A Multi-Objective Reinforcement Learning Framework for UAV Path Planning Automation in Complex Fire and Rescue Environments

Yanming Chen*, and Saeid Pourroostaei Ardakani†
*School of Computer Science, University of Nottingham, Ningbo, China
†School of Engineering and Physical Sciences, University of Lincoln, UK

***Abstract*—Fire and rescue UAVs employ AI methods to automate navigation over fire zones to detect and extinguish fire flames and/or rescue fire-trapped lives through low-cost and collision-free flight routes. This paper introduces a reinforcement learning framework that allows UAVs to detect and prioritise fire-trapped targets based on their risk levels. It establishes three simultaneous objectives including maximised power conservation, rescue success rate, and flight safety to plan fire and rescue operations in complex environments where field size, the number of trapped targets, and fire source count vary. An extensive empirical evaluation is conducted to test and evaluate the performance of the proposal against three well-known benchmarks including Double Deep Q-network, Advantage Actor-Critic, and Genetic Algorithm. The results demonstrate that the proposed solution outperforms the benchmarks in most circumstances especially when the fire and rescue environment is large and complex.**

***Index Terms*—Reinforcement Learning, Q-learning, Deep Learning, Fire and Rescue, UAV Path Planning.**

I. Introduction

FIRES lead to serious environmental and health consequences, including soil degradation, infrastructure damage, and pollution, if they are not quickly and properly responded. However, firefighting missions are delayed or left uncompleted due to environment variations, obstacles (e.g., buildings or trees), inaccurate fire alarms/data, and the lack of proper firefighting equipment especially in wide and complex areas such as rainforests or metropolitan [1]. To avoid this, professional Fire and Rescue (FAR) teams are enabled with advanced technologies such as the Internet of Things, Robotics, Intelligent Cameras, and Wearable Sensors to efficiently plan fire suppression, manage first aid, and rescue fire-trapped lives.

Unmanned Aerial Vehicle (UAV) technology is widely used in FAR missions due to its remote-sensing/communication and manoeuvring capacities [2], [3]. It has the capacity to offer benefits such as environment monitoring and surveillance and search and rescue in hard-to-access and/or unreachable environments. UAVs are usually equipped with restricted power sources (i.e., battery) and navigated via GPS data or radio remote control devices [4], [5]. They use onboard sensors and/or cameras (e.g., thermal, multi-spectral, or hyper-spectral) to capture environmental events (i.e., fires and smoke), embedded computing units to process data, and wireless transceivers to communicate with remote firefighting stations/teams [1], [6], [7].

Artificial Intelligence (AI) methods such as Reinforcement Learning (RL) [8], and Genetic Algorithm (GA) [9] enable UAVs to automate their trajectory planning. RL is a self-learning AI methodology that explores the environment through trial and error methods. RL techniques such as Q-learning and A2C are widely used in UAV route planning to automate UAV jobs. They allow UAVs to interact with the environment and take the best-fitted actions to achieve mission objectives (i.e., maximise a reward function) [10]. Q-learning [11] is a value-based RL algorithm that enables UAVs to plan optimal flight routes (e.g., shortest, and/or collision-free) by exploring the environment and utilising rewards obtained from detecting obstacles (e.g., trees or buildings) and/or completing tasks (e.g., rescuing trapped targets), while A2C [12] aims to find the best-fitting path planning strategies according to the environmental feedback received.

A. Motivation

AI methods (i.e., reinforcement learning) have the capacity to enable UAVs to automatically plan environment exploration and FAR flights [13]. Yet, there is a trade-off triangle of safety (i.e., minimised obstacle collisions and flight failures), resource conservation (e.g., minimised power consumption), and coverage (e.g., maximised rescue rate) that should be taken into account to avoid failure of AI-driven UAV path planning in FAR operations due to battery run out, obstacle collision (e.g., trees), and/or communication signal loss [14], [15]. This trade-off triangle drives our research to develop an AI-powered path-planning solution for FAR UAVs that addresses the following objectives:

1) Minimised flight cost: the UAV should find/rescue the fire-trapped targets with minimised battery consumption.
2) Maximised flight safety: the UAV needs to plan collision-free routes to avoid mission/flight failures in crowded and dense areas.
3) Maximised rescue rate: the UAV should be able to detect, prioritise and rescue fire-trapped targets (e.g., animals) according to their risk levels (e.g., proximity to the fire).

B. Contribution

This research aims to propose a reinforcement learning solution through which UAVs automatically plan the best-fitted

FAR routes (i.e., shortest and collision-free) to find/rescue the maximum number of trapped individuals/animals in complex environments. The key contributions of this research are outlined as follows:

- Propose a novel reinforcement learning framework that enables FAR UAVs to dynamically adjust their routes in response to the risk levels of fire-trapped targets within complex fire environments where the number of targets, fire sources, and field dimensions vary.
- Develop a Q-learning route planning solution that simultaneously targets multiple FAR objectives -mainly maximising flight energy efficiency, minimising the risk of fire/obstacle collisions, and maximising rescue success rates, and manages the inherent trade-offs among them in complex fire environments.
- Conduct a rigorous approach to experiment design, and an extensive empirical analysis to evaluate the performance of the proposed solution against three well-known AI benchmarks including Double Deep Q-network (DDQN) [16], Advantage Actor-Critic (A2C) [17], and Genetic Algorithm (GA) [18].

The remainder of this paper is organised as follows. Section II reviews AI-driven UAV path planning methods in environment surveying and search and rescue applications. Section III introduces the research methodology and the proposed solution, while Section IV presents the simulation and experimental setup, discusses and reports the research findings, and evaluates the performance of the proposed approach against the benchmarks to highlight the key benefits of the proposed solution in complex FAR applications. Section V summarises the key findings of this research and addresses future works.

II. Literature Review

This section introduces AI methods -mainly Q-learning, Deep RL, and A2C for UAV route planning used in environment monitoring and surveillance applications. This is not a statistical survey but rather an overview of promising methods that have the potential to enhance UAV flight automation and efficiency in FAR scenarios.

Traditional AI methods such as Grey Wolf Optimiser (GWO), Evolutionary Algorithm (EA), Particle Swarm Optimisation (PSO), and GA are widely used to automate UAV path planning and performance optimisation -mainly flight cost reduction (i.e., power consumption) [19], [20]. For example, [21] utilises Simplified GWO and Modified Symbiotic Organisms Search (HSGWO-MSOS) methods to enable UAVs to heuristically explore a given map and build an optimal flight path to fly through and reach the environmental targets. The GWO algorithm initiates the flights in line with an environment map, while Symbiotic Organisms Search is a meta-heuristic algorithm that aims to find the global optimal flight solution. The performance of this approach is tested via simulation in a 1000×1000 area. According to the results, HSGWO-MSOS outperforms heuristic map exploration algorithms in terms of flight distance. [22] uses an EA method to model UAV path planning for search and rescue in 3D simulated disaster scenarios. For this, the itinerary planning is modelled and optimised using EA for which the fitness function gives the flight length according to the UAV's height, angle, and slope constraints. [23] uses the Partially Observable Markov Decision Processes technique (POMDPs) to route UAVs for (multiple) object tracking. It uses UAV sensory data, target state, and the action taken to find an optimal route through which a cumulative cost (i.e., power) is minimised. The traditional AI solutions offer several benefits such as flight cost reduction, however, they fail to conduct multi-objective optimisation for UAV path planning missions, and still suffer from the lack of generalisation. This stems from the fact that these solutions aim to optimise a single objective function (i.e., flight length) without taking additional objectives (i.e., mission success rate) or external factors (i.e., collisions or mission priorities) into account during the UAV trajectory planning.

