future internet

MDPI

Article

Optimizing Drone Energy Use for Emergency Communications in Disasters via Deep Reinforcement Learning

Wen Qiu [1], Xun Shao [2], Hiroshi Masui [1,*] and William Liu [3]

[1] Information Processing Center, Kitami Institute of Technology, Kitami 090-8507, Japan; clorisqiu1@gmail.com
[2] Department of Electrical and Electronic Information Engineering, Toyohashi University of Technology, Toyohashi 441-8580, Japan; x-shao@ieee.org
[3] Department of Information Technology and Software Engineering, School of Engineering, Computer and Mathematical Sciences, Unitec Institute of Technology, Auckland 1025, New Zealand; wliu@unitec.ac.nz
* Correspondence: hgmasui@mail.kitami-it.ac.jp

Abstract: For a communication control system in a disaster area where drones (also called unmanned aerial vehicles (UAVs)) are used as aerial base stations (ABSs), the reliability of communication is a key challenge for drones to provide emergency communication services. However, the effective configuration of UAVs remains a major challenge due to limitations in their communication range and energy capacity. In addition, the relatively high cost of drones and the issue of mutual communication interference make it impractical to deploy an unlimited number of drones in a given area. To maximize the communication services provided by a limited number of drones to the ground user equipment (UE) within a certain time frame while minimizing the drone energy consumption, we propose a multi-agent proximal policy optimization (MAPPO) algorithm. Considering the dynamic nature of the environment, we analyze diverse observation data structures and design novel objective functions to enhance the drone performance. We find that, when drone energy consumption is used as a penalty term in the objective function, the drones—acting as agents—can identify the optimal trajectory that maximizes the UE coverage while minimizing the energy consumption. At the same time, the experimental results reveal that, without considering the machine computing power required for training and convergence time, the proposed key algorithm demonstrates better performance in communication coverage and energy saving as compared with other methods. The average coverage performance is 10–45% higher than that of the other three methods, and it can save up to 3% more energy.

Keywords: drones; multi-agent deep reinforcement learning (MADRL); energy optimization; emergency communications

Citation: Qiu, W.; Shao, X.; Masui, H.; Liu, W. Optimizing Drone Energy Use for Emergency Communications in Disasters via Deep Reinforcement Learning. *Future Internet* **2024**, *16*, 245. https://doi.org/10.3390/fi16070245

Academic Editor: Gianluigi Ferrari

Received: 5 June 2024
Revised: 3 July 2024
Accepted: 5 July 2024
Published: 11 July 2024

1. Introduction

In the wake of natural disasters—such as earthquakes, hurricanes, and floods—the conventional communication infrastructure is often severely damaged or completely destroyed. This disruption impedes rescue operations, coordination efforts, and the dissemination of critical information, thereby exacerbating a crisis. Ensuring reliable communication in such scenarios is critical for effective disaster response and recovery. Consequently, there is a pressing need for innovative solutions that can swiftly restore the communication services in the affected areas during a natural disaster [1]. Unmanned aerial vehicles (UAVs), commonly known as drones, have emerged as a versatile tool in various domains, including disaster management [2,3]. Drones have taken the market by storm, with sales expected to grow to USD 4.28 billion by 2025, which is three times the amount in 2018 [4]. Their ability to operate independently of the ground infrastructure and their rapid deployment ability make them ideal candidates for establishing emergency communication networks [5]. By functioning as relay stations, UAVs can create temporary communication links, extending the coverage to areas where the infrastructure is compromised. However,

the efficient deployment and operation of UAVs in disaster scenarios present significant challenges. The dynamic nature of the environment, varying user densities, and the limited battery life of drones necessitate a strategic approach to their deployment. Optimizing the flight paths of drones to maximize user coverage while minimizing energy consumption is a complex problem that requires advanced computational techniques [6].

This paper proposes a novel solution to this problem by employing deep reinforcement learning (DRL) algorithms to control the movement trajectories of drones. DRL, a subset of machine learning, is well-suited for problems involving sequential decision-making under uncertainty [7]. By training drones to learn the optimal strategy, our approach ensures that they can adapt to real-time changes in the environment and user equipment (UE) distribution, taking into account the energy consumption of the drones. Based on this, it can be said that drones have the capability of providing efficient and reliable communication services.

The primary contributions of this research are described below.

- To address drone-assisted emergency communications in disaster scenarios, we first modeled the movement of rescue workers in post-disaster situations. Then, based on this model, we developed a DRL-based algorithm specifically for the service of drones in disaster scenarios.
- During the communication between drones and UEs, we used a novel signal to interference and noise ratio (SINR) calculation method, taking into account the communication interference generated between the drones. We set and analyzed the communication threshold to ensure QoS and used this to calculate the coverage of drones to UEs.
- We carefully designed the reward function and considered both coverage and energy consumption terms to ensure that the system provides motivating reward values.
- We conducted extensive simulations to evaluate the performance of our approach, demonstrating significant improvements in user coverage and energy efficiency compared to the conventional methods.

The remainder of this paper is organized in the following manner: Section 2 reviews the related work in the field of UAV-based communication and reinforcement learning applications. Section 3 details the system model and the problem formulation. Section 4 illustrates the proposed DRL algorithm, training techniques, and design of the reward function. Section 5 presents the simulation setup, results, and a discussion of the findings. Finally, Section 6 concludes the paper and outlines future research directions.

2. Related Work

The use of UAVs in emergency communication networks has received significant attention in recent years [8]. Numerous studies have explored various aspects of deploying UAVs for disaster response, including optimal placement, trajectory planning, and energy efficiency [9]. This section reviews the related work in the areas of UAV-based communication systems, trajectory optimization, energy management, and the application of reinforcement learning in UAV control.

UAVs have been extensively studied for their potential to establish temporary communication networks in post-disaster areas. Sharvari et al. (2023) propose the multi-hop opportunistic 3D routing (MO3DR) algorithm to address post-disaster routing challenges such as coverage requirements, inter-UAV collision avoidance, and reliable multi-hop routing without trajectory planning. Their simulations validate that maintaining the UAVs within a threshold inter-UAV distance effectively meets the coverage and collision constraints and thus maximizes the expected progress of data toward the terrestrial base station (TBS) [10]. Zhang et al. (2023) propose an air–ground cooperation architecture based on an ad hoc UAV network to address the challenges of damaged ground servers in disaster scenarios. They define system cost as a weighted sum of task delay and energy consumption and propose a joint optimization algorithm that iteratively solves the task scheduling and UAV deployment sub-problems. Their simulation results demonstrate

that the proposed algorithm significantly reduces task delay and energy consumption while achieving a good trade-off between these metrics for diverse tasks [11]. These studies highlight the importance of UAVs in maintaining communication services when the ground infrastructure is unavailable or damaged.

Trajectory optimization is a critical aspect of UAV deployment and directly impacts the efficiency and effectiveness of the communication network. Several approaches have been proposed to address this challenge. For example, Pan et al. (2023) [12] address the trajectory planning problem in their work on joint power and 3D trajectory optimization for UAV-enabled wireless powered communication networks (WPCNs) in the presence of obstacles. They decompose the problem into two sub-problems: power allocation and 3D trajectory optimization. The authors propose an improved non-dominated sorting genetic algorithm-II with a K-means initialization operator and variable dimension mechanism (NSGA-II-KV) for power allocation as well as an improved particle swarm optimization (PSO-NGDP) for trajectory optimization. Their approach effectively increases the number of covered wireless devices, enhances time efficiency, and reduces UAV flight distance, thereby demonstrating significant improvements in the energy utilization efficiency in complex environments [13]. Similarly, Zhang et al. [13] introduce a heuristic crossing search-and-rescue optimization algorithm (HC-SAR) for UAV path planning, which integrates a heuristic crossover strategy with a basic SAR algorithm to improve convergence speed and maintain population diversity. The HC-SAR algorithm demonstrates high performance in both two-dimensional and three-dimensional environments, significantly outperforming the traditional algorithms, such as differential evolution (DE) and ant lion optimizer (ALO), in terms of path length and fuel efficiency [14].

