Dynamic UAV Swarm Control in Disaster Recovery via GenAI-based Graph Reinforcement Learning

Bishmita Hazarika[†], Piyush Singh[*], Keshav Singh[‡], Octavia A. Dobre[†] and Trung Q. Duong[†]
[*]Department of Electrical Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan
[†]Faculty of Engineering and Applied Science, Memorial University of Newfoundland, St. Johns, Canada
[‡]Institute of Communications Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan
Email: bhazarika@mun.ca, piyush230497@gmail.com, ksingh1980@ieee.org, {odobre,tduong}@mun.ca

***Abstract*—This study presents a dynamic UAV swarm framework to support ground networks in disaster zones. The framework leverages Generative AI (GenAI) for real-time hover point generation to guide waypoint-based UAV navigation and realistic task modeling, integrated with graph neural networks (GNN) for safe navigation and obstacle avoidance. A multi-agent graph reinforcement learning (MAGRL) mechanism optimizes UAV coordination, enhancing energy efficiency, task completion, and load balancing in response to environmental changes. The framework's graph attention mechanism further improves inter-UAV communication, enabling adaptive task allocation and efficient coverage of high-risk zones. Extensive simulations show that the integrated GenAI-GNN and MAGRL approach achieves superior performance in task completion, energy savings, and system utility, outperforming benchmarks including MADDPG, GCRL, PSO, and Greedy strategies in dynamic disaster scenarios.**

***Index Terms*—Unmanned Aerial Vehicles (UAV), Generative AI (GenAI), Multi-Agent Reinforcement Learning (MARL)**

I. Introduction

NATURAL disasters, including earthquakes, floods, and wildfires, often disrupt communication networks, making it crucial to deploy resilient systems to aid in recovery. UAVs, integrated with IoT, provide mobile edge computing, sensing, and wide-area coverage, making them ideal for disaster recovery. However, their deployment faces challenges related to energy efficiency, adaptability, task allocation, and navigation, especially in complex and rapidly changing environments. Previous studies have focused on optimizing UAV trajectories and energy [1], [2], but most assume static conditions, which limit adaptability in dynamic scenarios.

Deep reinforcement learning (DRL) has shown promise in UAV deployment and task management [3], [4], yet its high convergence times and dataset requirements impede real-time decision-making [5]. Multi-agent DRL (MADRL) [6], [7] supports coordination but often results in slow convergence as agent numbers grow. To overcome these limitations, graph reinforcement learning (GRL) has emerged as a more structured approach for handling multi-agent coordination in complex environments [8] by modeling UAVs and ground terminals as nodes in a graph, which enables better communication [9]. While GRL supports dynamic adjustments, its iterative nature still leads to slow convergence in disaster environments.

Another approach gaining traction in UAV swarm coordination is federated learning (FL), which enables decentralized learning across UAV swarms without centralized data collection [10]; however, it struggles with intermittent connectivity and lacks real-world adaptability due to simulated training data [11]. Generative AI (GenAI) has the potential to address these gaps by producing realistic simulated data that aids UAV navigation and model training [12], [13]. Although GenAI enhances prediction accuracy, it does not independently support the real-time multi-agent coordination needed in disaster scenarios.

This study addresses the limitations in UAV swarm deployment for disaster recovery by introducing a framework that combines GenAI for real-time hover point generation, graph neural networks (GNN) for safe UAV coordination, and a multi-agent graph reinforcement learning (MAGRL) model to optimize task allocation, energy management, and coverage. Our approach leverages a graph attention mechanism to enhance inter-UAV communication, enabling high task completion and energy efficiency under dynamic conditions. By integrating GenAI with a graph-based model for hover point and task generation and coupling it with MAGRL, this framework achieves real-time adaptability, efficient task offloading, and robust UAV communication. The key contributions of this study are outlined below:

- We establish a comprehensive framework for UAV swarm deployment in disaster recovery scenarios to assist ground terminals, formulating an optimization problem that maximizes an overall utility function. This utility function integrates key factors such as task offloading, energy efficiency, coverage, and load balancing, while also accounting for critical constraints like task deadlines and the dynamic, unpredictable nature of disaster zones.
- We develop a UAV swarm framework for disaster response, optimizing a dynamic utility function that incorporates task offloading, energy efficiency, coverage, and load balancing while addressing the dynamic constraints of disaster zones.
- We integrate a GenAI model with GNN to dynamically generate energy-efficient hover points that avoid NFZs, enable collision avoidance, and produce realistic tasks, ensuring safe and effective UAV navigation and task handling in complex scenarios.

- We design a multi-agent graph reinforcement learning (MAGRL) model that leverages GenAI-GNN and includes a graph attention mechanism to refine hover point selection, task allocation, and UAV coordination, adapting dynamically to real-time environmental changes and optimizing communication efficiency.
- Finally, we conduct detailed simulations demonstrating significant improvements in task completion rates, energy efficiency, total utility, and UAV swarm performance, showing the advantages of our framework over benchmarks like MADDPG, GCRL, PSO, and Greedy approaches.