GA is an evolutionary algorithm, inspired by natural selection, aiming to yield solutions for consecutive generations by combining the principles of mutation (introduces diversity through random genes), selection (picking up candidates based on their fitness scores), and crossover (merges parental genetic information to create new ones). [24] takes the benefits of GA and PSO methods to minimise UAV path length and power consumption. They claim that the approach has the capacity to build the shortest route for 70% of the flight scenarios in a 3D simulation environment. However, it fails to meet the real-time requirements of real-world applications due to the computation overhead of GA and PSO model training. [25] discusses the performance of GA solutions for UAV path planning and shows how this technique fails to meet real-time path planning due to slow route re-calibrating. It also highlights the poor performance of the GA algorithm for obstacle avoidance especially where obstacles are random. [26] uses risk analysis of the flight routes and takes the best-fitted strategies where multiple landing points with different conditions are available to improve the performance of GA path planning methods. As it reports, this approach reduces the length of flight routes as compared to conventional GAs. However, GA-enabled methods fail to meet the requirements of real-world applications due to frequent collisions and mission failures especially where the environment is crowded and complex.

A2C is a reinforcement learning approach that combines the benefits of both value-based and policy-based methods [27]. For this, the Actor is responsible for determining the best action given a state, while the Critic evaluates the action taken by the Actor using the value function estimation. [28] proposes the Simultaneous Target Assignment and Path Planning (STAPP) solution for collaborative UAV path planning in dynamic environments. It builds a Multi-Agent Deep Deterministic Policy Gradient (MADDPG) framework in which A2C is used to train UAVs and detect targets through collision-free routes. The A2C method works as a deterministic policy gradient approach over continuous action spaces to build a UAV swarm with continuous space optimisations. However, A2C methods suffer from high training variance, and potential for increased computational demands in real-time and time-sensitive applications [29], [30], [31].

Q-learning [32] is a model-free and value-based RL algo-

rithm that learns an optimal policy mapping from states to actions to maximise the expected cumulative reward. It aims to balance environment exploration and exploitation using a ϵ-greedy strategy that allows the Q-learning agent (i.e., the UAV) to either select the highest Q-value action with a probability of $1-\epsilon$ or take random actions with the probability of ϵ per each environment exploration epoch. The value of ϵ is calculated using Equation 1.

$$\epsilon = \max(\epsilon_{\min}, \epsilon \times \epsilon_{\text{decay}}) \quad (1)$$

Q-learning uses a Q-table to calculate the expected rewards based on the given actions. Q-table refers to the list of available actions for environment status/state at a certain timestamp. As Table I shows, each row is the expected cumulative reward value (or Q-value) for an action A_i in line with the environment state of S_i. Q-value is computed using Equation 2, where $\alpha \in$ (0,1) is the learning rate, r is the reward at time t, and $\gamma \in$ (0,1) is the discount factor for reward value convergence.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)] \quad (2)$$

TABLE I
Q-TABLE

State	$Action_1$	$Action_2$	...	$Action_n$
S_1	Q(S_1, A_1)	Q(S_1, A_2)	...	Q(S_1, A_n)
S_2	Q(S_2, A_1)	Q(S_2, A_2)	...	Q(S_2, A_n)
...	...	...	...	...
S_n	Q(S_n, A_1)	Q(S_n, A_2)	...	Q(S_n, A_n)

Q-learning is commonly used for UAV itinerary planning. [33] proposes a location and energy-aware Q-learning technique for UAV itinerary planning in smart farming, while [34] tunes a Snake Game method with deep Q-learning to propose an autonomous environment exploration for UAVs. [35] gives an adaptive federated Q-learning approach by having an epsilon-greedy RL policy to prevent UAV flights in jamming areas. The epsilon-greedy policy balances Q-learning exploration and exploitation using a random section method. The results show a 39.9% improvement as compared to traditional Q-learning solutions. Adaptive and Random Exploration (ARE) is a UAV itinerary planning solution that uses Q-learning to automate path planning [36]. It consists of three modules: 1) a learning module that stores and updates Q-values using a Q-learning neural network, and 2) a trap-escape module that builds a tree infrastructure, called Rapidly Random Tree (RRT), to avoid loops. The RRT measures the cost of each route and uses a threshold cost to detect and discard expensive routes, and 3) an action module that allows the UAV to take the best-fitted actions based on the output of the other two modules. Simulation results demonstrate that ARE can bypass at least three consecutive obstacles and successfully reach the destination. [37] proposes an optimised Q-learning algorithm for UAV path planning that improves the action-selection strategy and Q-function initialisation. It measures the Euclidean distance between the flying UAV and the destination and combines the ϵ-greedy and Boltzmann Q-learning exploration strategies to realise a faster exploration in unknown areas.

Deep Reinforcement Learning (DRL) tunes RL with deep learning techniques (e.g., Convolutional Neural Network [38]) to expand the state and action spaces to enhance generalisation [39] and [40]. However, DRL's learning rate is reduced if the approximation error and the neural network divergence are both increased. To avoid this, [38] introduces an approach that stabilises the learning process using the experience replay memory technique which stores experience tuples $(\text{state}, \text{action}, \text{reward}, \text{next state})$ per each environment discovery round/epoch (see TableII). It trains the learning agent (i.e., UAV) by uniform sampled batches which are collected from the replay memory, and results in minimised consecutive sample dependencies. [25] proposes a distributed control framework, e-Divert, for vehicle navigations (i.e., UAVs). It builds a DRL multi-agent approach using the Convolutional Neural Network (CNN) technique to extract spatial information and feed an A2C network for real-time decision-making. It also employs the Long Short-Term Memory (LSTM) technique to enable N-step temporal sequence modelling. Deep ESN Architecture [41] combines the new input of weight matrix W_{in}, current matrix W, and the output weight matrix W_{out} to build a deep learning model to plan UAV flights, while [42] utilises a novel environment observation technique that employs compressed global and uncompressed local maps to feed the DRL-enabled UAV trajectory planning. [43] introduces an improved deep learning policy that changes the probability distribution of learning experiences to enable the model to obtain further experiences at each learning stage. It uses CNN to build the Q-network in an 8×8 grid with a single target. However, DRL-enabled UAV path planning methods suffer from increased computational demands and slower convergence rates in complex environments [44].

TABLE II
REPLAY MEMORY

State	**Action**	**Reward**	**Next State**	**Done**
s_1	a_1	r_1	s_2	False
s_2	a_2	r_2	s_3	False
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
s_n	a_n	r_n	s_{n+1}	True

This literature review discloses existing gaps in AI-driven UAV route planning methods such as GA, A2C, Q-learning, and Deep RL. Traditional path planning methods (i.e., GA) are predominantly optimised for single-objective tasks in FAR applications. They are unable to intelligently learn the environment and typically rely on predefined map information to find a safe and obstacle-free route from a start point to the target in restricted environments. Yet, neural network-based approaches (i.e., Deep RL) demand extensive training to understand complex environments that lead to slow convergence and increased UAV power consumption in FAR applications. These limitations reduce the practical viability of the established AI-driven route planning methods in real-world applications especially where the fire field is large and

complex. Hence, there is a need for a novel path-planning solution that resolves the existing drawbacks and meets the key objectives, including maximised power conservation, rescue rate, and flight safety in large and complex FAR applications.

III. Methodology

This section presents the research methodology and proposes an RL-enabled UAV itinerary planning solution to enhance simultaneous objectives including energy conservation, flight safety, and rescue rates in complex FAR scenarios. For this, the system model is introduced and formulated, FAR path planning objectives are defined and explained, and the RL approach is described and discussed.

