remote sensing

MDPI

Article

Energy-Efficient Multi-UAVs Cooperative Trajectory Optimization for Communication Coverage: An MADRL Approach

Tianyong Ao [1,2], Kaixin Zhang [1,2], Huaguang Shi [1,2,*], Zhanqi Jin [1,2], Yi Zhou [1,2] and Fuqiang Liu [3]

[1] School of Artificial Intelligence, Henan University, Zhengzhou 450046, China; tyao@vip.henu.edu.cn (T.A.); zhangkaixin@henu.edu.cn (K.Z.); jinzhanqi@henu.edu.cn (Z.J.); zhouyi@henu.edu.cn (Y.Z.)
[2] International Joint Research Laboratory for Cooperative Vehicular Networks of Henan, Zhengzhou 450046, China
[3] College of Electronic and Information Engineering, Tongji University, Shanghai 201804, China; liufuqiang@tongji.edu.cn
* Correspondence: shihuaguang@henu.edu.cn

Abstract: Unmanned Aerial Vehicles (UAVs) can be deployed as aerial wireless base stations which dynamically cover the wireless communication networks for Ground Users (GUs). The most challenging problem is how to control multi-UAVs to achieve on-demand coverage of wireless communication networks while maintaining connectivity among them. In this paper, the cooperative trajectory optimization of UAVs is studied to maximize the communication efficiency in the dynamic deployment of UAVs for emergency communication scenarios. We transform the problem into a Markov game problem and propose a distributed trajectory optimization algorithm, Double-Stream Attention multi-agent Actor-Critic (DSAAC), based on Multi-Agent Deep Reinforcement Learning (MADRL). The throughput, safety distance, and power consumption of UAVs are comprehensively taken into account for designing a practical reward function. For complex emergency communication scenarios, we design a double data stream network structure that provides a capacity for the Actor network to process state changes. Thus, UAVs can sense the movement trends of the GUs as well as other UAVs. To establish effective cooperation strategies for UAVs, we develop a hierarchical multi-head attention encoder in the Critic network. This encoder can reduce the redundant information through the attention mechanism, which resolves the problem of the curse of dimensionality as the number of both UAVs and GUs increases. We construct a simulation environment for emergency networks with multi-UAVs and compare the effects of the different numbers of GUs and UAVs on algorithms. The DSAAC algorithm improves communication efficiency by 56.7%, throughput by 71.2%, energy saving by 19.8%, and reduces the number of crashes by 57.7%.

Keywords: Multi-Agent Deep Reinforcement Learning (MADRL); multi-UAVs trajectory optimization; emergency communication; attention mechanism

Citation: Ao, T.; Zhang, K.; Shi, H.; Jin, Z.; Zhou, Y.; Liu, F. Energy-Efficient Multi-UAVs Cooperative Trajectory Optimization for Communication Coverage: An MADRL Approach. *Remote Sens.* **2023**, *15*, 429. https://doi.org/10.3390/rs15020429

Academic Editor: Joaquín Martínez-Sánchez

Received: 9 November 2022
Revised: 5 January 2023
Accepted: 9 January 2023
Published: 11 January 2023

1. Introduction

In modern society, UAVs have become indispensable tools and are deployed in many complex environments to complete various tasks [1]. In the scenes of natural disasters, emergencies, wars, etc., the original communication facilities are damaged, UAVs can provide emergency communication [2]. UAVs are employed in communication networks with the advantages of flexibility, rapid deployment, and dynamic distribution on-demand [3]. However, there are many challenges due to the dynamic environment, limited power, limited distance, and other factors. Thus, how to establish a reasonable aerial base station through UAVs is receiving increasing attention from scholars [4–6].

Due to the limitations of bandwidth, coverage, and the number of UAVs, aerial base stations need to be dynamically deployed to fill the signal coverage gap in time and allocate

the network resources to Ground Users (GUs) on-demand when ground base stations fail [7,8]. Establishing a dynamic relay network with multi-UAVs can significantly improve the coverage radius and robustness of the networks [9]. However, a well-designed collaborative policy among UAVs is required to autonomously assign tasks and cooperatively optimize the flight trajectory [10,11]. In summary, UAVs mainly face two challenges in a complex deployment wireless relay network scenario: (1) How to maximize the communication efficiency (e.g., throughput, energy saving) of UAVs. (2) How to achieve cooperative trajectory optimization of UAVs.

For the above communication efficiency optimization problem, most studies regard the UAVs energy consumption and throughput as the primary optimization objective. In [12], a safe-deep-Q-network is proposed to maximize throughput and UAV energy efficiency. This work deployed an emergency communication network through a single UAV, which has less redundancy and robustness compared to a multi-UAV network system. Saxena et al. [13] proposed a DRL algorithm based on the Flow-Level Model to optimize the flight trajectory of UAVs for maximizing the throughput of GUs. In the above work, multi-UAVs do not establish a complex cooperative relationship and cannot continuously track services. Samir et al. [14] proposed a UAV trajectory optimization algorithm based on DRL to maximize the efficiency of highway vehicle network coverage. This work does not use the advantage of multi-UAV to achieve multi-hop relaying communication, which greatly increases the complexity of the system. In the above algorithms [12–14], UAVs only perform actions based on the currently obtained state information without sensing the movement trend of the GUs and other UAVs, which degrades the communication efficiency of UAVs.

For cooperative trajectory optimization of UAVs, the previous works focus on establishing effective cooperation strategies for UAVs. Wu et al. [15] proposed a federated multi-agent deep deterministic policy gradient based trajectory optimization algorithm for maximizing the average spectrum efficiency. This can solve the problem of environmental non-stationarity of multi-agent. However, the sequential training approach can cause the problem of unsynchronized training strategies among agents. Liu et al. [16] extended single-UAV Q-learning to multi-UAVs by training only single UAV at a time with the strategies of the other UAVs fixed. However, the number of UAVs deployed is low with only six. With the increasing number of UAVs and the complexity of the network topology, UAVs are required to consider different levels of association when cooperatively optimizing the flight trajectory [17]. In addition, the increasing number of UAVs and GUs will lead to the curse of dimensionality.

Motivated by the above discussion, this paper studies communication efficiency maximization and cooperative trajectory optimization of UAVs. The main contributions of this paper are summarized:

- To solve the non-convex optimization problem of deploying multi-UAVs relay emergency networks, we transform the multi-UAVs cooperative trajectory optimization problem into a Markov game problem. Additionally, we propose a distributed trajectory optimization algorithm DSAAC based on MADRL and an attention mechanism to maximize the communication efficiency of UAVs.
- To improve the dynamic performance of UAVs in unknown scenarios, we propose a double data stream structure in the Actor network of UAVs. This structure can process the differential state of UAVs through symmetric network branches. Thus, the dynamic information perception ability of the UAVs is enhanced, which can enable effective obstacle avoidance.
- We design a hierarchical multi-headed attention encoder to enable UAVs to establish an effective cooperation strategy. This encoder reduces information interference from irrelevant UAVs through the attention mechanism and effectively solves the problem of curse of dimensionality.

2. Related Work

Recently, UAVs as aerial base stations to provide wireless network coverage for GUs or equipment have attracted widespread attention from industry and academia [18]. In particular, one of the important challenges is maximizing the communication efficiency of UAVs. The related studies for this challenge can be divided into two categories. The first is a traditional convex optimization or control methods, and the second is the Deep Reinforcement Learning (DRL) methods.

The methods of traditional convex optimization or control methods can be used to maximize the communication efficiency of UAVs. Wu et al. [19] proposed an efficient iterative algorithm based on block coordinate descent and successive convex optimization techniques. The communication schedule, trajectory, and communication power of the UAV are alternately optimized to maximize the throughput of UAV and minimize energy consumption. This algorithm has a low complexity and a fast convergence rate, while the power consumption of the UAV is not taken into consideration. Sun et al. [20] also used a similar approach to optimize the power allocation and trajectory of UAV relay nodes to maximize task energy efficiency. This algorithm can achieve higher gains in convergence and resource allocation. However, this algorithm is only applicable to a single UAV. Huang et al. [21] studied the problem of minimizing UAV task time to optimize UAV communication efficiency. The UAV trajectory optimization algorithms are proposed based on optimal control theory. The above papers [19–21] adopt traditional methods to optimize the trajectory of the UAV. However, traditional trajectory optimization algorithms are applicable to simple scenarios and single-objective optimization problems.

DRL can effectively address complex non-convex optimization problems by continuously interacting with the environment to guide the agent to obtain maximum rewards. In the application of emergency network deployment by UAVs, DRL has been applied in the initial stages [22]. Wang et al. [23] studied the UAV navigation problem in large-scale complex environments and proposed a path planning algorithm for continuous action space based on online DRL. However, the authors [23] did not consider the energy consumption of the UAV. Liu et al. [16] proposed a localization algorithm based on the multi-agent Q-learning to optimize the initial location of the UAVs as well as the flight trajectory based on the actual location, network bandwidth requirement, and power consumption of users. Ding et al. [24] developed a UAV power consumption model and proposed a trajectory optimization based on DRL for optimizing the energy efficiency of UAVs. The above studies [16,23,24] did not consider the crash factor of the UAVs. In complex and changing scenarios, UAVs need to have the ability to avoid obstacles autonomously. Zhang et al. [12] proposed a UAV trajectory optimization algorithm based on the safe-DQN considering the complex environmental factors after a disaster. This algorithm maximizes the throughput and communication energy efficiency of the UAV without crashing. In the above studies [12,16,22–24], the UAVs only make corresponding actions based on the state of the current time slot, which cannot respond to the change of state. Therefore, these algorithms are not applicable to dynamic scenarios.

The deployment of multi-UAVs can achieve a larger coverage network coverage area. However, multi-UAVs cooperative optimization increases the system complexity and solution difficulty, which is an important research direction in the field of UAVs. Wang et al. [25] studied multi-UAVs to provide on-demand network coverage for ground devices. The distributed motion algorithm is proposed based on the centralized greedy search algorithm to minimize the number of UAVs and maximize the load balance among UAVs. Similarly, Wang et al. [26] used a particle swarm optimization algorithm to optimize the deployment location of multi-UAVs. The above studies [25,26] have two common features: (1) They only optimize the deployment location of the UAVs without applying dynamic GUs; (2) They are centralized optimization algorithms that will affect the functionality of the system when the UAV fails. Shi et al. [27] addressed the above problems using a distributed Deep Q-network (DQN), where each UAV is provided with a DQN and shares action decisions. This algorithm maximizes throughput while considering fair service

constraints. Cui et al. [28] proposed a similar approach to optimize the resource allocation problem for multi-UAVs communication networks. In the above studies [27,28], UAVs are required to maintain continuous communication.

