Coordinated Autonomous Drones for Human-Centered Fire Evacuation in Partially Observable Urban Environments

1st Maria G. Mendoza
Mechanical Engineering
University of California, Berkeley
Berkeley, USA
maria_mendoza@berkeley.edu

2nd Addison Kalanther
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, California
addikala@berkeley.edu

3rd Daniel Bostwick
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, USA
daniel.k.bostwick@berkeley.edu

4th Emma Stephan
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, USA
estephan@berkeley.edu

5th Chinmay Maheshwari
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, USA
chinmay_maheshwari@berkeley.edu

6th Shankar Sastry
Electrical Engineering and Computer Sciences
University of California, Berkeley
Berkeley, USA
sastry@coe.berkeley.edu

***Abstract*—Autonomous drone technology holds significant promise for enhancing search and rescue operations during evacuations by guiding humans toward safety and supporting broader emergency response efforts. However, their application in dynamic, real-time evacuation support remains limited. Existing models often overlook the psychological and emotional complexity of human behavior under extreme stress. In real-world fire scenarios, evacuees frequently deviate from designated safe routes due to panic and uncertainty.**

To address these challenges, this paper presents a multi-agent coordination framework in which autonomous Unmanned Aerial Vehicles (UAVs) assist human evacuees in real-time by locating, intercepting, and guiding them to safety under uncertain conditions. We model the problem as a Partially Observable Markov Decision Process (POMDP), where two heterogeneous UAV agents—a high-level rescuer (HLR) and a low-level rescuer (LLR)—coordinate through shared observations and complementary capabilities. Human behavior is captured using an agent-based model grounded in empirical psychology, where panic dynamically affects decision-making and movement in response to environmental stimuli.

The environment features stochastic fire spread, unknown evacuee locations, and limited visibility, requiring UAVs to plan over long horizons to search for a human and adapt in real time. Our framework employs the Proximal Policy Optimization (PPO) algorithm with recurrent policies to enable robust decision-making in partially observable settings. Simulation results demonstrate that the UAV team can rapidly locate and intercept evacuees, significantly reducing the time required for them to reach safety compared to scenarios without UAV assistance. We provide access to the results presented in this paper, along with additional simulations, at https://sastry-group.github.io/MultiRobot-HADR/

***Index Terms*—Fire Evacuation, Multi-agent System, Agent-based Modeling, Reinforcement Learning, Emergency Robotics, UAVs, Partial Observability, Disaster Relief**

I. Introduction

Wildfires increasingly pose significant threats to urban populations, where dense infrastructure and limited escape routes make timely evacuation both critical and challenging. In recent events, the absence of efficient search and evacuation mechanisms has led to substantial injuries and loss of life [1]. Rapid and informed responses are essential in such scenarios, especially when individuals are operating under extreme stress and limited situational awareness.

Unmanned Aerial Vehicles (UAVs) and robotic systems have emerged as promising tools for disaster response due to their affordability, mobility, and ability to operate in dangerous conditions without risking human lives. UAVs are already used for tasks such as aerial mapping, environmental monitoring, and damage assessment. However, their application in active rescue and evacuation efforts remains limited [2]. Drones have the potential to play a far more proactive role: locating and tracking individuals, communicating directions, reducing panic through real-time interaction, and alerting rescue teams to those left behind.

Motivated by this potential, we study the following question:

How can a team of autonomous agents coordinate to search for and guide humans experiencing panic and stress during fire evacuation?

We consider a dynamic urban environment in which a fire originates at a location and spreads over time. Our study focuses on a simplified setting involving a single human evacuee attempting to reach a designated safe zone. This problem presents several challenges:

- **Human behavior under panic:** Empirical and psychological studies show that panic and stress significantly impair rational decision-making in disaster scenarios [3]–[5].
- **Partial observability:** The UAV team does not initially know the evacuee's location. Urban environments often restrict visibility due to occlusions from buildings and other structures.