II. System Model

We examine a disaster-stricken area where traditional communication infrastructure is unavailable. The area is a $D_x \times D_y$ region with N_{GT} ground terminals (GTs) that support critical operations such as emergency response and sensor data processing. A swarm of N_{UAV} UAVs, each equipped with communication and mobile edge computing (MEC) capabilities, is deployed to provide coverage and offloading support. Operating at a fixed altitude h_u, UAVs handle resource-intensive tasks like video processing and sensor analysis. Fig. 1 illustrates a disaster scenario where UAVs establish inter-UAV communication links and connect with various ground terminals. UAV resources include CPU power $f_i(t)$, energy budget $E_{\max}(t)$, and limited buffer capacity to handle only a finite number of offloaded tasks from GTs. The UAVs dynamically adjust positions, $\mathbf{p}_u(t) = (x_u(t), y_u(t), h_u)$, to optimize task handling and avoid no-fly zones (NFZs). UAVs move to from position $\mathbf{p}_i(t)$ to $\mathbf{p}_i(t+\Delta t)$ optimize coverage, constrained by maximum speed $v_{\max}$ and must avoid NFZs:

$$\mathbf{p}_i(t+\Delta t) = \mathbf{p}_i(t) + \mathbf{v}_i(t)\cdot\Delta t\,, \tag{1}$$

where $\mathbf{v}_i(t)$ is the velocity vector of UAV i at time t.

Tasks generated by GT k are processed locally or offloaded to UAVs. A task T_k from GT k at time t requires D_k bits and μ_k CPU cycles per bit, with a deadline τ_k. Local and offloading processing times, $T_{\text{lo},k}(t)$ and $T_{\text{cmp},k,i}(t)$, are given by:

$$T_{\text{lo},k}(t) = \frac{D_k\cdot\mu_k}{f_k(t)}, \tag{2}$$

$$T_{\text{cmp},k,i}(t) = T_{\text{tr},i}(t) + \frac{D_k\cdot\mu_k}{f_i(t)}\,, \tag{3}$$

where $f_k(t)$ is the available computational resources at GT k, with transmission time at data rate $r_{i,k}(t)$ given as:

$$T_{\text{tr},i}(t) = \frac{D_k}{r_{i,k}(t)}. \tag{4}$$

Each UAV i has a computational load $L_i(t) = \sum_{k\in\mathcal{K}_i(t)} D_k\cdot\mu_k$, at time t, which is the cumulative processing demand of tasks offloaded where $\mathcal{K}_i(t)$ represents the GTs offloading tasks to UAV i at time t. To ensure timeliness, tasks—whether processed locally or offloaded—must be completed before their deadlines, i.e., $T_{lo,k}(t) \le \tau_k$ or $T_{cmp,k,i}(t) \le \tau_k$.

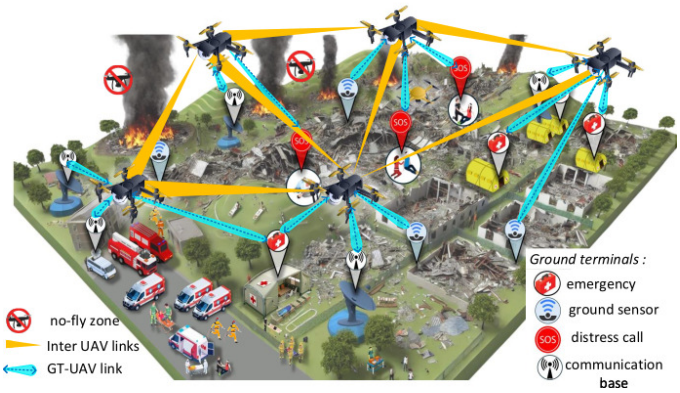

Fig. 1: Illustration of a UAV swarm providing support to various ground terminals in a disaster-stricken area.

A. UAV Operation Model

Each UAV i has a probabilistic sensing range R_s affected by terrain and obstacles. The detection probability of UAV i for GT k at time t depends on line-of-sight (LOS) conditions:

$$P_d(k,i,t) = P_{\text{LOS}}(k,i)\cdot\exp\left(-\frac{\|\mathbf{p}_k - \mathbf{p}_i(t)\|}{R_s}\right), \tag{5}$$

where $P_{\text{LOS}}(k,i)$ is the LOS probability, affected by the elevation angle $\theta_{k,i}$. UAV-GT communication follows an air-to-ground (ATG) channel model, incorporating LOS and non-line-of-sight (NLOS) conditions. Channel gain $G_{i,k}(t)$ and the signal-to-interference-plus-noise ratio (SINR) $\gamma_{i,k}(t)$ for GT k are given by:

$$G_{i,k}(t) = \frac{P_{\text{LOS}}(k,i)}{L_{\text{LOS}}(k,i)} + \frac{(1-P_{\text{LOS}}(k,i))}{L_{\text{NLOS}}(k,i)} \tag{6}$$

$$\gamma_{i,k}(t) = \frac{G_{i,k}(t)\cdot P_t}{I_{i,k}(t)+\sigma^2}, \tag{7}$$

with data rate $r_{i,k}(t) = B\log_2(1+\gamma_{i,k}(t))$, where B is bandwidth and P_t is UAV transmission power.

