. Part B Methodol.* **2011**, *45*, 1572–1589. [CrossRef]
24. Burger, M.; Francesco, M.D.; Markowich, P.A.; Wolfram, M.T. Mean field games with nonlinear mobilities in pedestrian dynamics. *Discret. Contin. Dyn. Syst.-B* **2014**, *19*, 1311–1333. [CrossRef]
25. Aurell, A.; Djehiche, B. Mean-Field Type Modeling of Nonlocal Crowd Aversion in Pedestrian Crowd Dynamics. *SIAM J. Control Optim.* **2018**, *56*, 434–455. [CrossRef]
26. Achdou, Y.; Lasry, J.M. Mean Field Games for Modeling Crowd Motion. In *Computational Methods in Applied Sciences*; Springer International Publishing: Cham, Switzerland, 2018; pp. 17–42. [CrossRef]
27. Liu, Z.; Wu, B.; Lin, H. A Mean Field Game Approach to Swarming Robots Control. In Proceedings of the IEEE 2018 Annual American Control Conference (ACC), Milwaukee, WI, USA, 27–29 June 2018. [CrossRef]
28. Elamvazhuthi, K.; Berman, S. Mean-field models in swarm robotics: A survey. *Bioinspir. Biomimet.* **2019**, *15*, 015001. [CrossRef]
29. Lee, W.; Liu, S.; Tembine, H.; Li, W.; Osher, S. Controlling Propagation of Epidemics via Mean-Field Control. *SIAM J. Appl. Math.* **2021**, *81*, 190–207. [CrossRef]
30. Chang, S.L.; Piraveenan, M.; Pattison, P.; Prokopenko, M. Game theoretic modelling of infectious disease dynamics and intervention methods: A review. *J. Biol. Dyn.* **2020**, *14*, 57–89. [CrossRef] [PubMed]
31. E, W.; Han, J.; Li, Q. A mean-field optimal control formulation of deep learning. *Res. Math. Sci.* **2018**, *6*, 10. [CrossRef]
32. Guo, X.; Hu, A.; Xu, R.; Zhang, J. Learning Mean-Field Games. In *Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
33. Carmona, R.; Laurière, M.; Tan, Z. Linear-Quadratic Mean-Field Reinforcement Learning: Convergence of Policy Gradient Methods. *arXiv* **2019**, arXiv:1910.04295.
34. Guéant, O.; Lasry, J.M.; Lions, P.L. Mean Field Games and Applications. In *Paris-Princeton Lectures on Mathematical Finance 2010*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 205–266. [CrossRef]
35. Lin, A.T.; Fung, S.W.; Li, W.; Nurbekyan, L.; Osher, S.J. Alternating the population and control neural networks to solve high-dimensional stochastic mean-field games. *Proc. Natl. Acad. Sci. USA* **2021**, *118*, e2024713118. [CrossRef] [PubMed]
36. Wang, G.; Yao, W.; Zhang, X.; Niu, Z. Coupled Alternating Neural Networks for Solving Multi-Population High-Dimensional Mean-Field Games with Stochasticity. *TechRxiv Preprint* **2022**. [CrossRef]
37. Wang, G.; Zhang, X.; Yao, W.; Ren, L. Cooperative attack–defense evolution of large-scale agents. In Proceedings of the ACM Genetic and Evolutionary Computation Conference Companion, Boston, MA, USA, 9–13 July 2022. [CrossRef]
38. Chang, K.; Xia, Y.; Huang, K. UAV formation control design with obstacle avoidance in dynamic three-dimensional environment. *SpringerPlus* **2016**, *5*, 1124. [CrossRef] [PubMed]
39. Onken, D.; Nurbekyan, L.; Li, X.; Fung, S.W.; Osher, S.; Ruthotto, L. A Neural Network Approach for High-Dimensional Optimal Control Applied to Multiagent Path Finding. *IEEE Trans. Control. Syst. Technol.* **2022**, 1–17. [CrossRef]
40. Schulte, J.M. Adjoint Methods for Hamilton–Jacobi–Bellman Equations. Ph.D. Thesis, University of Munster, Münster, Germany, 2010.
41. Fujii, M. Probabilistic Approach to Mean Field Games and Mean Field Type Control Problems with Multiple Populations. *SSRN Electron. J.* **2019.** [CrossRef]
42. Bensoussan, A.; Huang, T.; Laurière, M. Mean Field Control and Mean Field Game Models with Several Populations. *arXiv* **2018**, arXiv:1810.00783.
43. Arjovsky, M.; Chintala, S.; Bottou, L. Wasserstein Generative Adversarial Networks. In Proceedings of the 34th International Conference on Machine Learning—Volume 70 (ICML'17), Sydney, Australia, 6–11 August 2017; pp. 214–223.
44. Benamou, J.D.; Carlier, G.; Santambrogio, F. Variational Mean Field Games. In *Active Particles*; Modeling and Simulation in Science, Engineering & Technology; Springer International Publishing: Cham, Switzerland, 2017; Volume 1, pp. 141–171.
45. Cardaliaguet, P.; Graber, P.J. Mean field games systems of first order. *ESAIM Control. Optim. Calc. Var.* **2015**, *21*, 690–722. [CrossRef]
46. Cardaliaguet, P.; Graber, P.J.; Porretta, A.; Tonon, D. Second order mean field games with degenerate diffusion and local coupling. *Nonlinear Differ. Equ. Appl. NoDEA* **2015**, *22*, 1287–1317. [CrossRef]