The application of DRL in UAV trajectory optimization is a rapidly growing field. Na et al. (2023) [11] propose an improved PSO algorithm for the energy-efficient path planning of UAVs in mountainous terrain. By integrating a deep deterministic policy gradient (DDPG) model for adaptive parameter tuning, the algorithm significantly enhances the global search capability and avoids local optima. The simulation results demonstrate that this approach effectively reduces the nonessential energy consumption and improves the UAV mission efficiency in complex environments [12]. Li et al. (2023) [14] address the problem of computation and communication uncertainties in multi-UAV-assisted mobile edge computing (MEC) networks. This paper proposes a robust design to minimize the total weighted energy consumption by jointly optimizing the UAV trajectory, task partition, and resource allocation using a multi-agent proximal policy optimization (MAPPO) with a Beta distribution framework. The numerical results reveal the effectiveness and robustness of the proposed algorithm in minimizing the energy consumption under various uncertainties [15]. These studies reveal that DRL algorithms can adapt to changing environmental conditions and complex problems, thus making them well-suited for disaster scenarios.

Optimizing energy consumption in UAV networks is critical for prolonging the operation time and enhancing the overall system efficiency. Sun et al. (2023) [15] address the challenge of maximizing the energy efficiency in a wireless power transfer (WPT)-enabled UAV-assisted emergency communication system. The UAV functions as a base station, performing both communication and wireless charging tasks. The authors propose a low-complexity alternating iterative optimization algorithm that jointly optimizes the UAV trajectory, transmit power, WPT power, and user bandwidth. Their simulations demonstrate that this approach effectively balances the system throughput and UAV energy consumption, significantly improving the energy efficiency compared to the benchmark schemes [16]. Ao et al. (2023) [16] propose an innovative approach for energy-efficient multi-UAV cooperative trajectory optimization. Their multi-agent deep reinforcement learning (MADRL)-based algorithm, called double-stream attention multi-agent actor–critic (DSAAC), significantly improves the communication efficiency and energy savings by leveraging a hierarchical multihead attention encoder and a double data stream network structure in the actor network. The simulation results reveal a notable reduction in energy consumption and an increase in system robustness [17].

The existing body of research emphasizes the potential of UAVs in emergency communication networks and highlights the challenges associated with their deployment. Although the traditional optimization methods have made significant contributions, the advent of reinforcement learning, particularly DRL, offers promising new avenues for research. Our work builds on these foundations, using DRL to develop a robust and adaptive solution to optimize the trajectory of UAVs in disaster scenarios. By addressing the limitations of previous approaches, we aim to provide a comprehensive framework that enhances both coverage and energy efficiency, ultimately improving the resilience and effectiveness of emergency communication networks.

3. System Model and Problem Formulation

3.1. Communication Scenario

As depicted in Figure 1, we consider a rectangular disaster area with line-of-sight (LoS) characteristics and damaged communication infrastructure. In this area, a set of drones $\mathcal{U}$—$u = 1, 2, \ldots, U$—serves as mobile base stations to provide services for ground UEs. We use (x_u, y_u, H) to represent the position of the drone, u, where H represents the height of the drone. This article assumes that all drones fly on a horizontal plane at a constant height from the ground. Each drone is equipped with a fixed-capacity lithium battery and can only provide service for a limited time. When a drone's energy falls below a certain threshold, it will seamlessly switch with a backup drone. We also assume that each drone has a high-capacity fronthaul link, such as a millimeter-wave link, to a ground base station equipped with an agent central unit. This central processing agent receives the drones' observations of the dynamic environment and their own status information. It then stably learns the optimal trajectory strategy to minimize energy consumption and manages the cooperation among the deployed drones. As illustrated in Figure 2, we consider a time period divided into $2T$ time slots. At the beginning of each time slot, the drone first moves to a new position, with the duration of this time slot being uncertain. In the next time slot, the drone hovers and provides communication services for a duration of Δt.

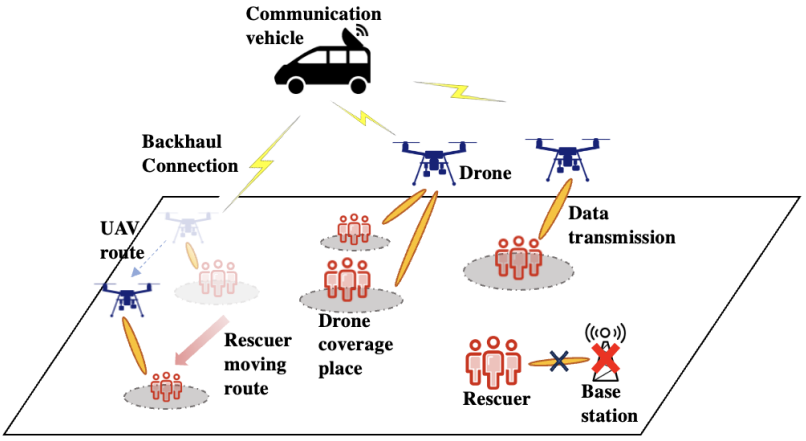


Figure 1. Drone-assisted communication region.

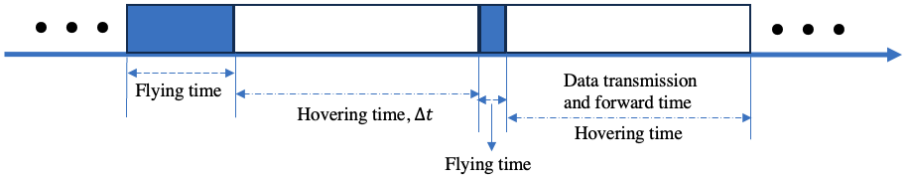


Figure 2. Drone-assisted communication region.

In this study, we consider a fixed number of rescuers, $\mathcal{N}$—$n = 1, 2, \ldots, N$—beginning from different rescue centers to conduct detailed searches of specified regions. The mission

of drones is to cooperate within a specified time period to provide the maximum possible communication coverage to these rescuers. We assume that the drone can communicate simultaneously with multiple UEs within its coverage area and perform interference-free communication by allocating appropriate orthogonal resources. In this post-disaster scenario, our subsequent analysis relies on the following basic assumptions:

- Each drone is randomly distributed in the certain region at the initial time. When a drone is almost exhausted, it retains sufficient energy to return to the charging station and then seamlessly switches with a backup drone. For simplicity, we keep the drone numbers the same before and after the switch. If a drone fails, we ignore the arrival time of the backup drone and also assume a seamless switch by default.
- Whether each drone can provide services to a particular UE depends on the number of UEs within its coverage. Additionally, each UE covered in the scenario is guaranteed a specific quality of service (QoS).
- The energy consumption of drones is mainly determined by flight and hovering. In this scenario, the energy consumption for communication is small and, thus, ignored [18].

3.2. User Movement Model

In this article, rescue workers are randomly distributed in different locations in various rescue centers. Their goal is to conduct a blanket search of the designated area. We assume that the rescuers' movement follows a Gaussian distribution with mean μ and variance σ^2, and that they are constantly in action throughout the period. We use (x_n, y_n, h) to represent the position coordinates of the UE n, where h represents the height of the UE. We assume that the height of all the UEs from the ground is h.