The total energy consumption for UAV i at time t includes movement, computation, and communication energy. The movement energy $E_{\text{mh},i}(t)$ comprises the power for hovering, $\mathcal{P}_{\text{hov}}$, and the power for movement, $\mathcal{P}_{\text{mov}}$:

$$E_{\text{mh},i}(t) = \mathcal{P}_{\text{hov}}\cdot T_{\text{hov},i}(t) + \mathcal{P}_{\text{mov}}\cdot\|\mathbf{p}_i(t+\Delta t) - \mathbf{p}_i(t)\|\,, \tag{8}$$

where $T_{\text{hov},i}(t)$ is the hovering time, and $\|\mathbf{p}_i(t+\Delta t) - \mathbf{p}_i(t)\|$ is the distance traveled. The task computing energy is:

$$E_{\text{comp},i}(t) = \kappa(f_i(t))^2\cdot T_{\text{process},i}(t)\,, \tag{9}$$

where κ is a processor efficiency factor, and $T_{\text{process},i}(t)$ is the processing time. The communication energy is estimated as:

$$E_{\text{com},i}(t) = P_t\cdot T_{\text{tr},i}(t)\,. \tag{10}$$

B. Utility Model

Our system optimizes UAV deployment in disaster management by maximizing coverage, minimizing energy consumption, and balancing computation offloading. The utility model consists of coverage utility, energy utility, and load balancing utility.

1) Coverage and Energy Utility: Coverage utility maximizes the UAVs' coverage over GTs, adjusted for obstacles and redundant coverage. The coverage utility for UAV i at time t is defined as:

$$U_{cov,i}(t) = \sum_{k \in \mathcal{K}_i(t)} \left(1 - \frac{1}{N_{UAV,k}(t)}\right) \cdot P_d^{\text{obs}}(k,i,t) \quad (11)$$

where $N_{UAV,k}(t)$ is the number of UAVs covering GT k at time t, and $P_d^{\text{obs}}(k,i,t)$ is the obstacle-adjusted detection probability. The energy utility minimizes the total energy consumption for UAV i, denoted as

$$U_{eng,i}(t) = -E_i(t) \quad (12)$$

where $E_i(t)$ is the total energy consumption calculated as

$$E_i(t) = E_{\text{mh},i}(t) + E_{\text{com},i}(t) + E_{\text{comp},i}(t) + E_{\text{swm},i}(t) \quad (13)$$

2) Load Balancing and Computation Offloading Utility: This utility ensures that tasks generated by GTs are efficiently processed, either locally or offloaded to UAVs, to balance the computational load and meet task deadlines. For UAV i, the load balancing utility is:

$$U_{of,i}(t) = \sum_{k \in \mathcal{K}_i(t)} \frac{1}{L_i(t)} \cdot \mathbb{1}\left(\min\left(T_{\text{lo},k}(t), T_{\text{cmp},k,i}(t)\right) \leq \tau_k\right) \quad (14)$$

where τ_k is the task deadline for GT k.

3) Total Utility: The total utility for UAV i combines coverage, energy, and load-balancing utilities:

$$U_{total,i}(t) = U_{cov,i}(t) + U_{eng,i}(t) + U_{of,i}(t) \quad (15)$$

The objective is to maximize $U_{total,i}(t)$, ensuring effective UAV operation in the disaster area.

III. Problem Statement & Solution Approach

This work aims to optimize UAV swarm deployment in disaster recovery, focusing on maximizing GT coverage, minimizing UAV energy consumption, and balancing computational load for tasks offloaded from GTs.

1) Problem Statement: The main objective is to maximize the total utility $U_{total,i}(t)$ for each UAV i at time t, which includes coverage utility $U_{cov,i}(t)$, energy utility $U_{eng,i}(t)$, and load balancing utility $U_{of,i}(t)$. This is achieved through the following optimization problem:

$$\begin{aligned}
(\mathcal{P}): \quad & \max_{\{\mathbf{p}_i(t), f_i(t)\}} \sum_{i=1}^{N_{UAV}} U_{total,i}(t) \\
\text{s.t.} \quad & (C.1) \quad \|\mathbf{p}_i(t) - \mathbf{p}_i(t+\Delta t)\| \leq v_{max} \cdot \Delta t, \\
& (C.2) \quad \mathbf{p}_i(t) \notin \text{NFZ}, \\
& (C.3) \quad E_i(t) \leq E_{\max,i}, \\
& (C.4) \quad T_{\text{lo},k}(t) \leq \tau_k \,||\, T_{\text{cmp},k,i}(t) \leq \tau_k, \\
& (C.5) \quad f_i(t) \leq f_{\max,i}, \\
& (C.6) \quad \sum_{i=1}^{N} \rho_i \cdot f'_m \leq f_m, \\
& (C.7) \quad T_{\text{cmp},k,i}(t) + T_{\text{tr},i}(t) \leq \tau_k. \quad (16)
\end{aligned}$$