However, there is electromagnetic interference in the emergency communication scenario that makes the UAV temporarily disconnected. Thus, UAVs need to be applied distributed trajectory optimization algorithms. Yin et al. [29] proposed a distributed training algorithm based on the QMIX algorithm for maximizing fair throughput by optimizing the trajectory of the UAVs. Ding et al. [11] proposed a multi-UAVs cooperative trajectory optimization algorithm based on a multi-agent deep deterministic policy gradient algorithm for maximizing fair throughput. The algorithm uses the Centralized Training Decentralized Execution (CTDE) architecture, which means that UAVs can cooperate without observing the actions of other UAVs after completing training. Xia et al. [30] proposed a multi-UAV soft Actor-Critic algorithm to optimize UAV tracking trajectories based on the CTDE architecture. Liu et al. [7] proposed a distributed multi-UAVs cooperative control algorithm based on DRL to achieve fair coverage and minimize task energy consumption. In the above studies [7,27–30], the MADRL approaches will suffer from the curse of dimensionality as the action space and the number of UAVs increase [31], which constrains the number of nodes in the UAVs network.

Consequently, we propose a distributed cooperative trajectory optimization algorithm based on MADRL and the attention mechanism to solve the address problems.

3. System Model

This paper considers an emergency communication scenario with a limited number of ground stations in complex terrain. UAVs are deployed as aerial base stations to provide stable communication for GUs. The system establishes a mesh network to dynamically extend the wireless network range by combining UAVs with emergency ground network base stations, as shown Figure 1. N UAVs are deployed and the set is denoted as $\mathcal{N} = \{1, 2, ..., N\}$, and the location of UAV$_i$ is denoted as $l_i(t) = [x_i(t), y_i(t), z_i(t)] \in \mathbb{R}^{3\times 1}, i \in \mathcal{N}$. To reduce the complexity of the model, the flight altitude of the UAVs is fixed, $z_i(t) = H, \forall i \in \mathcal{N}$. K GUs are distributed randomly, the set and location of GU$_k$ are denoted as $\mathcal{K} = \{1, 2, ..., K\}$ and $l_k(t) = [x_k(t), y_k(t), 0] \in \mathbb{R}^{3\times 1}, \forall k \in \mathcal{K}$, respectively. There are two main types of GUs: fast-moving rescue vehicles and slower-moving personnel in ever-changing directions. For a better reading of this paper, the important symbols are listed in Table 1.

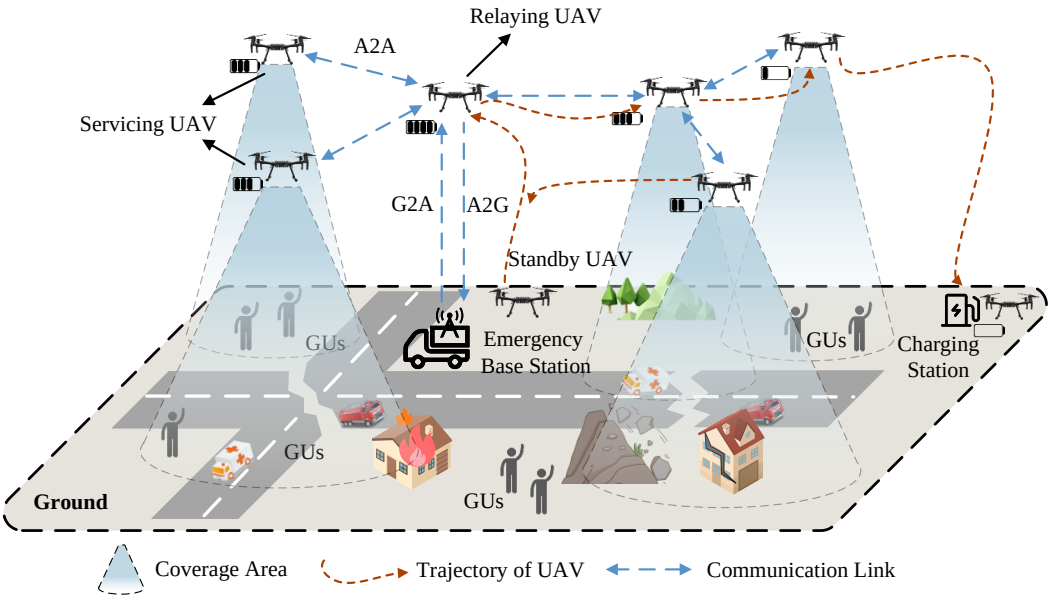

Figure 1. Multi-UAVs emergency communication scenario.

Table 1. Table of Important Symbols.

Symbol	Description	Symbol	Description
$\mathcal{N}$	Set of UAVs.	$l_i(t)$	The location of UAV_i.
$\mathcal{K}$	Set of GUs.	$l_k(t)$	The location of GU_k.
$R^{i,j}_{\text{A2A}}(t)$	Communication transmission rate between UAV_i and UAV_j.	$\gamma^{i,j}_{\text{A2A}}(t)$	Signal-to-Noise Ratio (SNR).
$R^{i,k}_{\text{A2G}}(t)$	Communication transmission rate between UAV_i and GU_k.	P^i_{com}	The communication power of UAV_i.
$E_i(T)$	Task power consumption of UAV_i.	$P^i_{\text{dyn}}(t)$	The flight power of UAV_i.
$\eta(T)$	Communication efficiency index of UAVs.	r_{com}	Communication reward.
r_{power}	Energy consumption reward.	r_{safe}	Safety reward.
π_{θ_i}, Q_{ψ_i}	Actor and Critic networks.	$\pi_{\bar{\theta}_i}$, $Q_{\bar{\psi}_i}$	Target Actor and Critic networks.
θ_i, ψ_i	The parameters of Actor and Critic networks.	$\bar{\psi}_i$, $\bar{\theta}_i$	The parameters of Target Actor and Critic networks.
$\nabla_{\theta_i} J_m(\pi_\theta)$	The strategy gradient.	$\mathcal{L}_Q(\psi_i)$	The loss function of Critic networks.
γ, ε	The discount factor of reward and the soft update factor.	α	The factor of action entropy.
$L^{i,k}_{\text{LoS}}(t)$, $L^{i,k}_{\text{NLoS}}(t)$	The average path loss for LoS and NLoS links.	$P^{i,k}_{\text{LoS}}(t)$	The probability of the LoS connection.
D_{safe}	Safety distance.	λ_{safe}	Safe speed factor.
S_m	The set of UAVs state.	A_m	The set of UAVs actions.
o^i_m	The local information of UAV_i	a^i_m	The action of UAV_i.

3.1. Communication Model

In this paper, two channel models are considered: the Air-to-Air (A2A) model for communication among UAVs, and the Air-to-Ground (A2G) model for communication among UAVs and GUs [32].

The communication links among UAVs are primarily a Line-of-Sight (LoS) connection, and the link loss model between UAV_i and UAV_j can be characterized as a free space propagation loss. The communication transmission rate between UAV_i and UAV_j is formulated as

$$R^{i,j}_{\text{A2A}}(t) = B\log_2\left(1+\gamma^{i,j}_{\text{A2A}}(t)\right), \tag{1}$$

where $\gamma^{i,j}_{\text{A2A}}(t)$ is the Signal-to-Noise Ratio (SNR), B is link bandwidth.

In the emergency communication environment, there are shadowing effects and reflection of signals from obstacles. Thus, the A2G channel is modeled by considering the LoS and Non-Line-of-Sight (NLoS) components [8]. The link loss between UAV_i and GU_k is calculated as

$$L^{i,k}_{\text{avg}}(t) = P^{i,k}_{\text{LoS}}(t)\times L^{i,k}_{\text{LoS}} + \left(1-P^{i,k}_{\text{LoS}}(t)\right)\times L^{i,k}_{\text{NLoS}}, \tag{2}$$

where $L^{i,k}_{\text{LoS}}(t)$ and $L^{i,k}_{\text{NLoS}}(t)$ denote the average path loss for LoS and NLoS links, respectively. $P^{i,k}_{\text{LoS}}(t)$ is the probability that the LoS connection is related to environmental factors [33]. The communication transmission rate between UAV_i and GU_k is given by

$$R^{i,k}_{\text{A2G}}(t) = B\log_2\left(1+\gamma^{i,k}_{\text{A2G}}(t)\right). \tag{3}$$

3.2. Energy Consumption Model of UAVs

The energy consumption of UAVs is mainly classified into two categories: communication energy consumption and flight energy consumption [34]. The communication energy consumption includes the processing and transmission of signals, which is much less than the flight energy consumption of UAVs. As a result, to simplify the complexity of

the system model, the power of the communication component is fixed as P^i_{com}. The flight energy consumption is mainly affected by speed and acceleration [34]. The total power consumption of UAV$_i$ is formulated as

$$E_i(T) = \int_0^T \left(P^i_{\text{dyn}}(t) + P^i_{\text{com}}\right) dt, \tag{4}$$

where $P^i_{\text{dyn}}(t)$ is the flight power of UAV$_i$ at time t, T is total mission time.

3.3. Problem Formulation

In this paper, the communication efficiency of UAVs is maximized while meeting the transmission rate requirements of GUs communication. To avoid a crash with other UAVs or obstacles, UAVs need to automatically avoid obstacles. To maximize the communication efficiency of UAVs, UAVs need to optimize trajectories and reduce unnecessary maneuvers. Based on the above models, the optimization problem $\mathbb{P}1$ is given as

$$\begin{aligned}
(\mathbb{P}1): \max_{\{l_i(t)\}_{\forall i \in \mathcal{N}}} \eta(T) = & \frac{\sum_{k=1}^{M} \int_o^T R^{i,k}_{\text{A2G}}(t) \mathrm{d}t}{\sum_{i=1}^{N} E_i(T)}, \\
s.t.\ & C1: E_i(T) \leq e_{\text{safe}}, \\
& C2: R^{i,k}_{\text{A2G}}(t) > R_{\min},\ R^{i,j}_{\text{A2A}}(t) > R_{\min}, \\
& C3: l_i(t) \notin \Omega_i,\ l_i(t) \notin \Omega_{\text{obs}}, \\
& C4: l_i(t),\ l_k(t) \in \Omega_{\text{task}}, \\
& C5: V_i < V_{\max}, \\
& C6: a_i < a_{\max}, \\
& \text{where } \forall \mathrm{i}, \mathrm{j} \in \mathcal{N},\ \forall k \in \mathcal{K},\ \forall t \in T,
\end{aligned} \tag{5}$$

where $l_i(t)$ is location of the UAV$_i$ at time t, $R^{i,k}_{\text{A2G}}(t)$ is communication transmission rate of GUs, $E_i(T)$ is total mission power consumption of UAV$_i$, and $\eta(T)$ is the communication efficiency index of UAVs . Constraint $C1$ is a safety energy limit to ensure that the UAVs preserve enough energy to return, where e_{safe} is the safety power of UAVs. The safety power is set flexibly depending on the size of the task area. in this paper. Constraint $C2$ indicates that the network nodes must achieve a certain level of communication transmission rate to meet the communication requirement of GUs, where $R_{\min}$ is the minimum value of the communication transmission rate established among network nodes. $C3$ is a safety constraint for UAVs, where Ω_i is the collision field for UAV$_i$, and $\Omega_{\text{obs}} \in \mathbb{R}^{3\times 1}$ is the obstacle collision field. Constraint $C4$ is the movement area constraints of UAVs and GUs, where $\Omega_{\text{task}} \in \mathbb{R}^{3\times 1}$ is the task field of UAVs. Constraints $C5$ and $C6$ are maximum speed and acceleration limits for UAVs, where $V_{\max}$ and $a_{\max}$ are the maximum speed and acceleration of UAVs, which are related to the parameters of the practical UAVs. The papers [35,36] were referred to set the change threshold values.