3.3. Channel Model

We calculate the mean path loss as the propagation loss of the wireless signal according to [19]. Path loss is divided into free space path loss and additional loss [20]. Here, we only consider the LoS situation between the drone and the UE:

$$PL = 20log(\frac{4\pi f_c d}{c}) + \eta, \tag{1}$$

where f_c indicates carrier frequency, d represents the Euclidean distance between drone and UE, c is the speed of light, and η denotes the mean additional loss for LoS.

Then, the received signal power P_r for UE n from drone u can be formulated as

$$P_{nu}^r = P_t - PL, \tag{2}$$

where P_t is the total transmit power from drones.

Thus, the signal to interference and noise ratio, SR_{nu}, for a drone–UE pair can be formulated as

$$SR_{nu} = \frac{P^r}{P_N + \sum_{i=1, i\neq u}^{U} P_{ni}^r}, \tag{3}$$

where P_N represents the additive white Gaussian noise power.

In practice, if the SR_{nu}^t served by drone u in time slot t is greater than the threshold SR_{th} and the number of UEs served by the drone does not exceed the upper limit, the UE n is considered to be covered by the drone u with acceptable QoS. If the number of users served by the drone u has reached the upper limit or the QoS is lower than the threshold, the next closest drone will be tried for use. If all drones cannot provide services, the UE is considered disconnected in time slot t.

3.4. The Drone Energy Consumption Model

We calculate the energy consumption of the drone during horizontal flight, lifting, and levitation, with the energy consumption during levitation being related to wind speed. The power consumed by the drone when flying horizontally with speed v can be

calculated in three parts: the power required to overcome the drag of the rotor blade profile, the fuselage that hinders the forward motion of the aircraft, and the power required to lift the payload [17]. Adding these three terms together, we obtain

$$P_h(v) = N_R P_b(1 + \frac{3v^2}{v_{tip}^2}) + \frac{1}{2} C_D A_f \rho(H) v^3 + W(\sqrt{\frac{W^2}{4N_R^2 \rho^2(H) A_r^2} + \frac{v^4}{4}} - \frac{v^2}{2})^{\frac{1}{2}}, \quad (4)$$

where W is the weight of the drone, and N_R and v_{tip} are the number of drone rotors and the tip speed of the rotor, respectively. C_D is the drag coefficient; A_f and A_r are the fuselage area and the rotor disc area. $P_b = \frac{\Delta}{8}\rho(H) s A_r v_{tip}^3$, Δ represents the profile drag coefficient, ρ is the air density function, and $\rho(H) = (1 - -2.2558 \cdot 10^{-5} H)^{4.2577}$.

The drone power consumed in a vertical climb with speed v_c is

$$P_v(v_c) = \frac{W}{2}(v_c + \sqrt{v_c^2 + \frac{2W}{N_R \rho(H) A_r}}) + N_R P_b. \quad (5)$$

When hovering, a horizontal speed, v_{hov}, is needed to counteract the wind speed. The hovering power consumption is provided as $P_h(v_{hov})$, which is in accordance with Equation (4).

3.5. The MDP Model

In this study, our objective is to control the trajectories of drones so that they provide maximum coverage for the UEs and minimize the consumption of energy from the drones. The drones must dynamically adjust their positions based on the distribution of the rescuers and the environmental conditions. The drone makes a decision in each time slot, and the decision in time slot t only depends on the scenario information at time $t-1$. This satisfies the Markov property, and the information observed by each drone is local; thus, we can model the problem as a partially observable Markov decision process (POMDP) [21]. The POMDP can be described as a tuple $< \mathcal{U}, \mathcal{S}, \mathcal{O}, \mathcal{A}, P, \boldsymbol{\pi}, R, \gamma >$; here, $\mathcal{U} = 1, 2, \ldots, U$, $\mathcal{S}$, $\mathcal{O} = o^1, \ldots, o^U$, and $\mathcal{A} = \mathcal{A}^1 \times \ldots \times \mathcal{A}^U$ are the set of corresponding drone agents, global state, the set of observations, and joint action, respectively. P represents the transition function, and $\gamma \in [0, 1)$ is the discount factor. At each time step, agent u receives observation o^u and provides action $a^u \in \mathcal{A}^u$. The details of the fundamental elements of our problem are provided below.

- Agents: The agents correspond to the drones. Each agent has an actor network, which determines the agent's action based on the input observation at each time step.
- Observations: The local observation information of each drone includes the position coordinates of the drone and the UEs it serves, the current energy level of the drone, and the system coverage value.
- States: There are many different input information modes to choose from [22]. In our study, the state fed into the algorithm consists of the local observations of all agents, which are combined into a global state representation.
- Actions: The action space of the drones is continuous, and this allows each agent to take actions in any direction and at any distance. The action of each agent is represented as a two-dimensional vector, $(\Delta x, \Delta y)$, which is determined at each time step by the actor network.
- Reward: The algorithm receives states, actions, and outputs rewards. Our study implements reward sharing, which implies that the total reward of all drones is used as the reward for each drone. The specific method for calculating rewards is introduced later.
- Policy π: The policy π determines the actions to be taken by the drones based on the current state, aiming to maximize the cumulative reward over time.

4. Solutions

The objective of our study is to maximize the coverage of users in a post-disaster area while minimizing the energy consumption of UAVs. We consider a set of UAVs operating as aerial base stations that provide communication services. To ensure cooperation among these drones, we present the methodology for using multi-agent proximal policy optimization (MAPPO) [23] to control the movement trajectories of drones for emergency communication services in disaster scenarios. Next, we describe the MAPPO algorithm, the reward function, and the implementation details of our approach.

4.1. Algorithm Structure

MAPPO is an extension of proximal policy optimization (PPO) [22] designed for multi-agent environments. It optimizes the policies of multiple agents (drones in our case) in a centralized manner while enabling decentralized execution. The following are the key components of the MAPPO algorithm:

- Centralized critic: A single critic evaluates the joint actions of all agents, thereby providing a more stable learning process.
- Decentralized actors: Each UAV has its own actor network, making decisions based on local observations.
- Clipped objective: Similar to PPO, MAPPO uses a clipped surrogate objective to ensure stable policy updates, thus preventing large deviations from the current policy. The MAPPO optimization objective is provided by

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t\left[\min\left(r_t(\theta)\hat{A}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t\right)\right], \tag{6}$$

  where $r_t(\theta)$ is the probability ratio between the new and old policies, $\hat{A}_t$ is the advantage estimate, and ϵ is a hyperparameter that controls the clipping range.

Algorithm 1 outlines an MAPPO-based MARL algorithm designed for post-disaster drone–UE communication scenarios. The MAPPO algorithm is implemented using a centralized training approach with decentralized execution. The training process involves simulating multiple episodes, where the UAVs learn to optimize their trajectories through interaction with the environment.

Algorithm 1 MAPPO-based drones trajectory algorithm

Orthogonally initialize actor networks $\pi_u(\theta_u)$ for each drone u.
Initialize a shared critic network $V(\phi)$.
Initialize replay buffer B.
Set hyperparameters: learning rate α, γ, clip range ϵ, batch size, replay buffer size, update interval.
for episode $= 1, 2, \ldots$ **do**
  for $t = 1$ to update interval **do**
    Obtain current states $s_t = \{o_{u,t}\}$ for all drones.
    for each drone u **do**
      Sample action $a_{u,t}$ from $\pi_u(o_{u,t}|\theta_u)$.
    end for
    Execute actions $a_{u,t}$ and observe rewards r_t and next states s_{t+1}.
    Store experiences $(s_t, \{a_{u,t}\}, r_t, s_{t+1})$ in replay buffer B.
    if episode is done **then**
      Reset environment.
    end if
  end for
  if size of $B \geq$ replay_buffer_size **then**
    for each update step **do**
      Sample mini-batch of experiences from B.