Here, $(C.1)$ restricts UAV mobility to a maximum speed v_{max}, $(C.2)$ enforces avoidance of no-fly zones (NFZs), and $(C.3)$ ensures energy consumption does not exceed $E_{\max,i}$. Constraint $(C.4)$ requires all tasks to meet deadlines, whether processed locally or offloaded, while $(C.5)$ limits CPU usage to $f_{\max,i}$. In $(C.6)$, the UAV's computational load, defined by ρ_i (where $\rho_i = 1$ for offloading), must not exceed capacity f_m. Finally, $(C.7)$ ensures offloaded tasks are completed within the required deadline τ_k by limiting the combined transmission and computation time.

2) Solution Approach: The solution framework optimizes UAV deployment in disaster recovery by dynamically prioritizing tasks based on risk and urgency, planning safe movement, and adapting swarm behavior. Disaster areas are divided into *Emergency Response* ($\mathcal{Z}_{ER}$) and *High Risk* ($\mathcal{Z}_{HR}$) zones, with tasks classified as *Emergency* ($\mathcal{C}_{\mathcal{E}}$), *Priority* ($\mathcal{C}_{\mathcal{P}}$), or *Routine* ($\mathcal{C}_{\mathcal{R}}$). UAVs allocate CPU resources by evaluating each task's size (D_k), deadline (τ_k), and available capacity, prioritizing high-risk tasks and signaling neighboring UAVs for assistance if needed. GenAI generates energy-efficient hover points, with UAVs using A^*-based path planning to avoid NFZs. A GNN-based collision-avoidance mechanism facilitates shared position and path data, enabling UAVs to adjust routes dynamically, prioritizing emergency tasks and UAVs with higher energy levels.

Adaptive swarm optimization is further achieved through a GenAI-based graph model for realistic hover point and task generation, while GNN supports real-time sharing of energy levels, task loads, and paths. A multi-agent graph reinforcement learning framework with attention mechanisms enables UAVs to update policies dynamically, prioritize tasks, conserve energy, and adapt to environmental changes, enhancing disaster response efficiency.

IV. Proposed Solution Framework

Our framework combines a Generative Adversarial Network with Graph Neural Network (GAN-GNN) model and a **M**ulti-**A**gent **G**raph **R**einforcement **L**earning (MAGRL) framework to optimize UAV deployment, task allocation, and adaptive decision-making in disaster recovery. This integrated approach enables UAVs to efficiently select hover points, manage tasks in real-time, and coordinate to maximize coverage, balance loads, and minimize energy consumption.

1) GAN-GNN for Hover Point and Task Generation: The GAN-GNN model initiates hover points and task assignments dynamically. The GAN's *Generator* (G) produces hover points $\mathbf{p}_i^{\text{hover}} = G(z_i, \mathcal{G})$ for UAVs and task parameters $T_k = G(z_k, \mathcal{G})$ at ground terminals (GTs), based on UAV and GT states encoded in latent vectors z_i, z_k, and communication links in graph $\mathcal{G}$. The *Discriminator* (D) ensures that generated points avoid NFZs, prevent collisions, and meet energy constraints.

$$D_{\text{NFZ}}(\mathbf{p}_i^{\text{hover}}) = \begin{cases} 1, & \text{if } \mathbf{p}_i^{\text{hover}} \notin \text{NFZ}, \\ 0, & \text{if } \mathbf{p}_i^{\text{hover}} \in \text{NFZ}, \end{cases} \quad (17)$$