A. The System Model

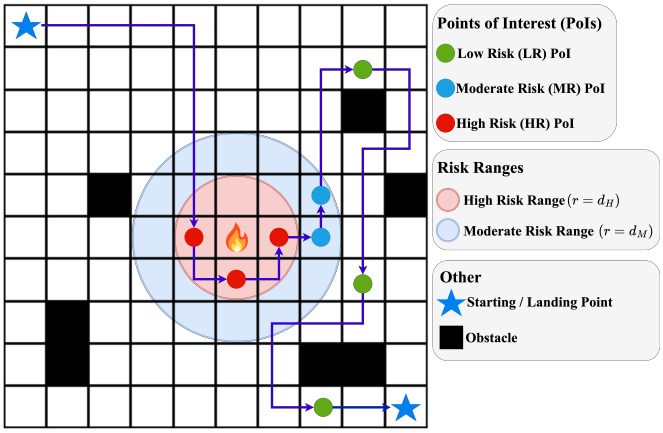

Fig. 1. FAR environment model.

In this research, a single UAV is forwarded through minimum energy-consuming and collision-free paths to find and rescue fire-trapped targets which are randomly distributed in a 2D ($N \times N \in \mathbb{M}^2$) grid. As Figure 1 depicts, the UAV is not allowed to fly out of the gird and should avoid collisions with ambient obstacles ($O \triangleq \{o = 1, 2, \ldots, O\}$), mainly trees, hills, and/or fire sparks. A location is marked as a Point of Interest (PoI) if it needs to be visited/monitored by the UAV according to the FAR strategies (i.e., fire risks and/or rescue policies). PoIs are classified/prioritised according to their proximity or level of exposure to fire risks as follows:

- High Risk (HR): they are very close to the fire flames and need immediate rescue interventions with the highest priorities.
- Moderate Risk (MR): they are not at immediate risk but will be if the fire spreads.
- Low Risk (LR): they are safe as they are located far from the fire flames.

Equation 3 represents the priority state of PoI_i at time slot t. It calculates the euclidean distance d_{F_i} between PoI (x_i, y_i) and the nearest fire source (x_F, y_F), then compares this distance with the thresholds d_M and d_H to determine the appropriate fire safety priority level.

$$\text{State}_{i,t} = \begin{cases} HR & \text{if } d_{F_i} \leq d_H \\ MR & \text{if } d_H < d_{F_i} \leq d_M \\ LR & \text{if } d_{F_i} > d_M \end{cases} \tag{3}$$

B. FAR Objectives: Definition and Formulation

This research defines three key simultaneous objectives including maximising energy conservation, rescue rate, and flight safety (collision avoidance). By this, the path planning method should enable the UAV to 1) visit/rescue a maximum number of PoIs according to their priorities, 2) avoid heuristic/blind paths (i.e., loops) to minimise power consumption and reduce flight time, and 3) detect and mark obstacles/barriers to minimise collisions and mission failures. Equation 4 formulates the FAR objectives, where UAV energy conservation $En_T(\pi)$, rescue rate $Re_T(\pi)$, and flight safety $Sf_T(\pi)$ are assessed in the context of flight policy π.

$$\text{FAR} : \max_{\pi}(Sf_T(\pi), Re_T(\pi), En_T(\pi)) \tag{4}$$

Equation 5 calculates the flight rescue rate $Re_T(\pi)$ based on the total number of successful PoI visits/rescues and received rescue rewards during the FAR mission. Function ϕ refers to the rescue reward of PoI_i under flight policy π at time slot t. Yet, the reward function will be discussed later under section III-C.

$$Re_T(\pi) = \sum_{t=1}^{T} \phi(\pi, PoI_i, t) \tag{5}$$

UAV energy conservation $En(\pi)$ refers to the total remaining UAV energy/battery at the end of the FAR mission at time T. According to Equation 6, the consumed power $e(\pi, t)$ during the FAR operation is measured and subsequently subtracted from the initial power CE to determine the remaining power. The consumed power is measured using Equation 7, with E_{base} denoting the energy required for each flight step, while α and β represent the weight parameters for the energy expended during PoI visits and obstacle/fire collisions (OFC), respectively.

$$En_T(\pi) = CE - \sum_{t=1}^{T} e(\pi, t) \tag{6}$$

$$e(\pi, t) = E_{\text{base}} + \alpha_i(PoI_i) + \beta_j OFC_j \tag{7}$$

Flight safety is defined as the ratio of collision-free flight steps to the total number of steps taken. It is calculated using Equation 8, with $c(\pi, t)$ denoting a binary indicator that equals 1 if a collision occurs at time step t under the flight policy π, and 0 if no collision occurs, while L refers to the total flight steps taken during the FAR mission.

$$Sf_T(\pi) = \frac{L - \sum_{t=1}^{T} c(\pi, t)}{L} \tag{8}$$

C. The Proposed Approach

The proposed approach is a Q-learning path planning method to automatically route a UAV during a FAR mission in line with three key concurrent objectives including maximised rescue rate, UAV power conservation, and flight safety. It employs an iterative method to continuously observe the environment and update the action space to maximise the total reward received according to the FAR objectives. A Markov Decision Process (MDP) [45] is used to build the state space, action space, and reward function as follows:

1) State Space ($S \triangleq \{s_t\}$) is defined as Equation 9, where $E(t)$ gives the environment map information (i.e., the location of obstacles and starting/landing zones), $PL(t)$ refers to the PoI locations, and $UL(t)$ reports the UAV's live location at time slot t.

$$s(t) = (E(t), PL(t), UL(t)) \tag{9}$$

2) Action Space A refers to the actions taken by the UAV (e.g., North, East, South, West, and Land) during a FAR mission.
3) Reward function R is represented by Equation 10, where $r_{\text{end}}(t)$ is the reward of successful return to the landing point, while $r_{\text{poi}}(t)$, $r_{\text{pow}}(t)$, and $r_{\text{col}}(t)$ are objective rewards including maximising the success rate (number of visited PoIs), power conservation, and flight safety (minimising the collisions).

$$R \triangleq \{r_t\} = \{r_{\text{end}}(t), r_{\text{poi}}(t), r_{\text{col}}(t), r_{\text{pow}}(t)\} \tag{10}$$

- Return reward r_{end} is a positive reward given if the UAV successfully reaches the landing point (i.e., Base station) at the end of the FAR mission. Equation 11 calculates r_{end}, where P is the total number of PoIs and P_v represents the number of missing PoIs.

$$r_{\text{end}}(t) = 2 \times (P - P_v) \tag{11}$$

- Collision reward $r_{\text{col}}(t)$ is a penalty used to avoid mission failures due to fire/obstacle collisions. It is measured using Equation 12, where M refers to the map dimensions and G is a constant value normalised by the environment size (e.g., here is 4) to control the penalty degree.

$$r_{\text{col}}(t) = -\frac{G}{M * M} \tag{12}$$

- PoI reward $r_{\text{poi}}(t)$ is a positive value awarded when the UAV successfully detects or visits a PoI. The amount of $r_{\text{poi}}(t)$ depends on the risk level of the PoI e.g., higher-risk PoIs (HR) yield a higher reward than lower-risk PoIs (LR). As shown in Equation 13, a dynamic multiplier $N_{\text{LR}}(t)$ is employed to incentivise the early visitation of HR PoIs before LRs. The value of $N_{\text{LR}}(t)$ denotes the number of unvisited LR PoIs on the map at time t. According to it, $N_{\text{LR}}(t)$ increases when the number of unvisited LR PoIs grows. It results in increased rewards for visiting HR PoIs. However, $N_{\text{LR}}(t)$ decreases if there are fewer unvisited LR PoIs. It results in reduced reward for HR PoIs, and thus encouraging earlier visits to HR and MR PoIs before LR ones.