Algorithm 1 *Cont.*

Compute advantage estimates and targets using $V(\phi)$.
for each mini-batch **do**
Compute critic loss: $L_V = (targets - V(s))^2$.
Update critic network: $\phi \leftarrow \phi - \alpha \cdot \nabla_\phi L_V$.
end for
for each drone u **do**
Compute ratio: $r_t(\theta_u) = \exp(\log \pi_u(a_{u,t}|s_{u,t}) - \log \pi_u^{old}(a_{u,t}|s_{u,t}))$.
Compute clipped objective: $L^{CLIP} = \min(r_t(\theta_u) \cdot A_t, \text{clip}(r_t(\theta_u), 1-\epsilon, 1+\epsilon) \cdot A_t)$.
Compute actor loss: $L_\pi = -\mathbb{E}[L^{CLIP}]$.
Update actor network: $\theta_u \leftarrow \theta_u - \alpha \cdot \nabla_{\theta_u} L_\pi$.
end for
end for
Clear replay buffer B.
end if
end for

4.2. Training Process

Training deep reinforcement learning models, particularly in a multi-agent setting like MAPPO, can be computationally intensive and time-consuming. To accelerate the training process and improve the efficiency of learning, we employ several techniques:

- Input normalization: Normalization ensures that all features contribute equally to the learning process and prevents issues related to varying scales of input data. By normalizing the input data, we ensure that our MAPPO-based UAV control system operates on a stable and consistent input space, thereby leading to more efficient and effective learning.
- Experience replay: Experience replay helps in breaking the correlation between consecutive training samples, which can lead to more stable learning. In our implementation, we use a shared replay buffer in which all UAVs store their experiences. During training, mini-batches of experiences are randomly sampled from this buffer to update the network weights, which ensures that the UAVs learn from a diverse set of experiences. After each training session, the replay buffer is cleared to collect new information and retrain.
- Parameter sharing: Parameter sharing across UAVs can significantly reduce the number of parameters to be learned and thus enhance the learning process. In our approach, we share the parameters of the actor networks among all UAVs. This not only accelerates training but also ensures that the UAVs learn a coordinated strategy for maximizing coverage and minimizing energy consumption.
- Parallel training: To further speed up training, we utilize parallel training by running multiple simulations concurrently. Each simulation runs on a separate environment instance, which enables the UAVs to collect more experience in less time. The experiences from all parallel simulations are aggregated and used for updating the policy and value networks.

4.3. Reward Function

To meet our goals, we design a reward function that is strongly correlated with the system coverage and energy consumption of the drones. If a UE is covered by drone u in time slot t, and the number of UEs served by the drone does not reach the upper limit, the coverage factor C_n^t is 1—that is,

$$C_n^t = \begin{cases} 1, & SR_n u \leq SR_{th} \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Therefore, the total number of users served in the time slot t is

$$C^t = \sum_{n=1}^{N} C_n^t. \tag{8}$$

According to Equations (4) and (5), the energy consumed by drone u between time slot t can be calculated as

$$E_t = P_v(v_c)\frac{H_{as}^u}{v_c} + P_h(v_h)\frac{d^u}{v_h} + P_h(v_{hov})\Delta t, \tag{9}$$

where H_{as}^u and d^u imply ascending or descending flying distance and horizontal flying distance of drone u.

The reward function that we designed aims to balance the trade-off between maximizing user coverage and minimizing energy consumption. According to Equations (8) and (9), it is expressed as

$$r_t = C_t - \xi \cdot E_t, \tag{10}$$

where C_t is the user coverage at time step t, E_t is the energy consumption of all drones in time slot t, and ξ is the weighting factor.

5. Simulation Results

To evaluate the performance of our MAPPO-based UAV control system, we conduct extensive simulations in a realistic disaster scenario. This section details the simulation setup, the parameters used, the evaluation metrics, and the results obtained from our experiments.

5.1. Simulation Setup

The simulation environment is designed to mimic a typical post-disaster area with the following characteristics:

- Area: A region measuring 3 km × 3 km is used to simulate the disaster area.
- UE distribution: As illustrated in Figure 3, the rescuers are randomly distributed among 10 rescue centers in the area; each group of rescuers is assigned a part of the area and each rescue center is assigned the same number of rescuers to conduct an undifferentiated manual search of the disaster scene.
- Dynamic conditions: The locations of rescuers and communication demands change over time to simulate the dynamic nature of real-world disaster scenarios.

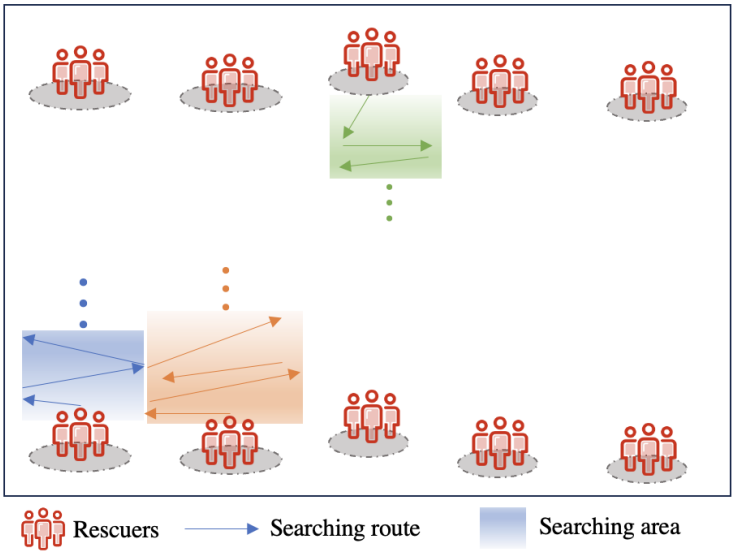


Figure 3. Rescue workers' search route map.

We deploy a different number of drones in the simulation. The drones are modeled with quad-copter dynamics, including constraints on speed (10 m/s), altitude (16 m), and maneuverability. The energy consumption of UAVs is calculated based on their speed, distance traveled, and hovering activities. Each drone–UE connection has a signal strength threshold within which it can provide services to its users. Using path loss exponential modeling, it is revealed that the communication service quality degrades with the distance from the UAV. Additionally, the interference between different drones to UEs must be considered. The key parameters used in the simulations are illustrated in Table 1. To reduce the experimental time, we open 12 parallel environments to simultaneously collect data. After each data collection episode, the algorithm runs 15 training epochs to fully utilize the data. The data size for our small batch training is 512.

Table 1. Numerical parameters.