Algorithm 1 GAN-GNN based MAGRL Framework

Input: UAV/GT states $\mathbf{s}_i(t)$, $\mathbf{s}_k(t)$; graph $\mathcal{G}$; policies $\pi_i(a_i|s_i)$, values $V_i(s_i)$; $E_{\max}$, $\mathbf{p}_i^{\text{hover}}$.
Output: Optimized UAV actions and hover points.
Initialize G, D, and GNN-based message passing.
for each UAV i and GT k **do**
    Encode UAV and GT state as latent vectors z_i and z_k.
    Update UAV embeddings with GNN message passing on $\mathcal{G}$ as in (25).
end for
for each hover point $\mathbf{p}_i^{\text{hover}}$ **do**
    Verify D_{NFZ}, D_{c}, and D_{en} from (19).
    Minimize generator and discriminator losses (20), (21).
end for
for each UAV i **do**
    Gather neighboring states via GNN-based message passing (27).
    Update UAV state embeddings $\mathbf{h}_i$ with GAT (25).
    Integrate T_k and τ_k from neighbors.
end for
for each UAV i **do**
    Select $a_i^{\text{move}}(t)$ based on $\pi_i(a_i|s_i)$ using hover point $\mathbf{p}_i^{\text{hover}}$ and A* path planning to avoid NFZ.
    Compute and communication actions $a_i^{\text{comp}}(t)$ and $a_i^{\text{comm}}(t)$ based on $L_i(t)$ and link requirements.
end for
for each UAV i **do**
    Calculate $r_i(t)$ (22) and TD error $\delta_i(t)$ (23).
    Update policy gradient and minimize $L_{\text{value},i}$ (29).
    Update total loss $\mathcal{L}_i$ with entropy regularization (30).
end for
for each UAV i and $j \in \mathcal{N}(i)$ **do**
    Adjust paths based on $\mathbf{p}_j^{\text{target}}(t)$ and collision indicator $\mathbb{1}_{\text{collide}}(i,j)$ (27).
end for
Repeat until convergence.

$$D_{\text{c}}(\mathbf{p}_i^{\text{hover}}) = \begin{cases} 1, & \text{if UAV } i \text{ avoids collisions,} \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

$$D_{\text{en}}(\mathbf{p}_i^{\text{hover}}) = \begin{cases} 1, & E_{\text{move},i} \le E_{\max,i}, \\ 0, & E_{\text{move},i} > E_{\max,i}. \end{cases} \tag{19}$$

The generator optimizes constraint through a loss function:

$$\mathcal{L}_G = \mathbb{E}\left[1 - D_{\text{NFZ}}\right] + \mathbb{E}\left[1 - D_{\text{en}}\right] + \mathbb{E}\left[1 - D_{\text{ts}}\right] + \mathbb{E}\left[1 - D_{\text{c}}\right], \tag{20}$$

The discriminator loss is defined as:

$$\mathcal{L}_D = \mathbb{E}\left[\log D\left(\mathbf{p}_i^{\text{hover}}, T_k\right)\right] + \mathbb{E}\left[\log\left(1 - D\left(G(z_i, \mathcal{G})\right)\right)\right], \tag{21}$$

2) Multi-Agent Graph Reinforcement Learning (MAGRL): The MAGRL framework optimizes UAV swarm coordination by refining hover points and task allocations from GAN-GNN, using a graph attention actor-critic mechanism (**MAGAC**) that balances UAV-GT interactions and adapts decisions based on real-time feedback. The key components of the MAGRL framework are:

State: Each UAV i's state $s_i(t)$ includes local data like $\mathbf{p}_i(t)$, $E_{\max}(t)$, $f_i(t)$, $L_i(t)$, and $\mathbf{p}_i^{\text{hover}}$. GNN-based message passing incorporates information on neighbors $\mathcal{N}(i)$ and GT task details (T_k, τ_k for $k \in \mathcal{K}_i(t)$).

Action: UAV actions $A_i(t)$ include movement $a_i^{\text{move}}(t)$ to update $\mathbf{p}_i(t+1)$, computation $a_i^{\text{comp}}(t)$ for processing or offloading, and communication $a_i^{\text{comm}}(t)$ for link management.

Reward: The reward $r_i(t)$ balances utility and penalizes rule violations, estimated as:

$$r_i(t) = U_{\text{total},i}(t) + \lambda_{\text{plty}} \cdot \mathbb{1}_{\text{NFZ}} + \lambda_{\text{dln}} \cdot \left(\mathbb{1}_{\text{miss}}(\tau_k) - \mathbb{1}_{\text{comp}}(\tau_k)\right) + \eta H(\pi_i(t)), \tag{22}$$

where $U_{\text{total},i}(t)$ measures utility, $\mathbb{1}_{\text{NFZ}}$ penalizes NFZ violations, λ_{dln} controls task completion penalties/rewards, and $H(\pi_i(t))$ encourages exploration.

Each UAV updates its policy $\pi_i(a_i|s_i)$ and value function $V_i(s_i)$, represented by neural networks. Policy gradients optimize action probabilities:

$$\nabla_{\theta_i} \pi_i(a_i|s_i) = \mathbb{E}\left[\nabla_{\theta_i} \log \pi_i(a_i|s_i) \cdot \delta_i(t)\right], \tag{23}$$

where $\delta_i(t) = r_i(t) + \beta V_i(s_i(t+1)) - V_i(s_i(t))$ represents the temporal difference (TD) error, with discount factor β. Stability mechanisms include gradient clipping and learning rate decay:

$$\nabla_{\text{clip}} = \text{clip}\left(\nabla_{\theta_i}, -c_{\text{clip}}, c_{\text{clip}}\right), \quad \eta_{\text{new}} = \gamma_{\text{decay}} \cdot \eta_{\text{old}}. \tag{24}$$