$$r_{\text{poi}}(t) = \phi(\pi, p_t, t) \times \begin{cases} 2 \times N_{\text{LR}}(t) & \text{if } p_t \in \text{HR}, \\ 1 \times N_{\text{LR}}(t) & \text{if } p_t \in \text{MR}, \\ 0.5 & \text{if } p_t \in \text{LR}. \end{cases} \tag{13}$$

- Blind reward $r_{\text{pow}}(t)$ is a penalty given if the UAV moves to empty grid cells or flies out of the fire field. It is calculated using Equation 14 to avoid blind flights or loops.

$$r_{\text{pow}} = -\frac{0.15}{M * M} \tag{14}$$

As Algorithm 1 shows, the proposed approach enables the UAV to learn an optimal policy π^* that effectively balances the FAR objectives. It uses the expected cumulative reward of the Bellman Equation 15 to initialise and iteratively update the Q-values. By this, Q-learning actions are taken based on the $\epsilon-$ greedy strategy when the exploration rate ϵ decreases according to ϵ_{decay} as Q-learning environment exploration moves to exploitation.

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \tag{15}$$

Algorithm 1 Q-learning Algorithm

Require: State space S, Action space A, learning rate α, discount factor γ, exploration rate ϵ, minimum exploration rate $\epsilon_{\min}$, decay factor for ϵ ϵ_{decay}
Ensure: Optimized policy π^*

```
Initialize Q(s, a) arbitrarily for every s ∈ S and a ∈ A
for each episode do
    Set the initial state s
    while the state s is not terminal do
        With probability (1 − ϵ):
            if (random number ≤ 1 − ϵ) then
                Select a = arg max_a' Q(s, a')
            else
                Select a random action a
            end if
        Perform action a, observe reward r, and next state s'
        Update Q(s, a) ← Q(s, a) + α[r + γ max_a' Q(s', a') − Q(s, a)]
        Set s ← s'
    end while
    Update ϵ ← max(ϵ_min, ϵ × ϵ_decay)
end for
```

IV. Experiments and Evaluations

A. Simulation Set-up

A simulation is used to deploy the proposed approach and evaluate its performance in this research. It is built based on the $2D$ system model with *m* PoIs, and *n* fire sources which are randomly scattered throughout a grid environment.

A single UAV stays in charge of FAR operation and starts its mission from the Base-station which is located in the top left for all scenarios. It moves horizontally or vertically with a constant speed to find/visit the PoIs. The UAV power consumption is modelled using Matrice 200 energy model (0.055 unit/s per flight step) [46]. To test the functionality, scalability and performance of the proposed solution, three FAR scenarios including *Forest8*, *Forest10*, and *Forest12* are implemented with variations in field size, number of PoIs, and fire source/obstacle count. *Forest8* is deployed in a 8×8 grid with $\{4, 8, 16\}$ PoIs and $\{2, 4, 8\}$ fire obstacles. Similarly, *Forest10* is configured in a 10×10 environment, featuring $\{8, 16, 32\}$ PoIs and $\{4, 8, 16\}$ fire sources, while *Forest12* is built in a 12×12 grid containing $\{16, 32, 64\}$ PoIs and $\{8, 16, 32\}$ fire obstacles. The following metrics are measured to evaluate the performance of the proposed solution [25], [28]:

1) Coverage Ratio (CR) is the ratio of the visited/marked PoIs to the total number of PoIs.
2) Safety Ratio (SR) is calculated as the proportion of collision-free flight steps to the total number of traversed steps.
3) Energy Conservation (EC) refers to the residual energy at the end of the FAR mission.

B. Training and Convergence

This section demonstrates the training and convergence of the proposed Q-learning method which is deployed using the setup parameters listed in Table III.

TABLE III
Q-LEARNING SETUP PARAMETERS

Parameter	Value
Learning Rate (α)	0.001
Discount Factor (γ)	0.97
Initial Epsilon	1
Minimum Epsilon	0.001
Epsilon Decay	0.9996

As Figure 2 shows, flight steps are reduced when the Q-learning model is progressively trained. This is because the proposed approach learns the environment (i.e., PoIs and obstacles) after a sufficient number of exploration episodes. Hence, the flight length is decreased and a faster convergence is achieved if the environment size and/or complexity (i.e., the number of fire sources or PoIs) is reduced.

Figure 3 reports the total number of UAV collisions. According to it, the collision rate is sharply reduced once the Q-learning receives sufficient training. The training episodes enable the UAV to iteratively update the location of fire sources and obstacles resulting in increased collision avoidance.

Figure 4 and Figure 5 demonstrate the accumulative reward and visiting PoIs. The UAV initially takes numerous exploration flights/steps, but eventually refines the routes and flies through shortest routes when the Q-learning algorithm is fully trained. It increases the reward trend and optimises FAR objectives by reducing flight length (energy conservation), increasing the number of PoIs (rescue rate), and decreasing collisions (flight safety). However, there are still fluctuations, especially in large and complex environments, as the proposed algorithm still oscillates among near-optimal strategies after its convergence. This is because of the uncertainties of the sequence of the visiting PoIs where the field deployed is large and complex.

C. The Performance of the Proposal

The FAR trajectories produced by our proposed Q-learning method for the three forest scenarios are depicted in Figure 6. As it illustrates, the UAV initiates its FAR mission from the Basestation, navigating towards the PoIs (HR PoIs in red, MR in blue, and LR in green) following the fire risk priorities, and avoiding blind manoeuvres and collisions.

TABLE IV
THE PERFORMANCE OF THE PROPOSED APPROACH

Environment	PoI	Fire obstacles	EC	SR	CR
Forest 8 (8x8)	[16,8]	[8,4]	0.73	1.00	0.90
	[8,4]	[4,2]	0.82	1.00	0.97
	[4,2]	[2,0]	0.86	1.00	1.00
Forest 10 (10x10)	[32,16]	[16,8]	0.59	1.00	0.89
	[16,8]	[8,4]	0.68	1.00	0.95
	[8,4]	[4,0]	0.77	1.00	0.98
Forest 10 (12x12)	[64,32]	[32,16]	0.03	1.00	0.83
	[32,16]	[16,8]	0.48	1.00	0.93
	[16,8]	[8,0]	0.61	1.00	0.98

Table IV shows the results of the proposed Q-learning approach according to the three FAR scenarios each of which with 50 random PoI/fire localisation seeds. As the results demonstrate, EC is reduced when the environment size and complexity (i.e., the number of PoIs and fire sources) are increased. It sharply drops to 0.03 in the most complex scenario (12×12 environment with 64 PoIs and 32 fire obstacles) as computation overhead and training costs increase significantly. SR results show that the proposed approach has the capacity to successfully detect the location of fire flames/obstacles and establish collision-free routes, while CR results demonstrate that the proposed approach is able to discover more than 90% of PoIs in small environments. However, CR slightly decreases due to the existing trade-off between energy conservation and coverage ratio in large and complex environments.