Problem $\mathbb{P}1$ is a mixed-integer optimization problem that is difficult to be solved by traditional trajectory optimization algorithms. Therefore, problem $\mathbb{P}1$ is formulated as a kind of Markov game problem that can be solved by employing an MADRL algorithm.

3.4. Problem Transformation

In this paper, the continuous problem $\mathbb{P}1$ is discretized and divided the task time T into M time slots δ_t, where $m = 0, 1, 2, ..., M$, $T = M\delta_t$. Due to the relatively small size of each time slot, the locations, policies, and network parameters of the UAVs are considered to be constant. Problem $\mathbb{P}1$ is transformed into

$$
\begin{aligned}
(\mathbb{P}2): \max_{\{l_i(m)\}_{\forall i \in \mathcal{N}}} \eta(M) &= \frac{\sum_{k=1}^{K}\sum_{m=1}^{M} R_k(m)}{\sum_{i=1}^{N}\sum_{m=1}^{M} E_i(m)}, \\
s.t.\ C1 &: E_i(M) \leq e_{\text{safe}}, \\
C2 &: R^{i,k}_{\text{A2G}}(m) > R_{\min},\ R^{i,j}_{\text{A2A}}(m) > R_{\min}, \\
C3 &: l_i(m) \notin \Omega_i,\ l_i(m) \notin \Omega_{\text{obs}}, \\
C4 &: l_i(m),\ l_k(m) \in \Omega_{\text{task}}, \\
C5 &: V_i < V_{\max}, \\
C6 &: a_i < a_{\max}, \\
&\text{where } \forall \text{i,j} \in \mathcal{N},\ \forall k \in \mathcal{K},\ \forall m \in M.
\end{aligned} \tag{6}
$$

There are five basic elements in Markov games, $\{S, A, P, R, \gamma\}$which are defined as follows:

S represents the set of UAVs state $s^i_m \in S$, where s^i_m is the state of the UAVs, $s^i_m = \{\{l_i\}, \{d_{i,j}(m), d_{i,k}(m), d_{i,obs}(m)\}, R_i(m), E_i(m)\}_{\forall i,j \in \mathcal{N},\ \forall \text{k} \in \mathcal{K}}$, and $l_i(m)$ is location of UAV_i at time slot m. The relative distance between UAV_i and UAV_i is denoted as $d_{i,j}(m)$. The relative distance between UAV_i and GU_k is denoted as $d_{i,k}(m)$. The relative distance between UAV_i and obstacles is calculated as $d_{i,obs}(m)$. The communication transmission rate and the remaining power of the UAV_i are denoted as $R_i(m)$ and $E_i(m)$, respectively.

A represents the set of UAVs actions $a^i_m \in A$, where $a^i_m = \{F_i(m) \in \mathbb{R}^{3\times 1}\}_{\forall i \in \mathcal{N},\ \forall m \in M}$.

P represents the state transfer function. The large state space of the model makes it difficult to predict the transfer probability of a specific state.

γ represents the discount factor of reward, which is employed to adjust the decay rate of future rewards.

R is defined as the reward function of the model, which is necessary for DRL to complete the training and directly influences the performance of the model. The reward function is divided into local rewards and global rewards of UAVs. The local rewards are awarded to UAVs for completing their own tasks. The global rewards are awarded to all UAVs upon meeting certain conditions to promote cooperation among UAVs. Three reward functions involving communication, energy, and safety are considered:

- Communication reward function is defined as

$$
r^{i,m}_{\text{com}} = \begin{cases} 0, if\ R^{i,k}_{\text{A2G}}(m) < R_{\min} \\ r_g, if\ R^{j,k}_{\text{A2G}}(m) \geq R_{\min} \text{ and } \text{R}^{\text{i,j}}_{\text{A2A}}(\text{m}) \geq \text{R}_{\min} \\ r_c + r_g, if\ R^{i,k}_{\text{A2G}}(m) \geq R_{\min},\ \forall i,j \in \mathcal{N},\ \forall k \in \mathcal{K},\ \forall m \in M \end{cases}, \tag{7}
$$

  where r_c is the local reward obtained by the UAV_i when establishing a connection with the GUs, and r_g is global connectivity reward for all UAVs on this link.
- Energy consumption reward function is defined as

$$
r^{i,m}_{\text{power}} = \begin{cases} \mu E_{\text{UAV}_i}(m), if\ E_{\text{UAV}_i}(m) > e_{\text{safe}} \\ 0, \text{otherwise},\ \forall \text{i} \in \mathcal{N},\ \forall m \in M \end{cases}, \tag{8}
$$

  where μ is the power reward factor, and e_{safe} is the safety power threshold of UAVs. In this paper, the remaining power of UAVs is adopted as an energy consumption reward to optimize UAV trajectories and reduce unnecessary maneuvers to preserve power.

- Safety reward function is defined as

$$r_{\text{safe}}^{i,m} = \begin{cases} \dfrac{-\eta}{d_{i,\text{obs}}(m) + \Delta d}, if\ d_{i,\text{obs}}(m) \leq D_{\text{safe}} + \lambda_{\text{safe}} v_i \\ 0, \text{otherwise}, \forall i \in \mathcal{N}, \forall m \in M \end{cases}, \tag{9}$$

where D_{safe} is the safety distance threshold, and Δd is a small value to ensure that the denominator is non-zero, and v_i and λ_{safe} are speed and safe speed factor of UAV_i. UAV obstacle avoidance is an important function for the safety of the whole system. Therefore, this paper establishes a safe reward function to improve the obstacle avoidance capability of UAVs. UAVs and obstacles are set up with potential fields whose ranges adjust dynamically with speed. The safe reward will reduce as UAVs approach the center of the potential field.

4. The DSAAC Algorithm

The multi-UAVs cooperative trajectory optimization problem is transformed into Markov games, which is a multi-agent extension of the Markov decision. The DSAAC algorithm is proposed based on multi-agent reinforcement learning. Figure 2 shows the process of multi-UAV cooperative optimization of communication efficiency. In the task initialization phase, all UAVs initialize the location and network status. In the second step, UAVs cooperate flight according to network communication rate, obstacle avoidance, and energy consumption. The UAVs outputs actions based on their state information and the relative distance of other UAVs. The cooperation between UAVs is realized by sensing the relative distance. In the third and fourth step, the DSAAC algorithm is used to optimize the cooperative policy of the UAV to maximize communication efficiency. The strategy models of UAVs are optimized by centralized training. Finally, the models of the UAVs are updated for the next iteration.

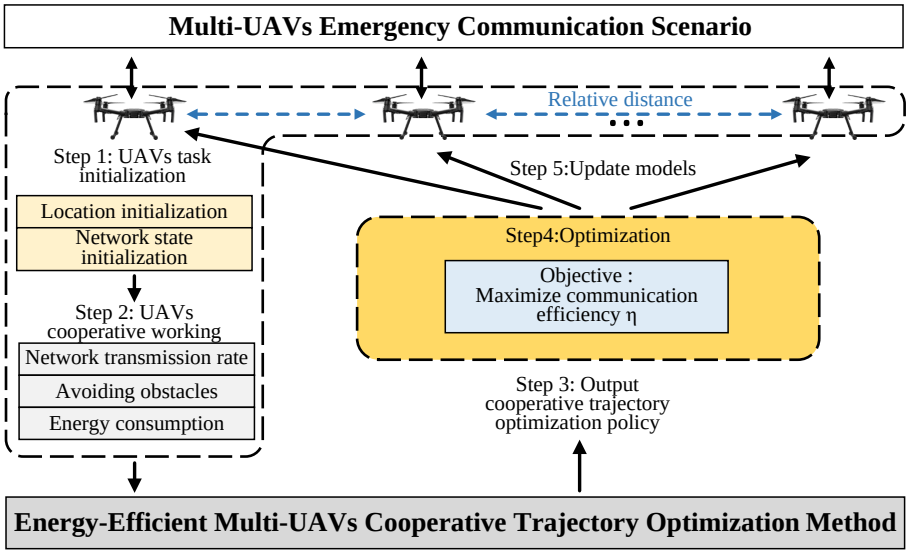

Figure 2. The flowchart of the DSAAC algorithm.

4.1. Framework of the DSAAC Algorithm

Assume that UAVs have a set of state $\{o_m^1, o_m^2, \ldots, o_m^N\} \in S_m$, where o_m^i is the local information observed by UAV_i and A_m is the set of actions $\{a_m^1, a_m^2, \ldots, a_m^N\}$ at time slot m. $P(S_{m+1}|S_m, A_m)$ is defined as the probability of making UAVs perform action A_m in state

S_m and transfer to state S_{m+1}. $R(S_m, A_m)$ is defined as the reward obtained by causing agent to perform action A_m in state S_m. Expected discounted return function is formulated as

$$J_i(\pi_i) = \mathrm{E}_{a_1 \sim \pi_1, \ldots, a_N \sim \pi_N}\left[\sum_{m=0}^{M} \gamma_m R_m^i\left(S_m, a_m^1, a_m^2, \ldots, a_m^N\right)\right], \tag{10}$$

where $R_m^i\left(S_m, a_m^1, a_m^2, \ldots, a_m^N\right)$ is the reward obtained by all UAVs in the S_m state after performing action $\{a_m^1, a_m^2, \ldots, a_m^N\}$, and π_i is the policy function for UAV$_i$.