3) Graph Attention for Efficient Decision-Making: A graph attention mechanism (GAT) in MAGRL enables focused information aggregation from relevant neighbors:

$$\mathbf{h}_i = \phi\Big(\sum_{j \in \mathcal{N}(i)} \alpha_{ij} W \mathbf{h}_j\Big), \tag{25}$$

where α_{ij} is an attention coefficient based on the importance of UAV j to UAV i. Task loads and deadlines aid reallocation, while embeddings are updated for collision management:

$$\mathbf{h}_i^{\text{task}}(t+1) = \sigma\Big(\sum_{j \in \mathcal{N}(i)} W^{\text{task}} \mathbf{T}_j(t) + W^{\text{dln}} \tau_j(t) + b\Big), \tag{26}$$

$$\mathbf{h}_i^{\text{path}}(t+1) = \sigma\Big(\sum_{j \in \mathcal{N}(i)} W \mathbf{p}_j^{\text{target}}(t) + \hat{W} \mathbb{1}_{\text{collide}}(i,j) + b\Big), \tag{27}$$

where task deadlines and paths assist in reallocation and collision avoidance.

A. MAGAC for Policy Optimization

MAGAC optimizes UAV policies and value functions with an advantage function $A_i(s_i, a_i) = Q_i(s_i, a_i) - V_i(s_i)$ where $Q_i(s_i, a_i)$ approximates action-value. The policy loss is:

$$L_{\text{policy},i} = -\mathbb{E}\left[\log \pi_i(a_i|s_i) A_i(s_i, a_i)\right], \tag{28}$$

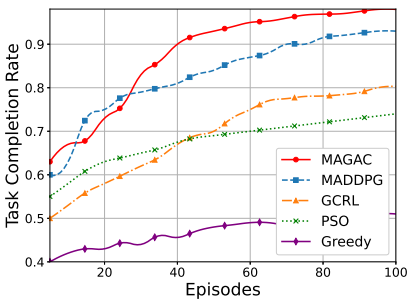

Fig. 2: Task completion rate vs. Episodes.

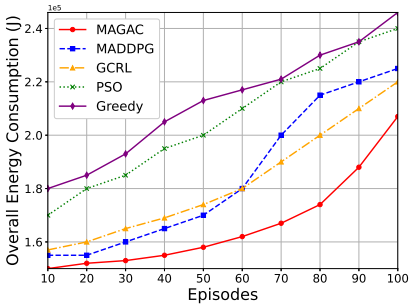

Fig. 3: Total energy vs. Episodes.

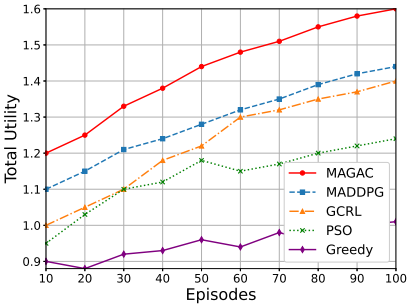

Fig. 4: Total utility vs. Episode.

TABLE I: Parameters

Parameter	Value	Parameter	Value
N_{GT}	100	N_{UAV}	20
h_u	100 m	NFZ_{buf}	10 m
v_{max}	25 m/s	R_s	500 m
P_{comp}^{UAV}	4×10^{-9} J/cycle	P_{comp}^{GT}	4×10^{-9} J/cycle
P_{comm}	8×10^{-9} J/bit	P_{move}	30 W/m
P_{hover}	180 W	$E_{\max}^{UAV}$	1×10^{6} J
B	2×10^{7} Hz	P_t	5 W
σ^2	1×10^{-9} W	$tasks/GT$	100
D_k	$(1-5)\times 10^{5}$ bits	f_k	$(1-10) \times 10^{5}$
τ_k	5 to 30 seconds	κ	1×10^{-10}
Batch	64	$LR_{\text{generator}}$	0.001
$LR_{\text{disc.}}$	0.001	LR_{actor}	0.0005
LR_{critic}	0.0005	z_i	100
c_v	0.5	c_e	0.01
β	0.99	ϵ	0.01
λ_{plty}	1.0	λ_{dln}	2.0
η	0.1	T_{max}	$100let0$
c_{clip}	0.5	γ_{decay}	0.95

while the value network minimizes the squared TD error:

$$L_{\text{value},i} = \mathbb{E}\left[\left(V_i(s_i(t)) - (r_i(t) + \beta V_i(s_i(t+1)))\right)^2\right]. \quad (29)$$

The total loss includes entropy regularization to encourage exploration:

$$\mathcal{L}_i = L_{\text{policy},i} + c_v L_{\text{value},i} - c_e H(\pi_i), \quad (30)$$

where $H(\pi_i) = -\sum_{a_i} \pi_i(a_i|s_i) \log \pi_i(a_i|s_i)$, and c_v, c_e control the weight of value loss and entropy regularization.