D. Benchmarks

This research employs the following benchmarks to evaluate the performance of the proposed approach:

- The Double Deep Q-network (DDQN) benchmark tunes the proposed Q-learning method with two neural networks, and is deployed using the setup parameters listed in Table V. As Algorithm 2 shows, it enables the UAV to deeply learn about environmental states and approximate Q-values to take best-fitted actions. It leads to the Q-value generalisation, enabling support for a broader spectrum of RL scenarios. The DDQN employs a Prioritised Experience Replay (PER) technique [47] to optimise the sample efficiency and determine the priority of experiences using TD error which measures the difference between the estimated and target Q-values. It highlights experiences

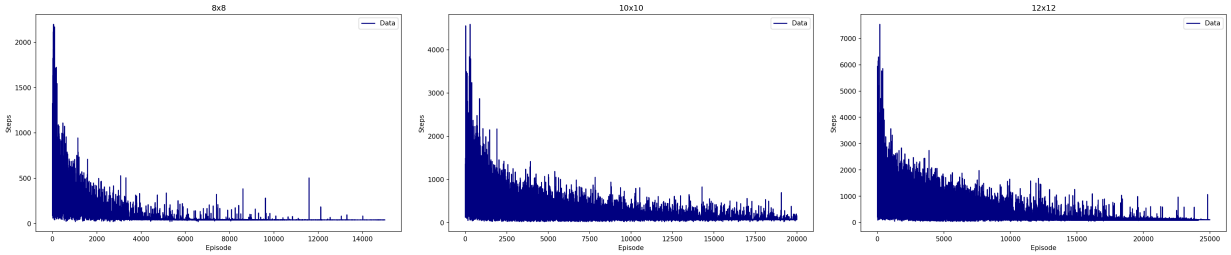

Fig. 2. Traversed flight steps during Q-learning training.

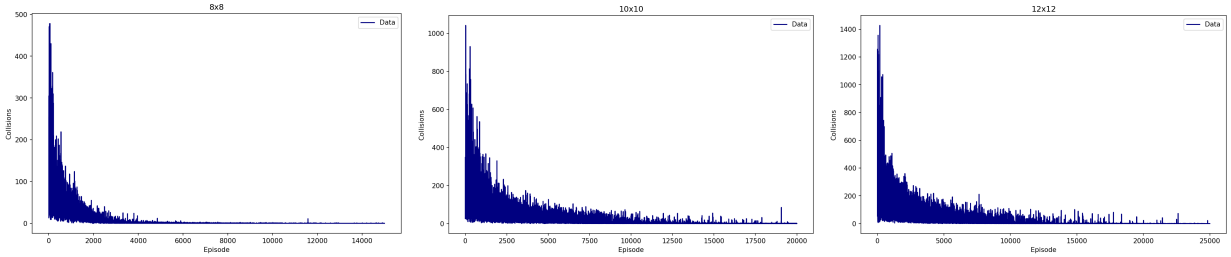

Fig. 3. UAV collisions during Q-learning training.

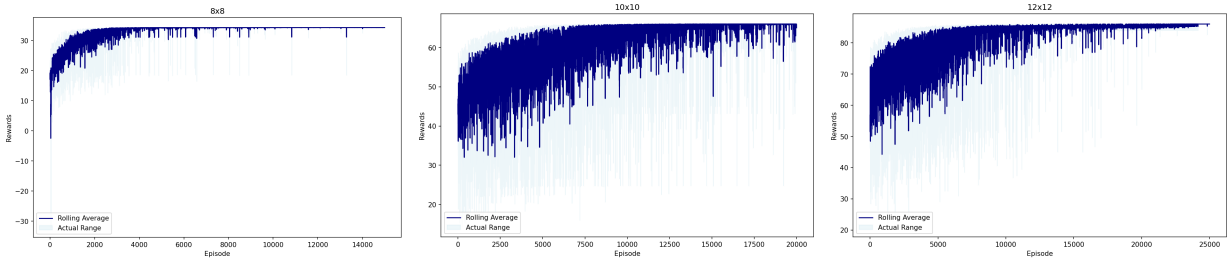

Fig. 4. PoI Rewards during Q-learning training.

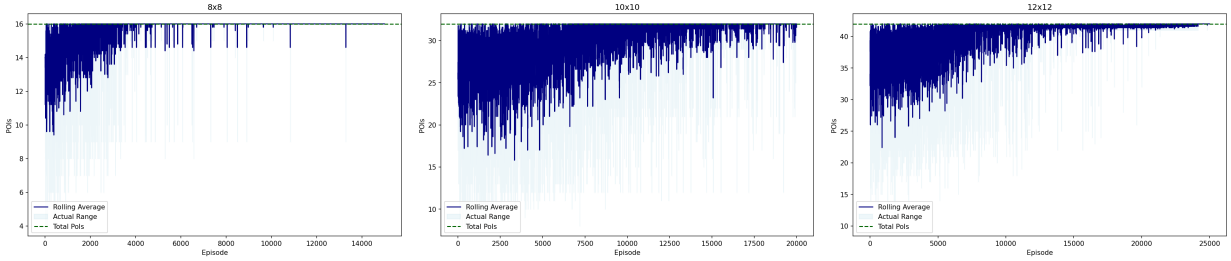

Fig. 5. Visiting PoIs during Q-learning training.

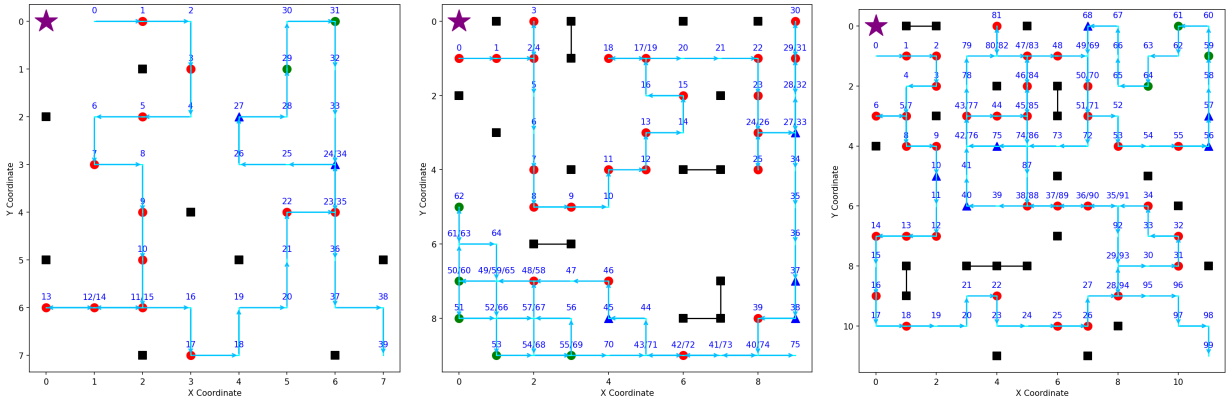

Fig. 6. UAV Trajectories Formulated by Proposed Method.