The policy function of traditional MADRL outputs all action probabilities and selects the action with the maximum probability [36]. However, the adoption of deterministic strategies is difficult to adapt to the dynamic environment, and thus the performance of the agent degrades rapidly when the environment changes. The introduction of action sampling entropy will result in greater policy bandwidth, and UAVs will quickly learn new policies when environmental changes. Therefore, the DSAAC algorithm introduces the action sampling entropy that is inversely proportional to the probability of the selected action. The action sampling entropy encourages UAVs to explore new strategies in a dynamic environment of multi-UAVs cooperation [37]. The strategy gradient formula for introducing action entropy is given as follows

$$\nabla_{\theta_i} J_m(\pi_\theta) = \mathrm{E}_{o \sim B, a \sim \pi}\left[\nabla_{\theta_i} \log(\pi_{\theta_i}(a_m^i | o_m^i))\left(Q_{\psi_i}\left(o_m^{all}, a_m^{all}\right) - \alpha \log\left(\pi_{\theta_i}\left(a_m^i | o_m^i\right)\right)\right)\right], \tag{11}$$

where π_{θ_i} is the Actor network of UAV$_i$, which can output the probability value of each action, and Q_{ψ_i} is the Critic network of UAV$_i$. o_m^{all} and a_m^{all} are states and actions of all UAVs, respectively. We use a CTDE architecture whose Critic network shares a loss function and jointly updates the network parameters to minimize the error values, where α is the coefficient of action entropy and is employed to characterize the degree of exploration of the UAVs. θ_i and ψ_i are the network parameters of Actor and Critic network of UAV$_i$, respectively. The action entropy of UAV$_i$ is formulated as $\log a_m^i = \log\left(\pi_{\theta_i}\left(a_m^i | o_m^i\right)\right)$. The experience $\{o_m^i, a_m^i, o_{m+1}^i, r_m^i\}$ at time slot m is stored into the replay pool B.

The loss function of the Critic network can be formulated as

$$\mathcal{L}_Q(\psi_i) = \sum_{i=1}^{N}\left(Q_{\psi_i}\left(o_m^{all}, a_m^{all}\right) - y_m^i\right)^2, \tag{12}$$

$$\text{where } y_m^i = r_m^i + \gamma Q_{\bar{\psi}_i}\left(o_{m+1}^{all}, a_{m+1}^{all}\right) - \alpha \log\left(\pi_{\bar{\theta}_i}\left(a_{m+1}^i | o_{m+1}^i\right)\right),$$

$$\overline{\theta_i} = \varepsilon\theta_i + (1-\varepsilon)\overline{\theta_i},\ \overline{\psi_i} = \varepsilon\psi_i + (1-\varepsilon)\overline{\psi_i}, \varepsilon \in [0,1],$$

where $Q_{\bar{\psi}}$ is Target-Critic network, and $\pi_{\bar{\theta}_i}$ is Target-Actor network. $\bar{\psi}_i$ and $\bar{\theta}_i$ are the network parameters of the Target-Critic and Target-Actor network of UAV$_i$, which are updated by soft updates, where ε is the soft update factor. The structure of the DSAAC algorithm is shown in Figure 3. In the task initialization phase, all UAVs initialize the location and network status. In the second step, UAVs cooperate flight according to network communication rate, obstacle avoidance, and energy consumption. In the third and fourth step, the DSAAC algorithm is used to optimize the cooperative policy of the UAV to maximize communication efficiency. Finally, the models of the UAVs are updated for the next iteration.

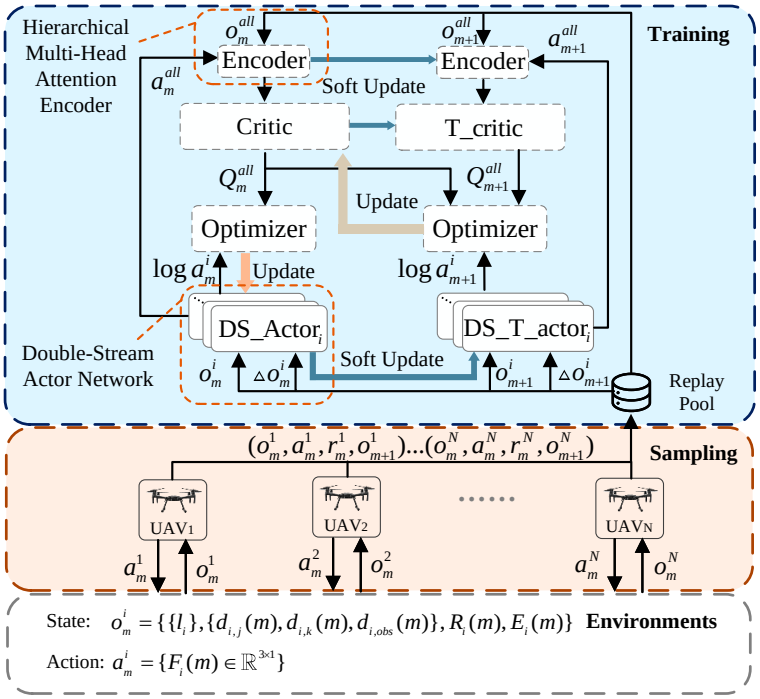

Figure 3. The architecture of the DSAAC algorithm.

4.2. Double-Stream Actor Network

The autonomous flight of UAVs is key in the problem of deploying emergency networks by UAVs. Therefore, UAVs need to have certain obstacle-avoidance capabilities. The traditional Actor network of DRL has great performance in static scenarios, but the performance will degrade in highly dynamic scenarios. This is due to the fact that the agent lacks the ability to sense dynamic information and can only make action decisions based on the current state of information. For example, when the target or obstacle is dynamic, UAVs cannot determine whether they are moving away from or closer to the target from the current distance information [38]. Additionally, UAVs need to establish a highly cooperative relationship with each other when making cooperative decisions with other UAVs based on their trajectories and states. Therefore, the differential of state is added into the Actor network to make the UAVs with certain dynamic sensing capabilities, which is structured in Figure 4. The double-stream Actor network consists of the Multi-Layer Perceptron (MLP) layer, the Batch Normalization (BN) layer, and the residual connection. The MLP layer provides basic perception capability for the Actor network. The residual connection prevents vanishing gradients. The BN layer is utilized to improve the training speed.

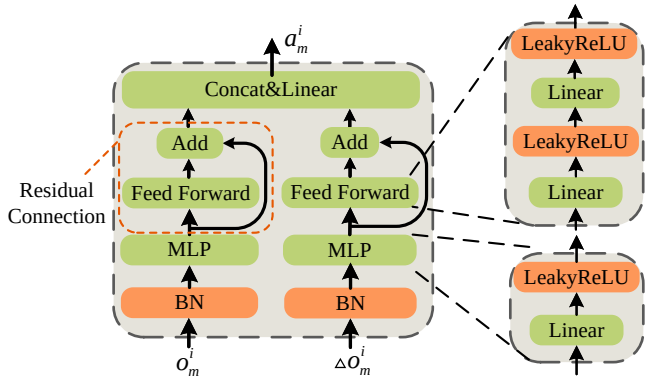

Figure 4. The structure of Double-Stream Actor network.

The differentiation of the state can be calculated as

$$\Delta o_m^i = \begin{cases} o_m^i - o_{m-1}^i, & if m > 0 \\ 0, & if m = 0 \end{cases}. \tag{13}$$

The difference between the state o_m^i of time slot m and the state o_{m-1}^i of the last time slot is obtained as Δo_m^i. With additional information Δo_m^i, the UAVs can detect the movement trend of obstacles or other UAVs and take appropriate actions in advance. Additionally, differential information can be used to track the signal trend of the links in real time. Examples include jumps in the connection status of network nodes and communication transmission rate changes of nodes.

4.3. Hierarchical Multi-Head Attention Encoder

The DSAAC adopted a CTDE structure. UAVs observe local information to complete the cooperative task after the model is well-trained, without the need for centralized control of the UAVs. Thus, it is easier to deploy our algorithms to practical application scenarios. During training, the Critic network evaluates action based on the state of all UAVs and their corresponding actions to adjust the strategies. However, the CTDE structure leads to two problems: (1) As the numbers of both UAVs and GUs increase, the problem of the dimensional curse will arise; (2) The information about unrelated UAVs will interfere with building complex cooperative relationships.

This paper proposes a hierarchical multi-head attention encoder based on the transformer encode [39], whose structure is illustrated in Figure 5. This encoder consists of the Layer Normalization (LN) layer, the FeedForward layer, the Multi-Head Attention layer, and the residual connection. The LN layer improves the training speed as well as the BN layer, but it can avoid the effect of batch data. In addition to handling semantic problems in different sentences, the attention mechanism also works on multi-agent tasks. In a multi-agent environment, the role of a single agent needs to be derived from the state of other agents. Through an attention mechanism, the encoder increases the information encoding weight of associated UAVs. The information about irrelevant UAVs is suppressed to reduce interference. Thus the Critic network can more correctly evaluate the role of the movements of the UAV in cooperation with other UAVs. As a result, the correct assistance relationship can be established.

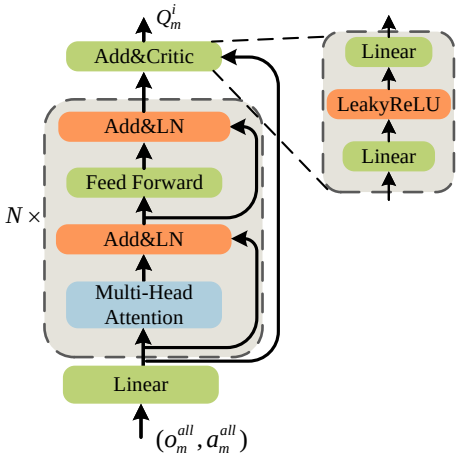

Figure 5. The structure of Hierarchical Multi-Head Attention encoder.

Figure 6 shows the application of self-attention in MADRL. The calculation formula is shown below

$$q_i = W_q e_i,\ Q = (q_1, q_2, ..., q_n), \tag{14}$$

$$k_i = W_k e_i,\ K = (k_1, k_2, ..., k_n), \tag{15}$$

$$v_i = W_v e_i,\ V = (v_1, v_2, ..., v_n), i \in N, \tag{16}$$

where e_i is obtained by embedding (o_m^{all}, a_m^{all}) through the Linear layer. Each header of the multi-headed attention module has three weight matrices: W_q, W_k and W_v, which are multiplied with e_i to obtain queries q_i, keys k_i and values v_i, respectively.

The soft attention weights are calculated as

$$\alpha_{\text{soft}} = \text{Softmax}(\frac{K^{\text{T}}Q}{\sqrt{d_k}}), \tag{17}$$

where α_{soft} is a vector of soft attention weights, and d_k is the attention scaling factor to prevent the gradient disappearance. As a result, the output matrix of the multi-head attention layer is given as follows

$$H = V\alpha_{\text{soft}}, H \in \{h_1, h_2, ..., h_N\}, \tag{18}$$

where h_i incorporates information about the attention-weighted other UAVs.