1) Integrated Operation with GAN-GNN and MAGRL: Initially, GAN-generated hover points $\mathbf{p}_i^{\text{hover}}$ provide optimized starting locations, which are refined through GNN updates. At each step, embeddings $\mathbf{h}_i(t)$ guide policy $\pi_i(a_i|s_i)$ and value networks $V_i(s_i)$ for movement actions:

$$a_i^{\text{move}}(t) = \arg\max_{a_i} \pi_i(a_i|s_i), \quad (31)$$

adjusting UAV positions to optimize energy, task allocation, and NFZ compliance,,ensuring effective disaster response.

V. Performance Evaluation

The disaster area for simulation was modeled as a 1000×1000 square meter grid, with tasks generated dynamically at a rate of 10 to 40 tasks per minute, using the GenAI model for realistic task dynamics. The *Telecom Italia Big Data Challenge* dataset was employed to emulate ground terminal task generation under urban disaster recovery conditions [14]. Fixed no-fly zones (NFZs) were modeled in circular, square, and rectangular patterns, with a buffer distance NFZ_{buf} to prevent UAV entry. The GAN model generated hover points dynamically, ensuring these points avoid NFZs and high-risk zones to optimize UAV positioning and task allocation. The primary goals for the UAV swarm included maximizing task completion, minimizing energy consumption, and avoiding NFZs, with real-time adaptive reallocation based on changing conditions. Simulation parameters are summarized in **Table I**.

A. Benchmark Comparisons

The proposed framework is accessed against following:

- *Multi-Agent Deep Deterministic Policy Gradient (MADDPG):* UAVs use MADDPG for task allocation and movement, beginning from grid-based hover points and adapting positions through continuous-action policies.
- *Graph Convolutional Reinforcement Learning (GCRL):* GCRL employs graph-based Q-learning for task allocation. UAVs start with grid-based hover points, refined through graph convolution layers.
- *Particle Swarm Optimization (PSO):* UAVs adjust positions iteratively using PSO to balance task completion and energy efficiency, updating hover points based on particle positions.
- *Greedy Approach:* UAVs make local, short-term decisions to minimize immediate energy use or maximize coverage, starting from grid-based hover points.

B. Results

1) Task Completion Rate Analysis: In Fig. 2, the task completion rates over time are shown, highlighting the effectiveness of the proposed MAGAC framework, which achieves a 98% completion rate by 100 episodes. This is due to GenAI's dynamic hover point optimization and GNN-driven inter-agent coordination, enabling real-time task adaptability. MADDPG reaches 93% completion with effective decentralized learning, though it lacks GNN-based coordination, limiting adaptability. GCRL attains 80.3% completion, benefiting from Q-learning's convergence over time. PSO achieves 74% with quick heuristic optimization but plateaus early. The Greedy approach performs lowest, at 51%, due to inefficient task allocation and minimal optimization.

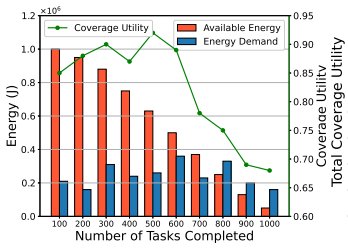

Fig. 5: Energy vs. Tasks vs. Coverage.

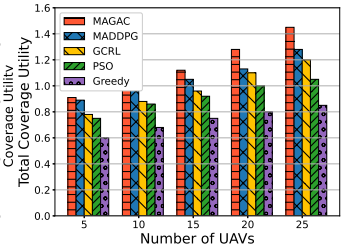

Fig. 6: Coverage utility vs. UAVs

2) Energy Consumption through episodes: Fig. 3 illustrates the overall energy consumption for different approaches across episodes. The MAGAC framework demonstrates superior energy efficiency, consuming the least energy due to the minimization of unnecessary movements and energy expenditures. MADDPG, while also energy-conscious, has slightly higher consumption as it lacks MAGAC's advanced inter-agent coordination, which leads to suboptimal energy allocation. GCRL initially shows moderate energy use but escalates in later episodes, reflecting slower learning convergence and less effective adaptation. PSO and Greedy approaches exhibit the highest energy consumption, with Greedy showing erratic increases due to its lack of a structured task allocation strategy.

3) Total Utility through episodes: Fig. 4 illustrates utility performance across approaches. MAGAC achieves superior utility by effectively balancing energy use, coverage, and task allocation, aided by GenAI-generated hover points and tasks that enable dynamic adaptation. MADDPG, although stable, lags due to its local optimization focus, lacking inter-agent coordination. GCRL improves in later episodes as graph-based information sharing reduces redundant actions, though its slower convergence initially limits performance. PSO's heuristic approach results in erratic utility growth, unable to adapt to complex task demands. Greedy prioritizes immediate rewards, leading to high energy consumption and poor long-term efficiency.