Algorithm 2 Double Deep Q-network (DDQN) Algorithm

Require: State space S, Action space A, learning rate α, discount factor γ, exploration rate ϵ, minimum exploration rate $\epsilon_{\min}$, and decay factor ϵ_{decay}
Ensure: Optimal policy π^*
Initialize $Q(s, a)$ randomly for every $s \in S$ and $a \in A$
Initialize the target network $Q'(s, a)$ with weights θ'
Set up replay memory D with capacity N
for each episode **do**
  Set initial state $s_1 = \{x_1\}$ and define $\phi_1 = \phi(s_1)$
  for $t = 1$ to T **do**
    With probability $(1 - \epsilon)$:
      Select $a_t = \arg\max_a Q(\phi(s_t), a; \theta)$
      Otherwise, choose a random action a_t
    Perform action a_t and observe reward r_t and the next observation x_{t+1}
    Set the next state $s_{t+1} = s_t, a_t, x_{t+1}$ and preprocess the next observation $\phi_{t+1} = \phi(s_{t+1})$
    Store the transition $(\phi_t, a_t, r_t, \phi_{t+1})$ into replay memory D
    Randomly sample a mini-batch of transitions $(\phi_j, a_j, r_j, \phi_{j+1})$ from D
    Define the target value y_j based on:
      $y_j = r_j$ for terminal states ϕ_{j+1}
      $y_j = r_j + \gamma \max_{a'} Q'(\phi_{j+1}, a'; \theta')$ for non-terminal states ϕ_{j+1}
    Apply a gradient descent step on the error $(y_j - Q(\phi_j, a_j; \theta))^2$
      with respect to the network parameters θ
  end for
  Update ϵ: $\epsilon \leftarrow \max(\epsilon_{\min}, \epsilon \times \epsilon_{\text{decay}})$
  Update the target network: $Q' = Q$
end for

with high TD errors during the training process to indicate the significant discrepancy between UAV exploration and the actual scenario/map. PER is an extension of the experience replay memory method that stores experience tuples (s, a, r, s') based on their priorities in contrast to the standard experience replay memory which uniformly samples the experiences. It facilitates training to focus on the most informative experience and improve sample efficiency and fast convergence. The priority of each experience (p_i) is calculated using Equation 16, where $Q_\theta(s, a)$ is the estimated Q-value, $Y(s, a, s')$ is the target Q-value.

$$p_i = |Q_\theta(s, a) - Y(s, a, s')| \tag{16}$$

Equation 17 is used to sample experiences from the DDQN priority queue. For this, $Pb(i)$ is the sampling probability value, p_i is the priority, and α is the hyper-parameter. The hyper-parameter α determines the degree of the prioritisation.

$$Pb(i) = \frac{p_i^\alpha}{\sum_k p_k^\alpha} \tag{17}$$

Deep Q-learning solutions usually suffer overestimation bias and bootstrapping due to the maximisation step. [48] resolves this drawback by decoupling the learning architecture using two neural networks. It comprises an online network that is parameterised by θ and a target network that is modelled by θ'. The former is used to select the best-fitted actions, while the latter is used to estimate the target Q-value for actions taken. By this, our DDQN benchmark aims to minimise loss function $L_{DDQN}(\theta)$ which is calculated using Equation 18, where $(Q_\theta(s, a) - Y_{DDQN}(s, a, s'))^2$ represents the squared TD error between the Q-value estimated by the online network and the target value which is calculated using Equation 19 at the target network.

$$L_{DDQN}(\theta) = E_{s,a,s' \sim D}[(Q_\theta(s, a) - Y_{DDQN}(s, a, s'))^2] \tag{18}$$

$$Y_{DDQN}(s, a, s') = r(s, a) + \gamma Q_{\theta'}(s', \arg\max_{a'} Q_\theta(s', a')) \tag{19}$$

TABLE V
DDQN SETUP PARAMETERS

Parameter	Value
Max Memory	5000
Batch Size	512
Learning Rate (α)	0.6
Initial Epsilon	1
Minimum Epsilon	0.005
Epsilon Decay	Linear
Update Interval	Per Episode

- Advantage Actor-Critic (A2C) benchmark [17] comprises a policy network $\pi(a|s; \theta)$ for action selection, and a Value Network $V(s; \omega)$ to evaluate actions which are chosen in the line with UAV navigation policy. TD error (δ_t) is used in the A2C advantage function to update action selection probabilities and measure the performance of the actions for which the discrepancy between the expected reward $(r_t + \gamma V(s_{t+1}; \omega))$ where the current value $(V(s_t; \omega))$ is estimated. Equation 20 is used to update the network parameters for the Actor function, while Equation 21 updates the Critic function where α and β are the learning rates for policy and value networks respectively.

$$\theta \leftarrow \theta + \beta \cdot \delta_t \cdot \nabla_\theta \log \pi(a_t|s_t; \theta) \tag{20}$$

$$\omega \leftarrow \omega + \alpha \cdot \delta_t \cdot \nabla_\omega V(s_t; \omega) \tag{21}$$

- The GA-based benchmark is redeployed from [18] in line with the PoI detection objective. For this, the proposed reward function is used as the fitness value of the GA. The GA measures collisions during the training process and stops the training if there is no observable improvement of the maximum fitness value for a sequence of generations.

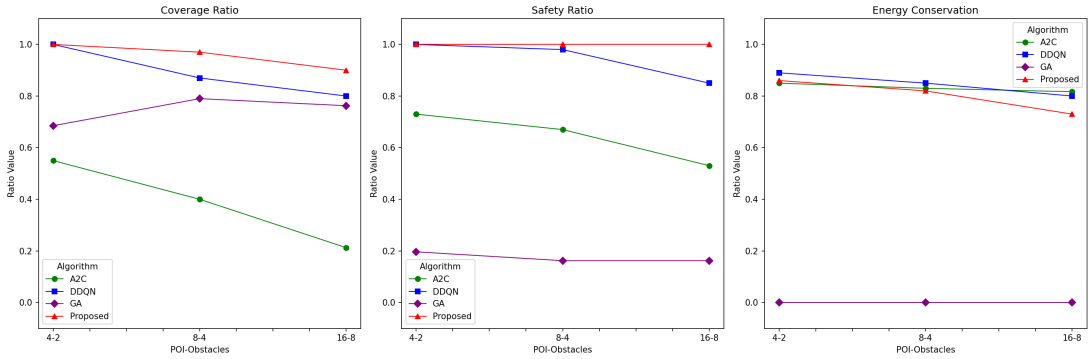

Fig. 7. Results of Forest8

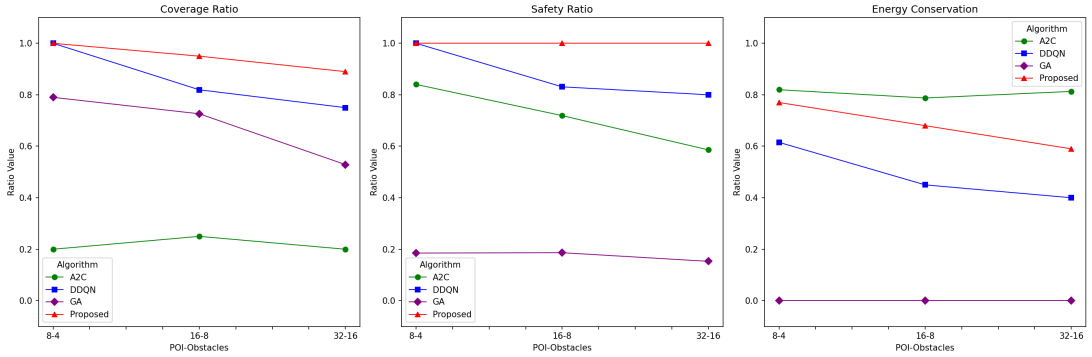

Fig. 8. Results of Forest10

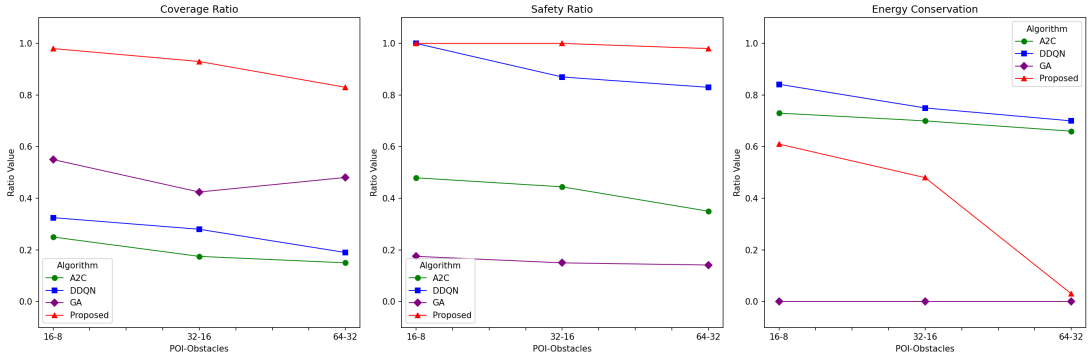

Fig. 9. Results of Forest12

E. Results and Discussions

This section presents and discusses the performance results of the proposed approach against the benchmarks.