Parameters	Values
Region	3 km × 3 km
Number of rescue centers	10
Drone hight, H	16 m
Drone number, U	3, 4, 5, 6
Moving range of a drone at one time	[0, 300 m]
Upper limit of served UE number for each drone	40
UE height, h	1.5 m
Number of UEs, N	200
Mean additional loss, η	1 dB
Received signal power, P_t	−3 dBW
Carrier frequency, f_c	1 GHz
SR_{th}^{low}	0 dB
SR_{th}^{high}	5 dB
Weight of drone, W	23.84 Newton
Rotor number, N_R	4
Horizontal flying speed of drone, v_h	10 m/s
Rotor tip speed, v_{tip}	102 m/s
Fuselage area, A_f	0.038 m^2
Drag coefficient, C_D	0.9
Rotor disc area, A_r	0.06 m^2
Profile drag coefficient, Δ	0.002
Rotor solidity	0.05
Hovering time slot Δt	60 s
Weighting factor of reward function ξ	0.1
Total number of episodes	800
Episode length	60 min
Number of parallel envs for training rollouts	12
Number of network layers	2
Dimension of hidden layers	512
Activation function	ReLU
Learning rate	3×10^{-6}
Critic learning rate	5×10^{-4}
Number of PPO epochs	15
PPO clip parameter	0.2
Number of batches	512
Entropy term coefficient	0.01
Discount factor	0.99
GAE λ parameter	0.95

To evaluate the performance of our approach, we use the following metrics:

- Coverage Ratio: The total number of users covered by the UAVs at each time step.
- Energy Efficiency: The proportion of energy consumed by all the UAVs over the simulation period.
- Reward: The cumulative reward obtained, reflecting the balance between coverage and energy consumption.

To the best of our knowledge, there is no benchmark for the related research thus far. We compare our method with three other different deployment methods proposed in [24]. We redesign the details of the drone configuration in these methods according to our specific scenarios. We believe that these methods are more likely to produce intuitive and convincing results compared to ours, both in terms of whether the drone is moving and in the specific styles of the drone's movements. The comparison results are presented in Section 5.5 below. In addition, it is worth mentioning that each comparison method is derived from an average of 100 running results.

5.2. Super-Parameter and Convergence Analysis

We first analyze the impact of the number of hidden layers in the algorithm on training. Considering the complexity of our problem, we analyze the training performance with two fully connected layers and three fully connected layers, respectively. Additionally, the following are the other settings in the experiment: the number of drones is four, and a high signal to inference and noise ratio is selected. As depicted in Figure 4, when the number of hidden layers is three, the network can initially learn and capture more features and complex relationships and, thus, lead to faster reward growth and quicker convergence. However, as the number of training episodes increases, the network with two hidden layers, despite converging slower, achieves better training results in the long run. Therefore, in this study, we set the number of network layers at two for the subsequent experiments.

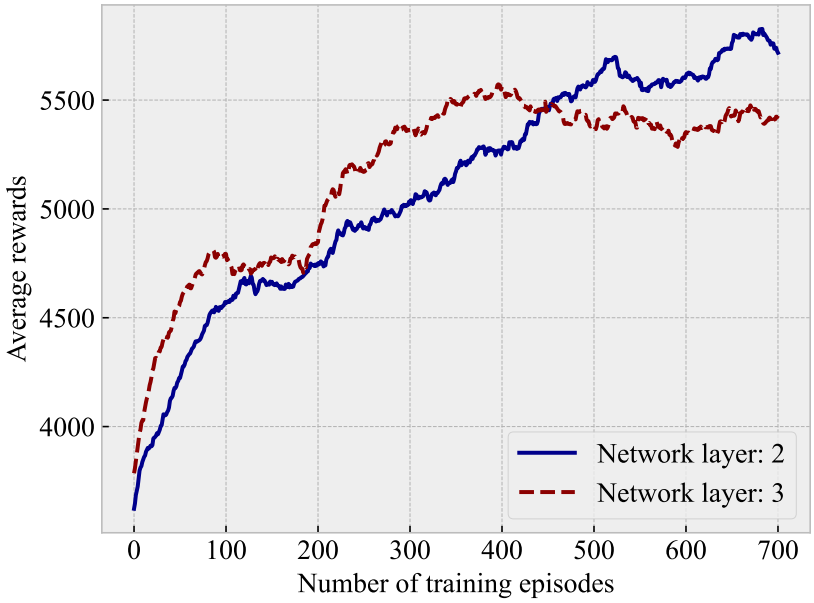
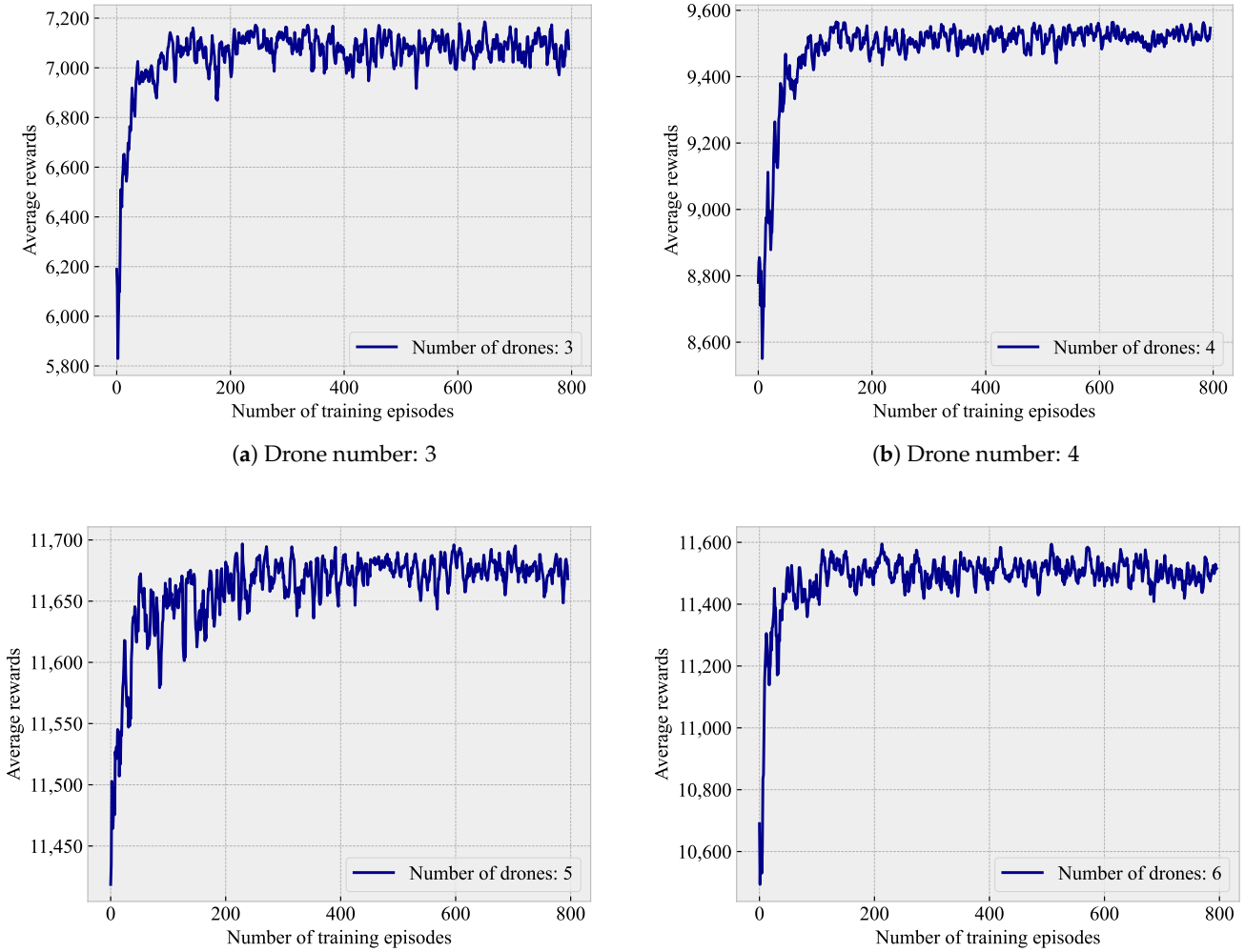


Figure 4. Impact of the number of hidden layers.