4) UAV Coverage Analysis: Fig. 5 and 6 analyze the coverage efficiency of the MAGAC framework under energy and UAV constraints. Fig. 5 shows that sustaining high coverage becomes challenging as tasks progress as UAV energy depletes. Initially, coverage remains robust, but it declines sharply after 300 and 800 task completions, reaching around 0.68 due to energy limitations that force UAVs to prioritize critical tasks in high-risk zones, thus limiting broad coverage. Fig. 6 demonstrates that MAGAC achieves the highest coverage utility by leveraging GenAI-generated hover points and GNN-based coordination, reaching 1.45 with 25 UAVs by reducing redundant paths and enhancing communication. Other approaches, such as MADDPG and GCRL, achieve moderate performance but lack robust inter-agent coordination, leading to slightly reduced and less efficient coverage.

VI. Conclusion

In this paper, we proposed an advanced framework for UAV swarm deployment in disaster recovery scenarios, integrating GenAI, GNN, and a MAGRL framework. Our approach addresses key challenges such as energy efficiency, task offloading, real-time adaptability, and safe navigation through dynamic hover point generation and GNN-based collision avoidance. Including graph attention further enhanced UAV coordination, leading to significant improvements across metrics like task completion rates, energy efficiency, and overall system utility. Through extensive simulations and benchmark comparisons with MADDPG, GCRL, PSO, and Greedy approaches, our proposed MAGAC framework demonstrated superior performance, achieving a 98% task completion rate while optimizing energy consumption. This synergy between GenAI and MAGRL allowed for efficient UAV swarm coordination and task allocation in dynamic and resource-constrained disaster scenarios.

References

[1] J. Dai, W. Pu, J. Yan, Q. Shi, and H. Liu, "Multi-UAV collaborative trajectory optimization for asynchronous 3-D passive multitarget tracking," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–16, 2023.

[2] H. Pan, Y. Liu, G. Sun, J. Fan, S. Liang, and C. Yuen, "Joint power and 3D trajectory optimization for UAV-enabled wireless powered communication networks with obstacles," *IEEE Trans. Commun.*, vol. 71, no. 4, pp. 2364–2380, 2023.

[3] P. Hou, Y. Huang, H. Zhu, Z. Lu, S.-C. Huang, Y. Yang, and H. Chai, "Distributed DRL-based intelligent over-the-air computation in unmanned aerial vehicle swarm-assisted intelligent transportation system," *IEEE Internet Things J.*, pp. 1–1, 2024.

[4] X. Mao, G. Wu, M. Fan, Z. Cao, and W. Pedrycz, "DL-DRL: A double-level deep reinforcement learning approach for large-scale task scheduling of multi-UAV," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–17, 2024.

[5] P. Singh, B. Hazarika, K. Singh, C. Pan, W.-J. Huang, and C.-P. Li, "DRL-based federated learning for efficient vehicular caching management," *IEEE Internet Things J.*, pp. 1–1, 2024.

[6] Y. Hou, J. Zhao, R. Zhang, X. Cheng, and L. Yang, "UAV swarm cooperative target search: A multi-agent reinforcement learning approach," *IEEE Trans. Intell. Veh.*, vol. 9, no. 1, pp. 568–578, 2024.

[7] X. Dai, Z. Lu, X. Chen, X. Xu, and F. Tang, "Multiagent RL-based joint trajectory scheduling and resource allocation in NOMA-assisted UAV swarm network," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14 153–14 167, 2024.

[8] M. Nie, D. Chen, and D. Wang, "Reinforcement learning on graphs: A survey," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 7, no. 4, pp. 1065–1082, 2023.

[9] Z. Liu, J. Zhang, E. Shi, Z. Liu, D. Niyato, B. Ai, and X. S. Shen, "Graph neural network meets multi-agent reinforcement learning: Fundamentals, applications, and future directions," *IEEE Wirel. Commun.*, pp. 1–9, 2024.

[10] B. Jiang, J. Du, C. Jiang, Z. Han, A. Alhammadi, and M. Debbah, "Over-the-air federated learning in digital twins empowered UAV swarms," *IEEE Trans. Wirel. Commun.*, pp. 1–1, 2024.

[11] B. Hazarika and K. Singh, "AFL-DMAAC: Integrated resource management and cooperative caching for URLLC-IoV networks," *IEEE Trans. Intell. Veh.*, vol. 9, no. 6, pp. 5101–5117, 2024.

[12] Y. Liu, H. Du, D. Niyato, J. Kang, Z. Xiong, D. I. Kim, and A. Jamalipour, "Deep generative model and its applications in efficient wireless network management: A tutorial and case study," *IEEE Wirel. Commun.*, vol. 31, no. 4, pp. 199–207, 2024.

[13] J. Wang, H. Du, D. Niyato, J. Kang, Z. Xiong, D. Rajan, S. Mao, and X. Shen, "A unified framework for guiding generative AI with wireless perception in resource constrained mobile edge networks," *IEEE Trans. Mob. Comput.*, pp. 1–17, 2024.

[14] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, "A multi-source dataset of urban life in the city of milan and the province of trentino," *Sci. data*, vol. 2, no. 1, pp. 1–15, 2015.