As Figure 7, 8, and 9 depict, the proposed Q-learning solution outperforms benchmarks in terms of coverage and safety ratio. It outperforms GA and DDQN benchmarks by 10% to 20% CR in small environments (i.e., *Forest10* or *Forest8*), while giving 50% improvement when the environment complexity is increased (i.e., *Forest12*). In addition, it gives greater flight safety confidence and outperforms the benchmarks with no obvious SR drop when the environment size/complexity is increased. This means the proposal has the capacity to establish collision-free routes that cover a significant number of PoIs. In contrast, CR and SR are reduced in DDQN as it suffer from extensive training cost that limits sample efficiency and environment exploration, especially in large and complex FAR scenarios (i.e., *Forest12*).

The proposed Q-learning underperforms A2C in terms of energy conservation. This stems from the fact that the Q-learning algorithm takes multiple objects into account to plan UAV flights. Q-learning's energy consumption is increased as it aims to find a greater number of PoIs to improve the coverage ratio at the same time. According to Figure 9, the energy conservation of Q-learning sharply drops in Forest 12 with 64 and 32 PoIs and fire sources. This is because the proposed Q-learning aims to find and prioritise a greater number of PoIs that result in increased flight steps and consequently reduce energy conservation. However, the

A2C algorithm only focuses on a single objective (minimising energy consumption) during route planning. Hence, A2C gives better energy conservation in large and dense fields, while its SR and CR sharply drop.

DDQN gives good results in small environments (i.e., 8×8 with 8 PoI and 4 fire sources), but its CR is gradually reduced in large and complex areas (i.e., *Forest12*). It is because DDQN requires a larger volume of training samples to adequately explore the environment, collect the spatial distribution of PoIs in the exploration stage, and iteratively refine the routes during each episode by interconnecting all detected routes to find the shortest flight path. However, DDQN fails to fully learn large and complex environments due to the extensive training required and environment exploration cost.

The proposed Q-learning outperforms GA as it avoids blind/random flights. The GA algorithm is unable to manage FAR applications that require balancing multiple objectives simultaneously, yet, it tends to guide the UAV along arbitrary paths focused on a single predefined objective.

V. Conclusion and future work

This research delves into UAV path planning and proposes a Q-learning approach tuned with a novel reward function to conduct path planning with conflicting objectives. It responds to the existing research gap on the use of UAVs in FAR applications especially in large and complex environments (i.e., Metropolitan or woodland). By this, a UAV flies over a complex fire field to automatically find fire flames and mark trapped lives/animals according to their fire risk priorities. This aims to establish an optimal path to rescue the targets where the flight length is reduced, UAV collision is minimised and the number of targets recused is maximised at the same time. A simulation is carried out across three different FAR scenarios, varying in field size, rescue target count, and number of fire sources.

The performance of the proposed Q-learning approach is measured and compared with three benchmarks including DDQN, A2C, and GA. According to the results, the proposed approach outperforms the benchmarks in terms of coverage ratio and safety for the three environment scenarios, while it gives slightly lower energy conservation when the environment is large and complex. This stems from the fact that the proposed Q-earning algorithm has the capacity to optimise a triangle trade-off to give maximised energy conservation, flight safety, and field coverage at the same time.

Further investigations on the Q-learning solution optimisation and generalisation are still needed. For this, a more efficient training strategy needs to be developed to train the AI-enabled path planning method with the capacity to meet more realistic environmental scenarios such as irregularities of the field shapes, odd field sizes, and mobile PoIs and/or fire sources. Deep learning solutions can be a promising approach to meet these requirements and improve generalisation if the statues of the environment are frequently mapped into pixel images and form a large state space. However, the challenge of exploration efficiency must be resolved to avoid excessive computational costs and slow convergence during training.

References

[1] P. Zhu, L. Wen, D. Du, X. Bian, H. Fan, Q. Hu, and H. Ling, "Detection and tracking meet drones challenge," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 44, no. 11, pp. 7380–7399, Nov. 2022.

[2] A. O. Hashesh, S. Hashima, R. M. Zaki, M. M. Fouda, K. Hatano, and A. S. T. Eldien, "Ai-enabled uav communications: Challenges and future directions," IEEE Access, vol. 10, pp. 92 048–92 066, 2022.

[3] F. B. Sorbelli, F. Coro, S. K. Das, and C. M. Pinotti, "Energy-constrained delivery of goods with drones under varying wind conditions," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 6048–6060, Sep. 2021.

[4] M. Hassanalian and A. Abdelkefi, "Classifications, applications, and design challenges of drones: A review," Progress in Aerospace Sciences, vol. 91, pp. 99–131, 2017.

[5] C. Xu, X. Liao, J. Tan, H. Ye, and H. Lu, "Recent research progress of unmanned aerial vehicle regulation policies and technologies in urban low altitude," IEEE Access, vol. 8, pp. 74 175–74 194, 2020.

[6] M. Khosravi, S. Enayati, H. Saeedi, and H. Pishro-Nik, "Multi-purpose drones for coverage and transport applications," IEEE Transactions on Wireless Communications, vol. 20, no. 6, pp. 3974–3987, Jun. 2021.

[7] A. Hinas, J. M. Roberts, and F. Gonzalez, "Vision-based target finding and inspection of a ground target using a multirotor uav system," Sensors, vol. 17, no. 12, p. 2929, 2017.

[8] M. Naeem, S. T. H. Rizvi, and A. Coronato, "A gentle introduction to reinforcement learning and its application in different fields," IEEE Access, vol. 8, pp. 209 320–209 344, 2020.

[9] K. Man, K. Tang, and S. Kwong, "Genetic algorithms: concepts and applications [in engineering design]," IEEE Transactions on Industrial Electronics, vol. 43, no. 5, pp. 519–534, 1996.

[10] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction. MIT press, 2018.

[11] B. Jang, M. Kim, G. Harerimana, and J. W. Kim, "Q-learning algorithms: A comprehensive classification and applications," IEEE Access, vol. 7, pp. 133 653–133 667, 2019.

[12] Z. Wang, X. Yang, H. Hu, and Y. Lou, "Actor-critic method-based search strategy for high precision peg-in-hole tasks," in 2019 IEEE International Conference on Real-time Computing and Robotics (RCAR). IEEE, Aug. 2019.

[13] Z. Ma, C. Wang, Y. Niu, X. Wang, and L. Shen, "A saliency-based reinforcement learning approach for a uav to avoid flying obstacles," Robotics and Autonomous Systems, vol. 100, pp. 108–118, 2018.

[14] M. Aljehani and M. Inoue, "Performance evaluation of multi-uav system in post-disaster application: Validated by hitl simulator," IEEE Access, vol. 7, pp. 64 386–64 400, 2019.

[15] X. Chen, B. Hopkins, H. Wang, L. O'Neill, F. Afghah, A. Razi, P. Fule, J. Coen, E. Rowell, and A. Watts, "Wildland fire detection and monitoring using a drone-collected rgb/ir image dataset," IEEE Access, vol. 10, pp. 121 301–121 317, 2022.