Here, we study the convergence performance of the proposed algorithm. The convergence conditions of training for different numbers of UAVs are depicted in Figure 5. It is evident from the figure that, at the beginning of training, the initial cumulative reward is relatively low because the drone has not yet learned the appropriate trajectory in the dynamic environment to cover the UEs. However, over time, the cumulative reward rapidly increases as the drone continues to learn. In addition, due to the nonstationary nature of

the environment, the rewards fluctuate around the average value. However, as training proceeds, the trend of cumulative rewards continues to increase until convergence.



(**a**) Drone number: 3

(**b**) Drone number: 4

(**c**) Drone number: 5

(**d**) Drone number: 6

Figure 5. Convergence conditions over time.

5.3. Analysis of the Communication QoS Threshold SR_{th}

We want to balance the relationship between communication coverage and communication QoS. To this end, we select two signal to interference and noise ratio thresholds, $SR_{th}^{low} = 0$ and $SR_{th}^{high} = 5$. The numerical results of these two thresholds in different environments are depicted in Figure 6. As the number of drones increases, the number of UEs that can meet the high SR threshold decreases. When the number of drones reaches five or six, the number of UEs that meet the high QoS service reduces to zero. However, for low SR threshold requirements, each drone can serve the maximum number of users it is capable of serving. In the scenarios tested with different numbers of drones, all the drones are able to meet the low QoS threshold requirements of the UEs. Therefore, in the subsequent experiments, we select a low SR threshold parameter by default.

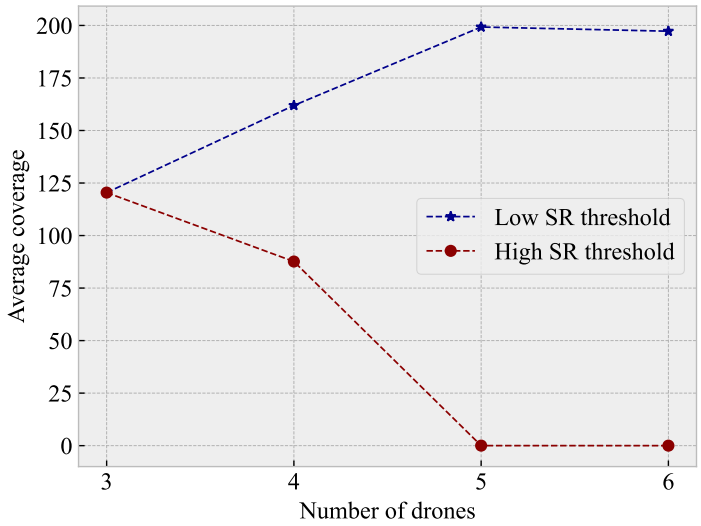
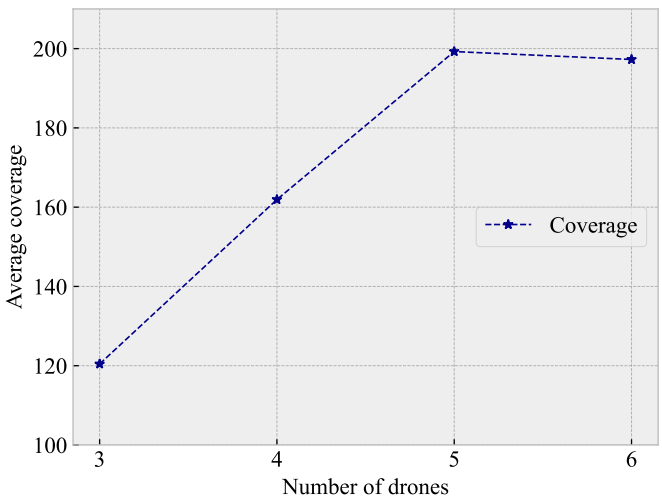


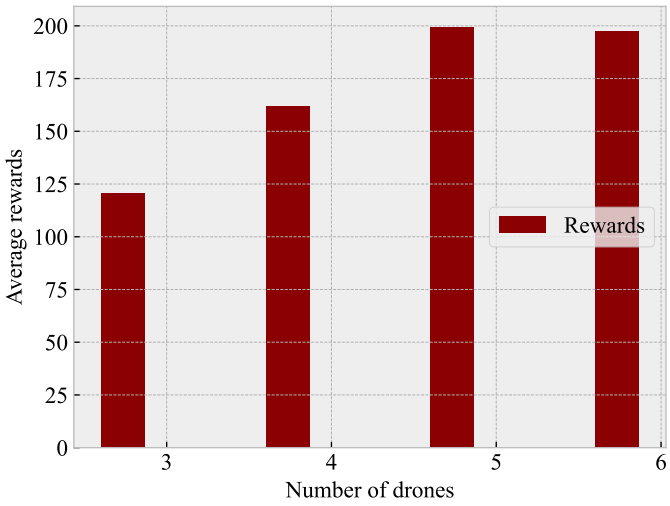
Figure 6. Impact of signal to interference and noise ratio thresholds.

5.4. Analysis of Different Drone Numbers

We test the performance of different numbers of drones in dynamic scenarios. As evident from Figure 7a, when the number of drones reaches five, the drone network can achieve full coverage of the UEs in the scene. However, when the number of drones is six, the system coverage is reduced due to communication interference between the drones, thereby resulting in a corresponding reduction in reward, as depicted in Figure 7b. It is also worth noting that, as the number of drones increases, the average energy consumption of the entire scene rises sharply. Figure 7c takes the energy consumption of the system when the number of drones is three as the benchmark and compares it to the energy consumption when testing other numbers of drones. Obviously, when the number of drones doubles to six, the energy consumed by the drone system also doubles. In conclusion, the number of drones that achieve the maximum cost-effectiveness for our system is five.