4.4. Training of the DSAAC Algorithm

In this section, the training process of the DSAAC algorithm will be detailed. The training procedure of the DSAAC algorithm based on the three-layer framework is given in Algorithm 1, which is described as follows:

Algorithm 1 DSAAC Algorithm

1: **Input**: UAVs state $s_m^i = \{\{l_i\}, \{d_{i,j}(m), d_{i,k}(m), d_{i,obs}(m)\}, R_i(m), E_i(m)\}_{\forall i,j \in \mathcal{N}, \forall k \in \mathcal{K}}$;
2: **Output**: UAVs actor $a_m^i = \{F_i(m) \in \mathbb{R}^{3\times 1}\}_{\forall i \in \mathcal{N},\ \forall m \in M}$;
3: ▷**Initialization**;
4: Initialize the Actor,Critic,target Actor and target Critic network with weights θ_i, ψ_i, $\overline{\theta_i}$, $\overline{\psi_i}$ for each UAV$_i$ in N, and experience replay buffer B;
5: **for** each episode in E **do**
6: Initialize the state of the UAV$_i$, and environment;
7: Receive the initial state $s_1 = \{o_1, ..., o_N\}$;
8: **for** each step m in M **do**
9: ▷**Experience sampling**;
10: **for** each step m in M **do**
11: Select action $a_m^i = \pi_{\theta_i}(a_m^i | o_m^i) + \eta$;
12: **end for**
13: UAVs execute their actions $a_m = (a_m^1, ..., a_m^N)$;
14: Receive next state s_{m+1}, and obtain reward $r_m = (r_m^1, ..., r_m^N)$;
15: Update s_m from s_{m+1};
16: Store (s_m, a_m, r_m, s_{m+1}) in the buffer B;
17: ▷**Parameter updating**;
18: **for** each UAV$_i$ in $\mathcal{N}$ **do**
19: Sample L random mini-batches $(s_m, a_m, r_m, s_{m+1}) \in B$;
20: Update weights θ_i,$\psi_i$by Equations (11) and (12);
21: Soft update weights by: $\overline{\theta_i} = \varepsilon\theta_i + (1-\varepsilon)\overline{\theta_i}$, $\overline{\psi_i} = \varepsilon\psi_i + (1-\varepsilon)\overline{\psi_i}, \varepsilon \in [0, 1]$;
22: **end for**
23: **end for**
24: **end for**

Initialization (Line 4): Where η is the noise term to increase the robustness of the algorithm. At the beginning of the training phase, initialize the network parameters of the Actor and Critic networks for each UAV and copy the parameters of both networks to the Target-Actor and the Target-Critic networks [40]. The experience replay pool B is

instantiated. In addition, the status values of all UAVs and the environment are reset to their initial state.

Experience Sampling (Lines 9–16): In this phase, UAVs perform the corresponding action a_m^i in accordance with the local state o_m^i they observe. The state of the UAVs is transferred to the next time slot state o_{m+1}^i and is rewarded with r_m^i. The experience $\{s_m, a_m, r_m, s_{m+1}\}$ obtained is stored in the experience pool B for the next parameter update of the next time slot.

Parameter Updating (Lines 17–21): In this phase, the data of batch size are first randomly selected from the experience pool B and normalized. The network parameters of the Actor and Critic networks of each UAV are updated by the policy gradients Equation (11) and loss functions Equation (12). After each parameter update, the network parameters of the Actor and Critic network are synchronized to Target-Actor and Target-Critic networks by soft update.

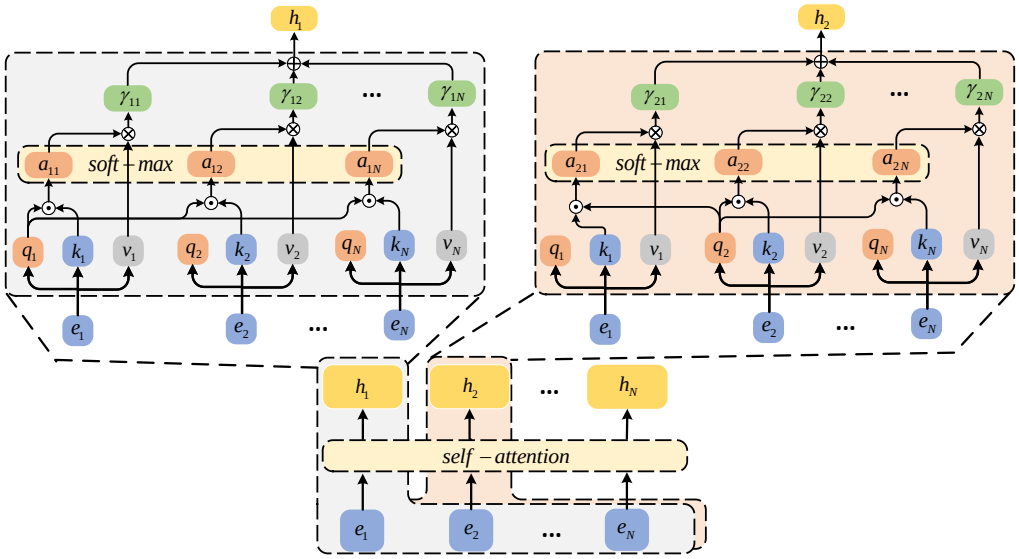

Figure 6. The structure of self-attention network.

4.5. Complexity Analysis

In this section, we examine the time and space complexity of DSAAC. The Actor network approximation consists of J fully connected layers, and the Critic network approximation consists of H fully connected layers. The time complexity of the Actor and Target-Actor network is given by

$$T_{\text{Actor}}^i = \sum_{j=0}^{J-1} (w_j \times w_{j+1} + 1), \tag{19}$$

the time complexity of the Critic and Target-Critic networks is given by

$$T_{\text{Critic}}^i = \sum_{h=0}^{H-1} (w_h \times w_{h+1} + 1), \tag{20}$$

where w_j and w_h are input dimension of the fully connected layer, w_{j+1} and w_{h+1} are output dimension of the fully connected layer. In summary, the time complexity T_{train} formula of DSAAC in the training phase is given by

$$T_{\text{train}} = \sum_{i=1}^{N} \left(2 \times T_{\text{Actor}}^i + 4 \times T_{\text{Critic}}^i\right), \tag{21}$$

the time complexity T_{eval} formula of DSAAC in the evaluation phase is calculated as

$$T_{\text{eval}} = \sum_{i=1}^{N} \left(2 \times T_{\text{Actor}}^{i}\right). \tag{22}$$

In the training phase, a cache needs to store the historical experience values of UAVs whose size is set to N_B. The total space complexity of the model in the training phase is given by

$$S_{\text{train}} = \sum_{i=1}^{N} \left(2 \times \sum_{j=0}^{J-1} \left(w_j \times w_{j+1} + 1\right) + 2 \times \sum_{h=0}^{H-1} \left(w_h \times w_{h+1} + 1\right)\right) + O(N_B). \tag{23}$$

In the validation phase, the Critic network experience pool is not present. Thus the total space complexity of the model in the evaluation phase is given by

$$S_{\text{eval}} = \sum_{i=1}^{N} \left(2 \times \sum_{j=0}^{J-1} \left(w_j \times w_{j+1} + 1\right)\right). \tag{24}$$

5. Performance Evaluation

In this section, we evaluate the performance of the distributed cooperative algorithm DSAAC for optimizing the trajectory of UAVs.

5.1. Simulation Settings

For training and testing, the experimental platform is built based on Ubuntu 20.04.4 server using PyTorch 1.7, Python 3.8, Intel Core i9-11900H, and NVIDIA GeForce RTX3090. This platform is built based on OpenAI multiagent particle environment. Figure 7 illustrates the simulation scenario. In a square area of 2 km × 2 km, the simple road network is constructed, and several GUs and obstacles are set. The ground GUs move along the road at a random speed, and the ground network base stations are set up in the central area. By relaying, UAVs cover the network of ground base stations to the area of GUs. UAVs are assigned the Actor network consisting of the artificial neural network, which is used to generate UAV actions based on real-time state information. In practical application scenarios, UAVs will be independently equipped with computing devices to run the Actor network. In the training phase, a centralized server is needed to train the UAVs and update the models. The experimental parameters are shown in Table 2.

Table 2. Simulation Settings.

Parameters	Values
Flight altitude (H)	50 m
Number of UAVs (N)	{2∼10}
Number of GUs (K)	{10∼30}
The weight of the UAV (M)	2 kg
Minimum transmission rate (R_{min})	1 Mbps
Safety distance (D_{safe})	5 m
Safe speed factor (λ_{safe})	0.1
UAV communication power (P_{com})	10 W
Maximum flight speed (V_{max})	20 m/s
Maximum flight acceleration (a_{max})	8 m/s^2
Safety power (e_{safe})	10%
Batch size (bs)	1024
Soft update rate (ε)	0.01
Discount factor (λ)	0.99
Learn rate (τ)	0.001
Coefficient of action entropy (α)	0.01

The input and output dimensions of the neural network will not match each other when the scenario changes. In practical applications, multiple scenario parameters are preset and multiple models are pre-trained for the rapid deployment in new scenarios.

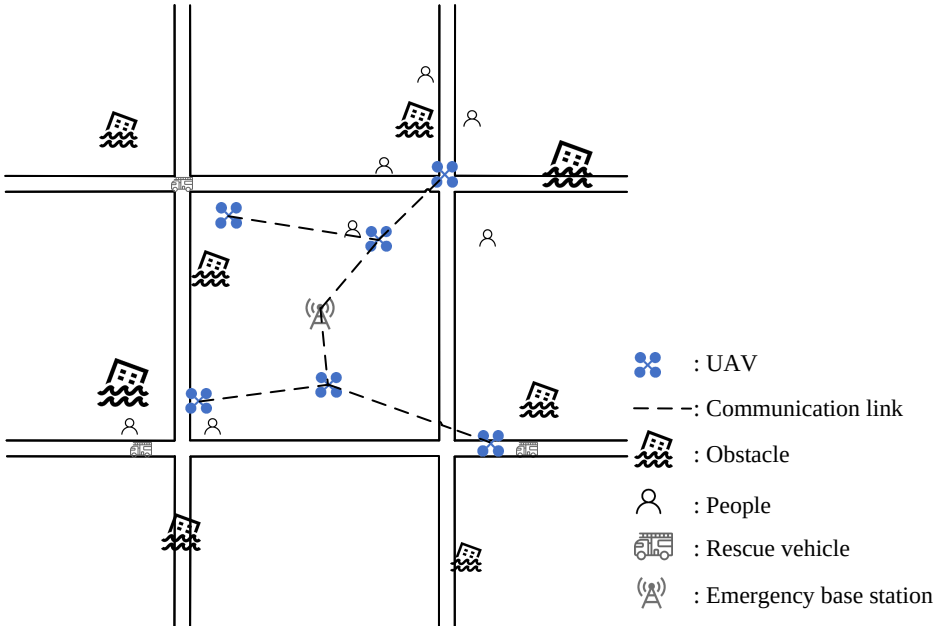

Figure 7. Simulation for the Multi-UAVs emergency network.