[16] M. Lei, S. Fowler, J. Wang, X. Zhang, B. Yu, and B. Yu, "Double deep q-learning network-based path planning in uav-assisted wireless powered noma communication networks," in 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall). IEEE, Sep. 2021.

[17] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in International conference on machine learning. PMLR, 2016, pp. 1928–1937.

[18] Z. Zhou, J. Feng, B. Gu, B. Ai, S. Mumtaz, J. Rodriguez, and M. Guizani, "When mobile crowd sensing meets uav: Energy-efficient task assignment and route planning," IEEE Transactions on Communications, vol. 66, no. 11, pp. 5526–5538, 2018.

[19] S. Rezwan and W. Choi, "Artificial intelligence approaches for uav navigation: Recent advances and future challenges," IEEE Access, vol. 10, pp. 26 320–26 339, 2022.

[20] L. Yang, J. Qi, J. Xiao, and X. Yong, "A literature review of uav 3d path planning," in Proceeding of the 11th World Congress on Intelligent Control and Automation. IEEE, 2014, pp. 2376–2381.

[21] C. Qu, W. Gai, J. Zhang, and M. Zhong, "A novel hybrid grey wolf optimizer algorithm for unmanned aerial vehicle (uav) path planning," Knowledge-Based Systems, vol. 194, p. 105530, 2020.

[22] X. Yu, C. Li, and J. Zhou, "A constrained differential evolution algorithm to solve uav path planning in disaster scenarios," Knowledge-Based Systems, vol. 204, p. 106209, 2020.

[23] S. Ragi and E. K. Chong, "Uav path planning in a dynamic environment via partially observable markov decision process," IEEE Transactions on Aerospace and Electronic Systems, vol. 49, no. 4, pp. 2397–2412, 2013.

[24] V. Roberge, M. Tarbouchi, and G. Labonté, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning," IEEE Transactions on industrial informatics, vol. 9, no. 1, pp. 132–141, 2012.
[25] C. H. Liu, Z. Dai, Y. Zhao, J. Crowcroft, D. Wu, and K. K. Leung, "Distributed and energy-efficient mobile crowdsensing with charging stations by deep reinforcement learning," IEEE Transactions on Mobile Computing, vol. 20, no. 1, pp. 130–146, 2019.
[26] B. Chen, S. Lai, C. Chen, P. Shu, S. Chen, Z. Lai, and L. Xu, "Uav path planning based on improved genetic algorithm," in 2021 3rd International Symposium on Robotics & Intelligent Manufacturing Technology (ISRIMT). IEEE, Sep. 2021.
[27] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 42, no. 6, pp. 1291–1307, Nov. 2012.
[28] H. Qie, D. Shi, T. Shen, X. Xu, Y. Li, and L. Wang, "Joint optimization of multi-uav target assignment and path planning based on multi-agent reinforcement learning," IEEE access, vol. 7, pp. 146 264–146 272, 2019.
[29] P. Osinenko, G. Yaremenko, G. Malaniya, and A. Bolychev, "An actor-critic framework for online control with environment stability guarantee," IEEE Access, vol. 11, pp. 89 188–89 204, 2023.
[30] C. C. Millan-Arias, B. J. T. Fernandes, F. Cruz, R. Dazeley, and S. Fernandes, "A robust approach for continuous interactive actor-critic algorithms," IEEE Access, vol. 9, pp. 104 242–104 260, 2021.
[31] A. Elhussein and M. S. Miah, "A novel model-free actor-critic reinforcement learning approach for dynamic target tracking," in 2020 IEEE Midwest Industry Conference (MIC). IEEE, Aug. 2020.
[32] C. J. Watkins and P. Dayan, "Q-learning," Machine learning, vol. 8, no. 3-4, pp. 279–292, 1992.
[33] S. Pourroostaei Ardakani and A. Cheshmehzangi, "Reinforcement learning-enabled uav itinerary planning for remote sensing applications in smart farming," Telecom, vol. 2, no. 3, pp. 255–270, Jul. 2021.
[34] C. Wu, B. Ju, Y. Wu, X. Lin, N. Xiong, G. Xu, H. Li, and X. Liang, "Uav autonomous target search based on deep reinforcement learning in complex disaster scene," IEEE Access, vol. 7, pp. 117 227–117 245, 2019.
[35] N. I. Mowla, N. H. Tran, I. Doh, and K. Chae, "Afrl: Adaptive federated reinforcement learning for intelligent jamming defense in fanet," Journal of Communications and Networks, vol. 22, no. 3, pp. 244–258, 2020.
[36] Z. Yijing, Z. Zheng, Z. Xiaoyi, and L. Yang, "Q learning algorithm based uav path learning and obstacle avoidence approach," in 2017 36th Chinese Control Conference (CCC). IEEE, 2017, pp. 3397–3402.
[37] C. Yan and X. Xiang, "A path planning algorithm for uav based on improved q-learning," in 2018 2nd International Conference on Robotics and Automation Sciences (ICRAS). IEEE, 2018, pp. 1–5.
[38] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski et al., "Human-level control through deep reinforcement learning," nature, vol. 518, no. 7540, pp. 529–533, 2015.
[39] H. Bayerlein, M. Theile, M. Caccamo, and D. Gesbert, "Uav path planning for wireless data harvesting: A deep reinforcement learning approach," in GLOBECOM 2020-2020 IEEE Global Communications Conference. IEEE, 2020, pp. 1–6.
[40] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding, and F. Shu, "Path planning for uav-mounted mobile edge computing with deep reinforcement learning," IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5723–5728, 2020.
[41] U. Challita, W. Saad, and C. Bettstetter, "Interference management for cellular-connected uavs: A deep reinforcement learning approach," IEEE Transactions on Wireless Communications, vol. 18, no. 4, pp. 2125–2140, 2019.
[42] M. Theile, H. Bayerlein, R. Nai, D. Gesbert, and M. Caccamo, "Uav path planning using global and local map information with deep reinforcement learning," in 2021 20th International Conference on Advanced Robotics (ICAR). IEEE, 2021, pp. 539–546.
[43] L. Lv, S. Zhang, D. Ding, and Y. Wang, "Path planning via an improved dqn-based learning policy," IEEE Access, vol. 7, pp. 67 319–67 330, 2019.
[44] L. Tai and M. Liu, "Mobile robots exploration through cnn-based reinforcement learning," Robotics and biomimetics, vol. 3, pp. 1–8, 2016.
[45] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1238–1274, 2013.
[46] DJI, "Matrice 200 series," 2024, https://www.dji.com/uk/matrice-200-series, Retrieved (Jan 2024).
[47] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," arXiv preprint arXiv:1511.05952, 2015.
[48] H. van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," 2015.

VI. Biography Section

Yanming CHEN is a computer science researcher and software developer at the University of Nottingham Ningbo China (UNNC). His research interests include UAV communication and path planning, AGV operation and scheduling, computer vision, and neural networks.

Saeid POURROOSTAEI ARDAKANI currently works as a Senior Lecturer in Computer Science at the University of Lincoln, UK. He is an academic member of MLearn research group and an associated member of Lincoln Centre for Autonomous Systems (L-CAS). Saeid's research expertise centres on smart and adaptive computing and/or communication solutions to build collaborative/federated (sensory/feedback)systems in Internet of Things (IoT) applications and cloud environments. He is also interested in (ML-enabled) Big Data processing and analysis applications. To date, Saeid has published two books and more than 100 journal articles, conference papers, and book chapters.