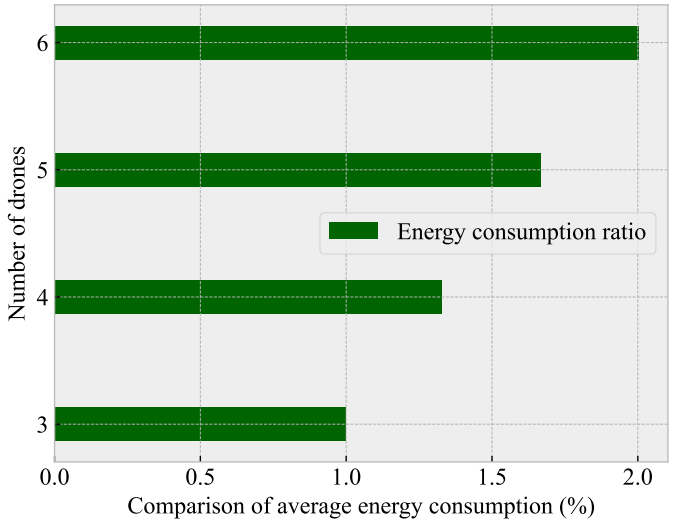


(**a**) Average coverage of drones



(**b**) Average rewards of drones

Figure 7. *Cont.*



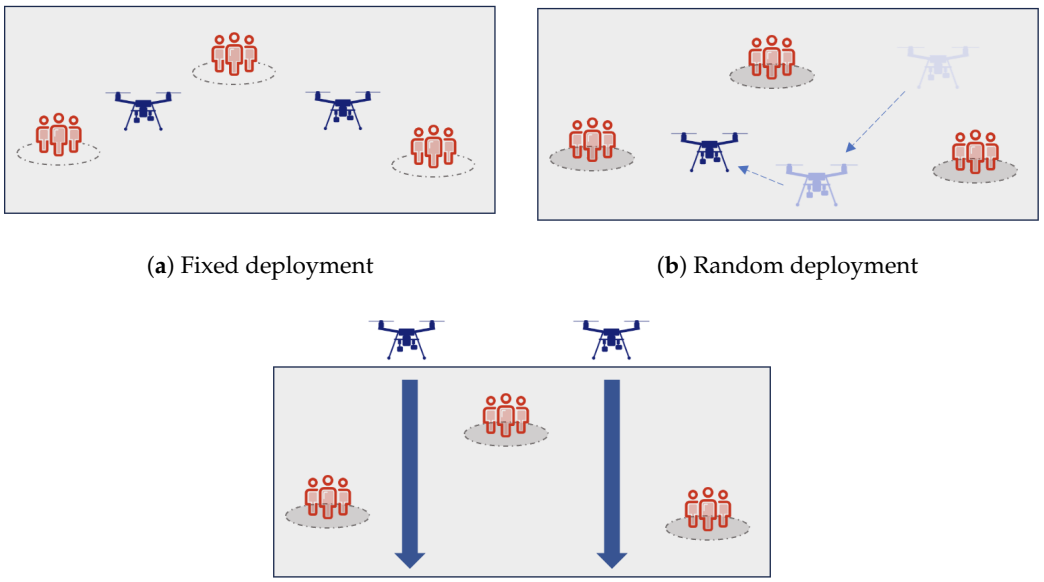
(**c**) Energy consumption comparison of drones

Figure 7. Performance of different drone numbers.

5.5. Performance Comparisons with Other Methods

General planning or optimization methods are typically designed to address static problems and are not inherently suited for dynamic scenarios. Our problem, however, is dynamic and represents a multi-agent Markov decision process (MDP) that requires continuous real-time decision-making. This complexity is further compounded by the extensive state and action spaces involved. The traditional optimization algorithms generally fail to manage this complexity effectively. Consequently, we cannot demonstrate the efficacy of our method by comparing it with traditional optimization algorithms. To evaluate the advantage of drone mobility in communication and the effectiveness of our mobility method, we conduct comparisons with three other drone configuration methods: (1) suspended in a fixed position (see Figure 8a)—drones hover at a fixed distance in the middle of the area; (2) move randomly (see Figure 8b)—the drones move randomly within the area, with the maximum moving distance in each movement interval not exceeding 300 m; and (3) move at a constant speed (see Figure 8c)—throughout the entire period, the drone follows the same search route as the rescuers, moving at a constant speed from one side of the area to the other. In addition, in all these comparative experiments, the disaster scenarios used are identical to the environment used in our method.

As evident in Figure 9, with an increase in the number of drones providing services, the average coverage of all the methods increases until the number of drones reaches five. When the number of drones in the system is six, the number of serviced UEs decreases due to communication interference between the drones. However, it is evident that the proposed algorithm consistently outperforms the other methods in each case. In particular, when the number of drones is four, our method achieves approximately 45% higher coverage of UEs than the constant deployment method. In general, our method achieves superior performance, with an average of 10–45% higher coverage than the other methods.



(**a**) Fixed deployment (**b**) Random deployment

(**c**) Moving with a constant velocity

Figure 8. Different drone deployment methods.

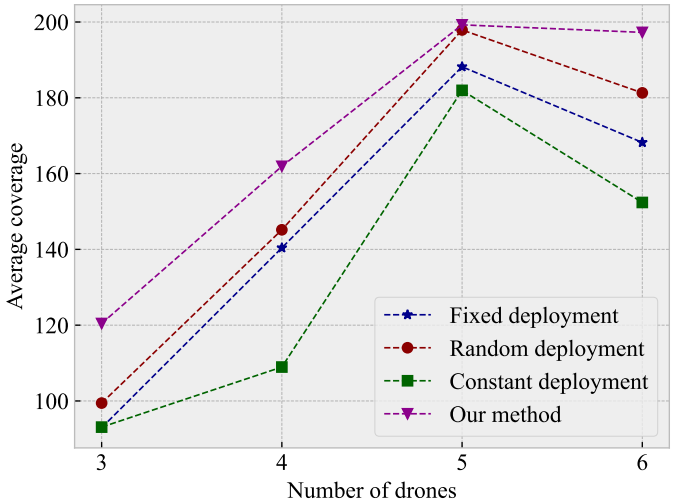
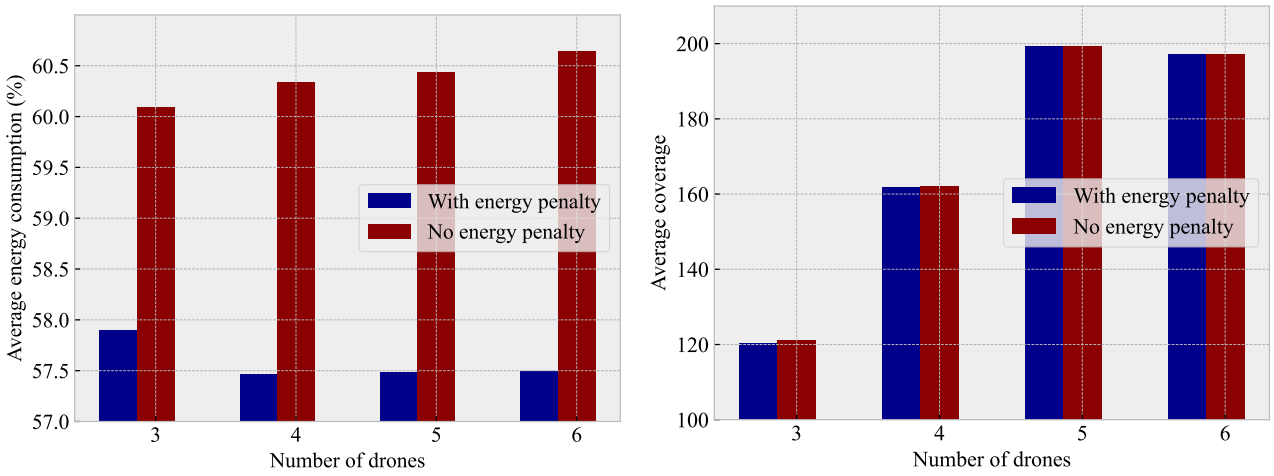


Figure 9. Performance analysis of different methods.

5.6. Reward Function Analysis

Finally, we analyze our designed reward function. As indicated in Equation (10), the reward function consists of two parts: system coverage and a drone energy consumption penalty. To demonstrate the impact of the energy penalty term on the system, we simulate the reward function both with and without the energy consumption penalty term. The simulation results are shown in Figure 10.

The simulation results highlight the effectiveness of our MAPPO-based UAV control system in providing robust and energy-efficient communication services in disaster scenarios. Obviously, although Figure 10b reveals that the coverage performance obtained using the two reward functions is comparable, Figure 10a indicates that the network trained using the reward function with an energy consumption penalty can save approximately 3% energy. Thus, significant improvements in energy efficiency demonstrate the potential of our approach to improve disaster response efforts.



(**a**) Average energy consumption of drones (**b**) System coverage

Figure 10. Impact of energy consumption penalty.