5.2. Result Analysis

In this section, the DSAAC algorithm is compared with MADDPG algorithm, MATD3 algorithm, and MASAC algorithm.

- MADDPG: It is a multi-agent reinforcement learning algorithm based on centralized training and decentralized execution architecture, which is widely applicable in multi-agent collaborative tasks. It effectively solves the problem in non-stationary environment during training, and a similar architecture is used in our algorithm. Reference [41] proposed the joint trajectory design algorithm for UAVs based on the MADDPG algorithm.
- MATD3: It uses Clipped Double-Q Learning and Target-Policy Smoothing to solve the problem that the MADDPG algorithm overestimates the Q-values. Thus MATD3 can obtain more robust cooperation strategies. In [42], the authors proposed an optimization algorithm based on the MATD3 for jointly designing trajectories, computation task allocation, and communication resource management of UAVs.
- MASAC: It is an extension of the SAC algorithm for multi-agent. The exploration ability of the agents is improved by encouraging them to choose inaccessible strategies. Thus MASAC can still achieve better performance in complex scenarios and its convergence is better. Reference [43] proposed an algorithm based on the MASAC for optimizing the task partitioning and power allocation strategies of UAVs.

To the best of our knowledge, these three algorithms, MADDPG, MATD3, and MASAC, are excellent representative algorithms in multi-agent cooperation. They have significant correlations with our algorithm in the application field and structure. Thus we adopt them as baseline algorithms to verify the performance of the DSAAC algorithm in a large-scale environment. In addition, the comparison with the three algorithms can verify that hierarchical multi-head attention encoder and double-stream actor network improve the performance of the algorithm.

Figure 8 shows the training curve of the DSAAC algorithm and the three baselines. We set up scenarios based on 3, 6, and 8 UAVs serving 10 ground GUs, respectively:

- Figure 8a shows the DSAAC algorithm and the baselines algorithm converge to the highest reward at around 30,000 episodes. The convergence values of DSAAC and MASAC are closer to better than MATD3 and MADDPG. This is because that action entropy is used to avoid local optimal solutions.
- Figure 8b shows that the training effect decreases when the number of UAVs increases to 6, indicating that the MASAC algorithm is more sensitive to the number of UAVs.
- Figure 8c shows that MASAC, MADDPG, and MATD3 reward curves all fail to converge in the scenario with 8 UAVs. This is because the increasing number of UAVs increases the complexity of the environment, and UAVs fail to cooperate effectively.
- Figure 8d shows that the performance of the algorithm in scenarios with a larger number of UAVs. We have adopted a larger scale of simulation and set up a scenario with 18 UAVs serving 30 GUs. The reward curve shows that the DSAAC algorithm can still learn the collaborative strategy, while the baseline algorithm has only random oscillations in the reward curve. In addition, the DSAAC algorithm oscillates in large round ranges. The reward curve converges slower when the scale of the scenario becomes larger. This is due to the fact that the learning of cooperative strategies requires constant exploration of trial. This phenomenon is more obvious when the number of UAVs increases. When some UAVs adjust their strategies, other UAVs require considerable time to learn new strategies.

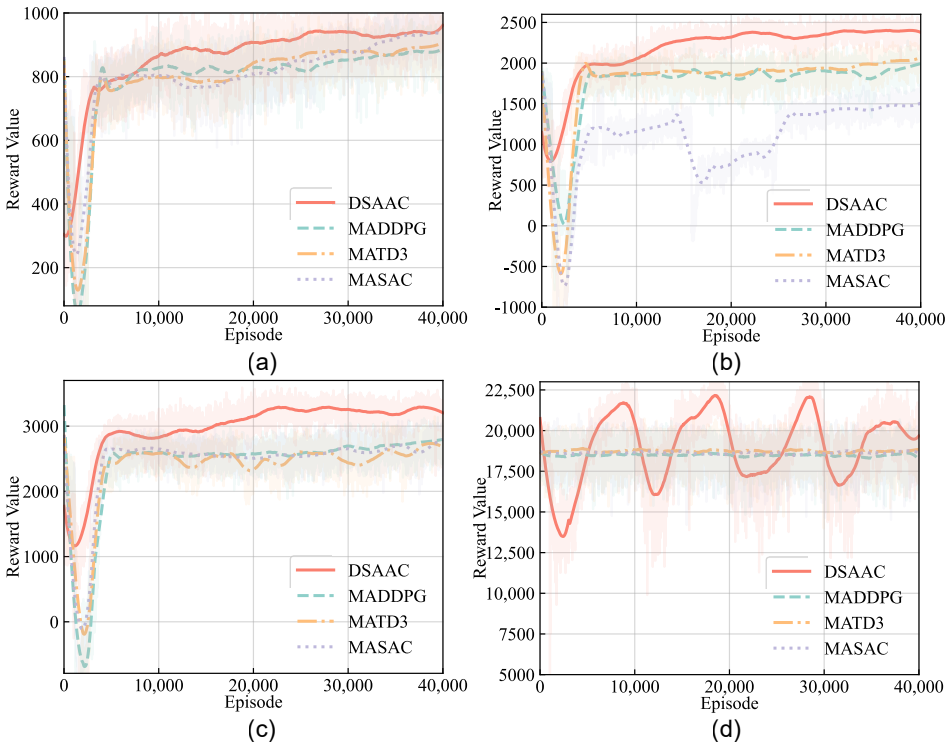

Figure 8. The training curve of reward for different numbers of UAVs: (**a**) 3 UAVs serving 10 GUs; (**b**) 6 UAVs serving 10 GUs; (**c**) 8 UAVs serving 10 GUs; (**d**) 18 UAVs serving 30 GUs.

Figure 9 shows that the DSAAC algorithm performs optimally for various numbers of UAVs. We used the average number of crashes, average system throughput, and power consumption as algorithm performance metrics. The crash is defined as the distance between UAV_i and UAV_j or an obstacle is less than the safe distance D_{safe}, and the system will generate a repulsive force to limit further approach.

- Figure 9a shows the average number of crashes of MASAC is smaller than that of DSAAC in the scenario with six UAVs. The main reason is that the MASAC algorithm cannot effectively perform the communication task and can only enhance the obstacle avoidance performance to improve the overall reward.

- Figure 9b shows the average throughput variation with the number of UAVs. When the number of UAVs is above 6, the average throughput no longer increases much. This proves the system requires at least six UAVs to achieve the throughput requirements of all users. The average throughput of the DSAAC algorithm is the highest in every scenario.
- Figure 9c shows the average energy consumption varies with the number of UAVs. The DSAAC algorithm has the lowest task energy consumption and is stable at about 32% in every scenario. The reason is that the UAVs have established an excellent cooperative relationship with each other UAVs and the tasks are properly distributed.

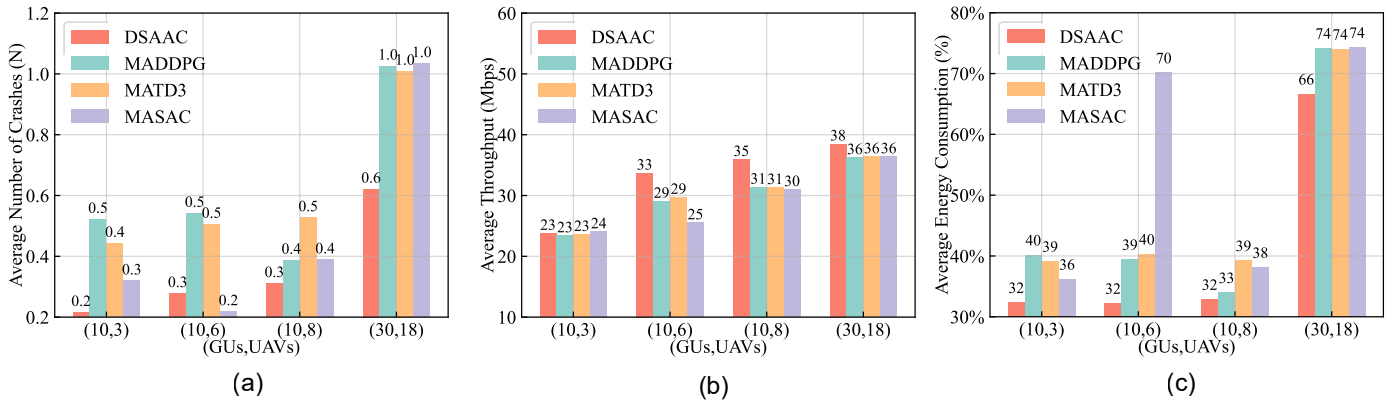

Figure 9. Performance comparison of different number of UAVs: (**a**) Average number of crashes; (**b**) Average system throughput; (**c**) Average energy consumption.

To verify the effect of different numbers of ground GUs on the algorithm, setting up scenarios based on six UAVs serving 10, 20, and 30 ground GUs, respectively.

- Figure 10 shows that DSAAC converges to the highest reward, and the reward convergence value increases roughly in proportion to the number of GUs. However, baseline algorithms converge with some decrease in reward value. This is because of the inclusion of the hierarchical multi-headed attention encoder in the Critic network of the DSAAC algorithm, which can remove the interference of redundant information and circumvent the curse of dimensionality to a certain extent.
- Figure 11a shows the effect on the UAV obstacle avoidance performance under different GUs scalability. Due to DSAAC having the Double-stream Actor network, UAVs can sense the trend of state change and make corresponding trajectory optimization in advance. The DSAAC algorithm is better than MADDPG and MATD3 in terms of obstacle avoidance performance, but the MASAC algorithm outperforms our algorithm in scenarios with more than 20 GUs for the same reason as Figure 8a.
- Figure 11b shows the impact on GUs throughput under different GUs. The DSAAC basically increases GUs throughput proportionally when the number of GUs increases, while MADDPG and MATD3 no longer increase in scenarios with more than 20 GUs, and MASAC decreases instead. Baseline algorithms fail to serve GUs well in more GUs scenarios. The reason is that when the number of GUs increases, the state space of UAVs also increases, which causes the dimensional explosion.
- Figure 11c shows the energy efficiency performance of the algorithm in different scenarios and demonstrates that our algorithm performs best in terms of average energy consumption.

The communication efficiency index of UAVs in the four algorithms with five combinations of different numbers of users and the number of UAVs is shown in Figure 12. The communication efficiency index is calculated as $\eta(T)$ in Equation (5). The communication efficiency index in the DSAAC algorithm is better than other algorithms, with 10.6% to 56.7%. The communication efficiency of UAVs in our algorithm tends to increase with the number of UAVs and GUs. However, the communication efficiency index of the baseline

algorithm decreases when the number of GUs is more than 20. The main reason is that UAVs do not establish practical cooperation and fail to build multi-hop relay networks to cover users over long distances.

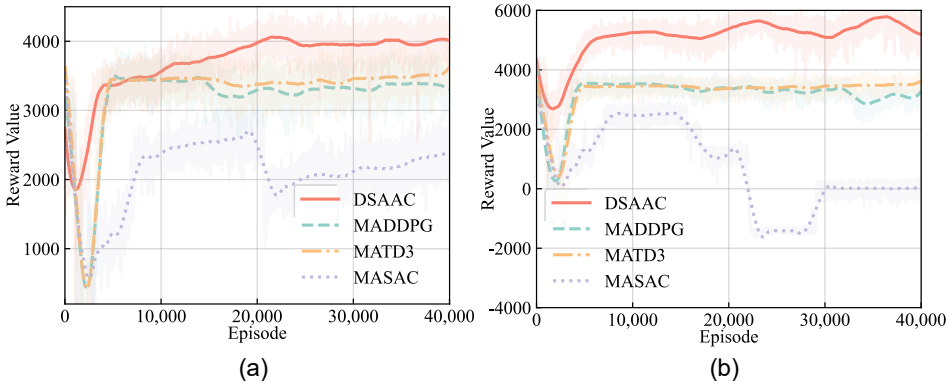

Figure 10. The training curve of reward for differents number of GUs: (**a**) 6 UAVs serving 20 GUs; (**b**) 6 UAVs serving 30 GUs.

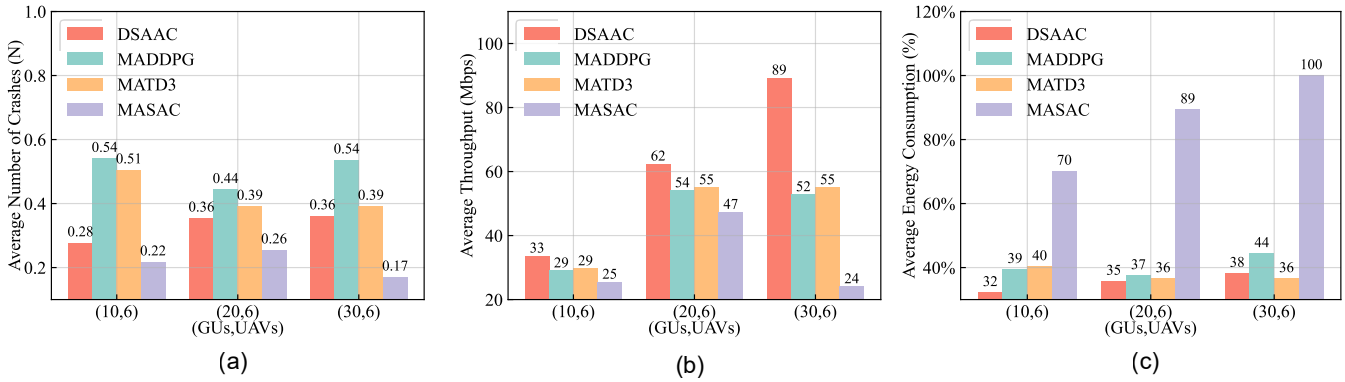

Figure 11. Performance comparison of different number of GUs: (**a**) Average number of crashes; (**b**) Average system throughput; (**c**) Average energy consumption.

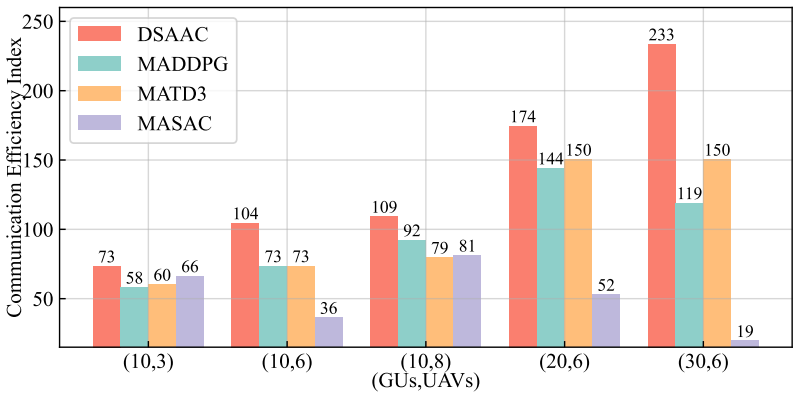

Figure 12. Communication efficiency index of UAVs.

6. Discussion

In this paper, we utilize multi-UAVs to provide emergency relay networks for GUs. This multi-UAVs cooperative system requires consideration of several problems: (1) How to maximize the efficiency of UAV communication. This is a multi-objective multi-constrained optimization problem, which is difficult to solve by traditional methods. Thus the DRL method is utilized to solve it. However, the DRL method cannot obtain the optimal solution,

but the approximate optimal model parameters are obtained by means of heuristic learning. The experimental results show that our algorithm is stable and effective compared to the baseline algorithms, and the highest communication efficiency can be obtained. (2) How to establish an effective cooperative strategy among multi-UAVs. The degree of correlation among UAVs is different in multi-node UAV relay networks. Due to the problem of a non-stationary environment caused by separately optimized UAV strategies, a joint optimization mechanism between multiple UAVs is needed. In addition, the UAV needs to deploy a distributed algorithm to ensure the reliability of the system. Therefore, we adopt the CTDE architecture, in which the UAVs only need to complete cooperative tasks based on their own local information during the execution phase. However, the CTDE architecture requires a fixed number of UAVs for training. Moreover, the systems with different numbers of UAVs need to train the corresponding models separately, which reduces the flexibility in practical projects. (3) How to improve the safety and dynamic performance of UAVs. When the UAV only responds to the state of the current time slot, it cannot sense whether the obstacle is far away or close. Thus, the double-stream Actor network is constructed to solve this problem.

In this paper, we have only verified the proposed DSAAC algorithm in the software simulation platform. The following factors may need to be considered to validate this algorithm in a practical experimental evaluations. First, deploying UAVs to verify algorithm performance in a real environment requires a more accurate physical model. in this paper, connectivity is mainly considered in the communication model of UAVs. The connection is established when a certain communication rate is reached between UAVs. We ignored the impact of UAVs attitude and altitude on communication. In addition, the communication energy consumption model is set to a constant value to simplify the energy consumption model. The simplification of models has some influence on the communication stability and rate in practical UAVs deployment. Thus it is necessary to establish a more accurate physical model according to the accurate hardware platform in practical scenarios. Second, the deployment of reinforcement learning algorithms in a practical environment requires a reasonable choice of observation space. If the observation space is excessively high, the learned strategy can easily overfit the simulated environment [44]. Next, it is necessary to build a map of the mission area prior to identifying the coordinates of the obstacles. However, this can be used in practice with satellite maps to obtain information about obstacles in a practice scenario. Finally, the UAVs need to avoid fine-grained or dynamic obstacles in a real-time manner.

In this paper, connectivity is mainly considered in the communication model of UAVs. The connection is established when a certain communication rate is reached between nodes. In the scenario of our paper, ensuring rigorous real-timeliness is not the main concern, and the latency of several seconds is allowed. Therefore, we give less consideration of latency and mainly focus on the optimization of connectivity among UAVs.

Although our algorithm reduces the crash probability in most cases, it cannot achieve perfect crash avoidance due to the formation of a Nash equilibrium by multiple optimization objectives. In future work, we will design a safe MADRL algorithm based on a local trajectory planning algorithm to improve the safety of UAVs in complex environments.

7. Conclusions

In this paper, we have studied the problem of optimizing the multi-UAVs cooperative trajectory for maximizing the communication efficiency in UAVs dynamic deployment emergency communication scenario. First, to improve the dynamic performance and obstacle avoidance of the UAVs in complex and changing scenarios, we have designed a double data stream network in the Actor network of the UAVs. Further, to establish an effective cooperation strategy for UAVs, we have designed a hierarchical multi-headed attention encoder in the Critic network. This encoder has effectively solved the problem of the curse of dimensionality when the number of UAVs and GUs increases. Finally, the simulation experiments have shown that the DSAAC algorithm improved communication efficiency by 56.7% , throughput by 71.2%, energy saving by 19.8%, and reduced the number

of crashes by 57.7%. UAVs are energy-sensitive agents, and improving the operation time of multi-UAV systems has been an important research direction. In our future work, we will study the charging scheduling problem of multi-UAVs to achieve the total power of the system dynamically maintained in a safe range.

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

Funding: This work was supported by the National Natural Science Foundation of China (No. 62176088), the Program for Science & Technology Development of Henan Province (No. 212102210274, 222102210022, 222102520028), and the Young Elite Scientist Sponsorship Program by Henan Association for Science and Technology (No. 2022HYTP013).

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

Abbreviations

The following abbreviations are used in this manuscript:

UAVs	Unmanned Aerial Vehicles
GUs	Ground Users
DSAAC	Double-Stream Attention multi-agent Actor-Critic
MADRL	Multi-Agent Deep Reinforcement Learning
DRL	Deep Reinforcement Learning
A2A	Air-to-Air
A2G	Air-to-Ground
SNR	Signal-to-Noise Ratio
CTDE	Centralized Training Decentralized Execution
MLP	Multi-Layer Perceptron
BN	Batch Normalization
LN	Layer Normalization