6. Conclusions

In this study, we proposed a DRL-based approach to optimize the trajectories of UAVs in disaster scenarios to provide efficient and reliable emergency communication services. Our primary objective was to maximize the user coverage while minimizing the energy consumption of the drones. The MAPPO algorithm demonstrated robust performance in serving the rescuers in complex and dynamic post-disaster areas. Our extensive simulations validated the efficacy of the reward function designed and the MAPPO algorithm. The proposed method consistently outperformed other deployment strategies in terms of user coverage and energy efficiency. Specifically, without considering the convergence speed of the algorithm, our approach achieved an average of 10–45% higher coverage compared to the fixed-, random-, and constant-velocity deployment methods. Moreover, the consideration of an energy consumption penalty in the reward function significantly improved the energy efficiency, saving approximately 3% more energy while maintaining comparable coverage performance.

The results highlight the potential of using DRL for UAV trajectory optimization in emergency communication networks. Using the adaptive learning capabilities of the MAPPO algorithm, UAVs can dynamically adjust their positions and strategies to meet the changing demands of the environment and UE distribution, thus enhancing the resilience and effectiveness of disaster response efforts. Future research directions could include exploring the integration of more advanced artificial intelligence techniques to further improve the decision-making capabilities of UAVs. Furthermore, investigating the impact of different types of environmental uncertainties, such as varying weather conditions and unpredictable obstacles, and user mobility patterns on the performance of the proposed algorithm could provide deeper insights into optimizing UAV-assisted communication networks. In addition, real-world field tests are necessary to verify the practical applicability and scalability of the proposed solutions in actual disaster scenarios. While the study demonstrates the potential of DRL for UAV trajectory optimization, it does have some limitations. For example, the simulation environment may not fully capture the complexities of real-world scenarios. Additionally, the algorithm performance could be affected by factors such as computational constraints and the need for real-time decision-making. Future research should focus on addressing these limitations by (1) enhancing the simulation environment to include more realistic scenarios; (2) investigating the algorithm's performance under computational constraints; (3) developing methods for real-time decision-making; and (4) conducting extensive real-world field tests to validate the proposed solutions.

By addressing these limitations, future studies can further refine and improve the practical applicability of DRL in UAV-assisted emergency communication networks.

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

Funding: This work was partially supported by JSPS KAKENHI Grant Number 24K14913; Research Grant from the Support Center for Advanced Telecommunications Technology Research, Japan.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

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

References

1. Gu, X.; Zhang, G. A survey on UAV-assisted wireless communications: Recent advances and future trends. *Comput. Commun.* **2023**, *208, 44–78*. [CrossRef]
2. Frattolillo, F.; Brunori, D.; Locchi, L. Scalable and cooperative deep reinforcement learning approaches for multi-UAV systems: A systematic review. *Drones* **2023**, *7*, 236. [CrossRef]
3. Bai, Y.; Zhao, H.; Zhang, X.; Chang, Z.; Jäntti, R.; Yang, K. Towards autonomous multi-UAV wireless network: A survey of reinforcement learning-based approaches. *IEEE Commun. Surv. Tutorials* **2023**, *25*, 3038–3067. [CrossRef]
4. Chittoor, P.K.; Bharatiraja, C. Solar Integrated Wireless Drone Charging System for Smart City Applications. In Proceedings of the 2021 IEEE 6th International Conference on Computing, Communication and Automation (ICCCA), Arad, Romania, 17–19 December 2021; pp. 407–412. [CrossRef]
5. Ullah, Z.; Al-Turjman, F.; Mostarda, L. Cognition in UAV-aided 5G and beyond communications: A survey. *IEEE Trans. Cogn. Commun. Netw.* **2020**, *6*, 872–891. [CrossRef]
6. Sobouti, M.J.; Mohajerzadeh, A.H.; Seno, S.A.H.; Yanikomeroglu, H. Managing sets of flying base stations using energy efficient 3D trajectory planning in cellular networks. *IEEE Sens. J.* **2023**, *23*, 10983–10997. [CrossRef]
7. Landers, M.; Doryab, A. Deep reinforcement learning verification: A survey. *ACM Comput. Surv.* **2023**, *55*, 1–31. [CrossRef]
8. Javaid, S.; Saeed, N.; Qadir, Z.; Fahim, H.; He, B.; Song, H.; Bilal, M. Communication and control in collaborative UAVs: Recent advances and future trends. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5719–5739. [CrossRef]
9. Sharvari, N.; Das, D.; Bapat, J.; Das, D. Connectivity and collision constrained opportunistic routing for emergency communication using UAV. *Comput. Netw.* **2023**, *220*, 109468. [CrossRef]
10. Zhang, T.; Chen, C.; Xu, Y.; Loo, J.; Xu, W. Joint task scheduling and multi-UAV deployment for aerial computing in emergency communication networks. *Sci. China Inf. Sci.* **2023**, *66*, 192303. [CrossRef]
11. Na, Y.; Li, Y.; Chen, D.; Yao, Y.; Li, T.; Liu, H.; Wang, K. Optimal energy consumption path planning for unmanned aerial vehicles based on improved particle swarm optimization. *Sustainability* **2023**, *15*, 12101. [CrossRef]
12. Pan, H.; Liu, Y.; Sun, G.; Fan, J.; Liang, S.; Yuen, C. Joint power and 3D trajectory optimization for UAV-enabled wireless powered communication networks with obstacles. *IEEE Trans. Commun.* **2023**, *71*, 2364–2380. [CrossRef]
13. Zhang, C.; Zhou, W.; Qin, W.; Tang, W. A novel UAV path planning approach: Heuristic crossing search and rescue optimization algorithm. *Expert Syst. Appl.* **2023**, *215*, 119243. [CrossRef]
14. Li, B.; Yang, R.; Liu, L.; Wang, J.; Zhang, N.; Dong, M. Robust computation offloading and trajectory optimization for multi-UAV-assisted mec: A multi-agent DRL approach. *IEEE Internet Things J.* **2023**, *11*, 4775–4786. [CrossRef]
15. Sun, J.; Sheng, Z.; Nasir, A.A.; Huang, Z.; Yu, H.; Fang, Y. Energy efficiency maximization for WPT-enabled UAV-assisted emergency communication with user mobility. *Phys. Commun.* **2023**, *61*, 102200. [CrossRef]
16. 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. [CrossRef]
17. Donevski, I.; Virgili, M.; Babu, N.; Nielsen, J.J.; Forsyth, A.J.; Papadias, C.B.; Popovski, P. Sustainable wireless services with UAV swarms tailored to renewable energy sources. *IEEE Trans. Smart Grid* **2022**, *14*, 3296–3308. [CrossRef]
18. Abeywickrama, H.V.; Jayawickrama, B.A.; He, Y.; Dutkiewicz, E. Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance. *IEEE Access* **2018**, *6*, 58383–58394. [CrossRef]
19. Alzenad, M.; El-Keyi, A.; Lagum, F.; Yanikomeroglu, H. 3D placement of an unmanned aerial vehicle base station (UAV-BS) for energy-efficient maximal coverage. *IEEE Wirel. Commun. Lett.* **2017**, *6*, 434–437. [CrossRef]
20. Khawaja, W.; Guvenc, I.; Matolak, D.W.; Fiebig, U.C.; Schneckenburger, N. A survey of air-to-ground propagation channel modeling for unmanned aerial vehicles. *IEEE Commun. Surv. Tutorials* **2019**, *21*, 2361–2391. [CrossRef]
21. Sutton, R.; Barto, A. Reinforcement learning: An introduction. *IEEE Trans. Neural Netw.* **1998**, *9*, 1054. [CrossRef]

22. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
23. Yu, C.; Velu, A.; Vinitsky, E.; Wang, Y.; Bayen, A.; Wu, Y. The surprising effectiveness of PPO in cooperative, multi-agent games. *arXiv* **2021**, arXiv:2103.01955.
24. 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**, *20*, 2835–2847. [CrossRef]