References

1. Guo, Q.; Peng, J.; Xu, W.; Liang, W.; Jia, X.; Xu, Z.; Yang, Y.; Wang, M. Minimizing the Longest Tour Time Among a Fleet of UAVs for Disaster Area Surveillance. *IEEE Trans. Mob. Comput.* **2022**, *7*, 2451–2465. [CrossRef]
2. Qadir, Z.; Ullah, F.; Munawar, H.S.; Al-Turjman, F. Addressing Disasters in Smart Cities Through UAVs Path Planning and 5G Communications: A Systematic Review. *Comput. Commun.* **2021**, *2*, 114–135. [CrossRef]
3. Wang, Y.; Su, Z.; Luan, T.H.; Li, R.; Zhang, K. Federated Learning with Fair Incentives and Robust Aggregation for UAV-Aided Crowdsensing. *IEEE Trans. Network Sci. Eng.* **2022**, *9*, 3179–3196. [CrossRef]
4. Mozaffari, M.; Saad, W.; Bennis, M.; Nam, Y.H.; Debbah, M. A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Commun. Surv. Tutorials* **2019**, *9*, 2334–2360. [CrossRef]
5. Cao, X.; Yang, P.; Alzenad, M.; Xi, X.; Wu, D.; Yanikomeroglu, H. Airborne Communication Networks: A Survey. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 1907–1926.
6. Pasha, J.; Elmi, Z.; Purkayastha, S.; Fathollahi-Fard, A.M.; Ge, Y.E.; Lau, Y.Y.; Dulebenets, M.A. The Drone Scheduling Problem: A Systematic State-of-the-Art Review. *IEEE Trans. Intell. Transp. Syst.* **2022**, *3*, 14224–14247. [CrossRef]
7. Liu, C.H.; Ma, X.; Gao, X.; Tang, J. Distributed Energy-Efficient Multi-UAV Navigation for Long-Term Communication Coverage by Deep Reinforcement Learning. *IEEE Trans. Mob. Comput.* **2020**, *6*, 1274–1285. [CrossRef]
8. Zhao, H.; Wang, H.; Wu, W.; Wei, J. Deployment Algorithms for UAV Airborne Networks Toward On-Demand Coverage. *IEEE J. Sel. Areas Commun.* **2018**, *36*, 2015–2031. [CrossRef]
9. Zhang, G.; Ou, X.; Cui, M.; Wu, Q.; Ma, S.; Chen, W. Cooperative UAV Enabled Relaying Systems: Joint Trajectory and Transmit Power Optimization. *IEEE Trans. Green Commun.* **2022**, *3*, 112–123. [CrossRef]
10. Fadlullah, Z.M.; Kato, N. HCP: Heterogeneous Computing Platform for Federated Learning Based Collaborative Content Caching Towards 6G Networks. *IEEE Trans. Emerg. Top. Comput.* **2022**, *1*, 112–123. [CrossRef]
11. Ding, R.J.; Xu, Y.D.; Gao, F.F.; Shen, X.M. Trajectory Design and Access Control for Air-Ground Coordinated Communications System with Multiagent Deep Reinforcement Learning.*IEEE Internet Things J.* **2022**, *4*, 5785–5798. [CrossRef]
12. Zhang, T.; Lei, J.; Liu, Y.; Feng, C.; Nallanathan, A. Trajectory Optimization for UAV Emergency Communication with Limited User Equipment Energy: A Safe-DQN Approach. *IEEE Trans. Veh. Technol.* **2022**, *8*, 9107–9112. [CrossRef]

13. Saxena, V.; Jalden, J.; Klessig, H. Optimal UAV Base Station Trajectories Using Flow-Level Models for Reinforcement Learning. *IEEE Trans. Cognit. Commun.* **2019**, *5*, 1101–1112. [CrossRef]
14. Samir, M.; Ebrahimi, D.; Assi, C.; Sharafeddine, S.; Ghrayeb, A. Leveraging UAVs for Coverage in Cell-Free Vehicular Networks: A Deep Reinforcement Learning Approach. *IEEE Trans. Mob. Comput.* **2021**, *2*, 2835–2847. [CrossRef]
15. Wu, S.; Xu, W.; Wang, F.; Li, G.; Pan, M. Distributed Federated Deep Reinforcement Learning based Trajectory Optimization for Air-ground Cooperative Emergency Networks. *IEEE Trans. Veh. Technol.* **2008**, *10*, 142–149. [CrossRef]
16. Liu, X.; Liu, Y.; Chen, Y.; Hanzo, L. Trajectory Design and Power Control for Multi-UAV Assisted Wireless Networks: A Machine Learning Approach. *IEEE Trans. Veh. Technol.* **2019**, *8*, 7957–7969. [CrossRef]
17. Wang, H.; Pu, Z.; Liu, Z.; Yi, J.; Qiu, T.; A Soft Graph Attention Reinforcement Learning for Multi-Agent Cooperation. In Proceedings of the 16th IEEE International Conference on Automation Science and Engineering (CASE), Electr Network, Hong Kong, China, 20–21 August 2020.
18. Noor, F.; Khan, M.A.; Al-Zahrani, A.; Ullah, I.; Al-Dhlan, K.A. A Review on Communications Perspective of Flying Ad-Hoc Networks: Key Enabling Wireless Technologies, Applications, Challenges and Open Research Topics. *Drones* **2020**, *4*, 65. [CrossRef]
19. Wu, Q.; Zeng, Y.; Zhang, R. Joint Trajectory and Communication Design for Multi-UAV Enabled Wireless Networks. *IEEE Trans. Wirel. Commun.* **2018**, *3*, 2109–2121. [CrossRef]
20. Sun, Z.; Yang, D.; Xiao, L.; Cuthbert, L.; Wu, F.; Zhu, Y. Joint Energy and Trajectory Optimization for UAV-Enabled Relaying Network with Multi-Pair Users. *IEEE Trans. Cognit. Commun.* **2021**, *7*, 939–954. [CrossRef]
21. Huang, Z.; Chen, C.; Pan, M. Multiobjective UAV Path Planning for Emergency Information Collection and Transmission. *IEEE Internet Things J.* **2020**, *7*, 6993–7009. [CrossRef]
22. Li, T.; Zhu, K.; Nguyen Cong, L.; Niyato, D.; Wu, Q.; Zhang, Y.; Chen, B. Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey. *IEEE Commun. Surv. Tutorials* **2022**, *6*, 1240–1279. [CrossRef]
23. Wang, C.; Wang, J.; Shen, Y.; Zhang, X. Autonomous Navigation of UAVs in Largescale Complex Environments: A Deep Reinforcement Learning Approach. *IEEE Trans. Veh. Technol.* **2019**, *3*, 2124–2136. [CrossRef]
24. Ding, R.; Gao, F.; Shen, X. 3D UAV Trajectory Design and Frequency Band Allocation for Energy-Efficient and Fair Communication: A Deep Reinforcement Learning Approach. *IEEE Trans. Wirel. Commun.* **2020**, *12*, 7796–7809. [CrossRef]
25. Wang, H.; Zhao, H.; Wu, W.; Xiong, J.; Ma, D.; Wei, J. Deployment Algorithms of Flying Base Stations: 5G and Beyond with UAVs. *IEEE Internet Things J.* **2019**, *12*, 10009–10027. [CrossRef]
26. Wang, L.; Zhang, H.; Guo, S.; Yuan, D. 3D UAV Deployment in Multi-UAV Networks with Statistical User Position Information. *IEEE Commun. Lett.* **2022**, *6*, 1363–1367. [CrossRef]
27. Shi, W.; Li, J.; Wu, H.; Zhou, C.; Cheng, N.; Shen, X. Drone-Cell Trajectory Planning and Resource Allocation for Highly Mobile Networks: A Hierarchical DRL Approach. *IEEE Internet Things J.* **2021**, *6*, 9800–9813. [CrossRef]
28. Cui, J.; Liu, Y.; Nallanathan, A. Multi-Agent Reinforcement Learning-Based Resource Allocation for UAV Networks. *IEEE Trans. Wirel. Commun.* **2020**, *2*, 729–743. [CrossRef]
29. Yin, S.; Yu, F.R. Resource Allocation and Trajectory Design in UAV-Aided Cellular Networks Based on Multiagent Reinforcement Learning. *IEEE Internet Things J.* **2022**, *2*, 2933–2943. [CrossRef]
30. Xia, Z.; Du, J.; Wang, J.; Jiang, C.; Ren, Y.; Li, G.; Han, Z. Multi-Agent Reinforcement Learning Aided Intelligent UAV Swarm for Target Tracking. *IEEE Trans. Veh. Technol.* **2022**, *1*, 931–945. [CrossRef]
31. Kondo, T.; Ito, K. A Reinforcement Learning with Evolutionary State Recruitment Strategy for Autonomous Mobile Robots Control. *Rob. Auton. Syst.* **2004**, *2*, 111–124. [CrossRef]
32. Rahman, S.U.; Kim, G.H.; Cho, Y.Z.; Khan, A. Positioning of UAVs for Throughput Maximization in Software-Defined Disaster Area UAV Communication Networks. *J. Commun. Netw.* **2018**, *10*, 452–463. [CrossRef]
33. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is All You Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
34. Bor-Yaliniz, R.I.; El-Keyi, A.; Yanikomeroglu, H. Efficient 3-D Placement of an Aerial Base Station in Next Generation Cellular Networks. In Proceedings of the IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016.
35. Yang, K.; Dong, W.; Tong, Y.; He, L. Leader-follower Formation Consensus of Quadrotor UAVs Based on Prescribed Performance Adaptive Constrained Backstepping Control. *Int. J. Control Autom. Syst.* **2022**, *10*, 3138–3154. [CrossRef]
36. Zeng, Y.; Xu, J.; Zhang, R. Energy Minimization for Wireless Communication with Rotary-Wing UAV. *IEEE Trans. Wirel. Commun.* **2019**, *2*, 2329–2345. [CrossRef]
37. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-Agent Actor-Critic for Mixed Cooperative-Competitive Environments. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
38. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor. In Proceedings of the 35th International Conference on Machine Learning (ICML), Stockholm, Sweden, 10–15 July 2018.
39. Zhang, S.; Li, Y.; Dong, Q. Autonomous Navigation of UAV in Multi-obstacle Environments Based on a Deep Reinforcement Learning Approach. *Appl. Soft Comput.* **2022**, *1*, 115. [CrossRef]

40. Zhang, W.; Yang, D.; Wu, W.; Peng, H.; Zhang, N.; Zhang, H.; Shen, X. Optimizing Federated Learning in Distributed Industrial IoT: A Multi-Agent Approach. *IEEE J. Sel. Areas Commun.* **2021**, *10*, 3688–3703. [CrossRef]
41. Zhang, Y.; Mou, Z.; Gao, F.; Jiang, J.; Ding, R.; Han, Z. UAV-Enabled Secure Communications by Multi-Agent Deep Reinforcement Learning. *IEEE Trans. Veh. Technol.* **2020**, *10*, 11599–11611. [CrossRef]
42. Zhao, N.; Ye, Z.; Pei, Y.; Liang, Y.C.; Niyato, D. Multi-Agent Deep Reinforcement Learning for Task Offloading in UAV-Assisted Mobile Edge Computing. *IEEE Trans. Wirel. Commun.* **2022**, *9*, 6949–6960. [CrossRef]
43. Cheng, Z.; Liwang, M.; Chen, N.; Huang, L.; Du, X.; Guizani, M. Deep Reinforcement Learning-Based Joint Task and Energy Offloading in UAV-aided 6G Intelligent Edge Networks. *Comput. Commun.* **2022**, *8*, 234–244. [CrossRef]
44. Tan, J.; Zhang, T.; Coumans, E.; Iscen, A.; Bai, Y.; Hafner, D.; Bohez, S.; Vanhoucke, V. Sim-to-Real: Learning Agile Locomotion For Quadruped Robots. In Proceedings of the 14th Conference on Robotics—Science and Systems, Carnegie Mellon Univ, Pittsburgh, PA, USA, 26–30 June 2018.