To address these challenges, we propose a novel team-based coordination strategy using two UAVs with asymmetric capabilities:

- The **High-Level Rescuer (HLR)** operates at a higher altitude to gain broad situational awareness and estimate the probable location of the human.
- The **Low-Level Rescuer (LLR)** navigates closer to the ground, allowing it to detect occluded regions, avoid obstacles, and interact with the evacuee to guide them to safety.

To model the evacuee, we adopt an agent-based modeling (ABM) approach that captures individualized human behavior influenced by local information and stress responses. Specifically, we incorporate a panic-based behavioral model grounded in empirical findings from Trivedi et al. [6], where motion is governed by a normalized average of panic stimuli. These stimuli are affected by the evacuee's distance from a safe zone, velocity alignment with nearby individuals, and visual triggers such as fire or injured people. This model allows us to simulate how panic may cause deviations from otherwise rational evacuation paths.

On the UAV side, we employ deep reinforcement learning to learn coordination strategies in a partially observable environment. Our framework trains a single policy to control asymmetric agents with different capabilities and objectives. To reason under uncertainty, we use recurrent neural network-based policies that leverage observation histories. We carefully design reward functions to balance exploration and exploitation during training, and use Proximal Policy Optimization (PPO) to train agents that can effectively assist humans during evacuation.

To evaluate our policies, we create testing environments with slight randomization in the start and end positions of the human evacuee, as well as the fire start locations. We observe that when the evacuees' start and end locations are the same or similar, the policy performs very well regardless of the fire start locations. We also find that human panic plays a significant role in guiding evacuees out of dangerous situations. When a person panics or fails to notice the UAV that can lead them to safety, they significantly increase their time spent in hazardous zones.

II. Related Works

Unmanned Aerial Vehicles (UAVs) and ground robots have been increasingly deployed for humanitarian assistance and disaster response, yet their use in real-time search and evacuation operations remains limited [2]. Much of the existing literature focuses on post-disaster tasks such as aerial mapping and structural assessment [7], [8], with relatively little attention to active human guidance during evacuations. One example of robot-assisted evacuation is the work of Nayyar et al. [9], where tele-operated ground robots were used to guide individuals in smoke-filled indoor environments. While this demonstrated the feasibility of robotic guidance, it relied on assumptions of proximity and prior knowledge of human location, limiting its applicability in more uncertain, outdoor, and dynamic environments.

Within robot-guided evacuation strategies, "shepherding"—where robots actively lead evacuees to exits—has been shown to outperform strategies where control is handed off between robots at key decision points [10]. However, such strategies require that robots first locate evacuees under uncertain conditions, including limited visibility and obstructed urban terrain. To address this, the literature on collaborative multi-robot systems offers several search techniques based on distributed exploration and information sharing [8], [11]–[13]. These methods can improve coverage and redundancy, thereby increasing the likelihood of locating distressed individuals in time. Yet, much of this work fails to fully integrate search with real-time guidance, especially in scenarios involving human unpredictability.

Finally, there is growing recognition that autonomous capabilities are crucial for robot deployment in disaster scenarios, where communication infrastructure may be unreliable or absent. Many systems still rely heavily on teleoperation [9], [14], which not only strains human operators but is also susceptible to failure under stress or scale [15], [16]. Autonomous systems with onboard sensing, perception, and decision-making can offer more resilient and scalable solutions. However, autonomy must be paired with coordinated behavior across teams and with models that account for human panic and behavioral variability.

These gaps highlight the need for generalizable, autonomous, and coordinated multi-robot systems that can search for, reason about, and guide human evacuees in real-time. Our work builds on these insights by combining agent-based human modeling, deep reinforcement learning, and asymmetric drone roles to enable robust evacuation support in partially observable, dynamic environments.

III. Problem Formulation

We consider an urban environment affected by fire, involving three agents: a human agent (referred to as the *evacuee*), who aims to reach a designated safe zone while avoiding the fire, and a team of UAVs (referred to as *rescuers*), whose objective is to locate and guide the evacuee to safety. The rescuers operate collaboratively, with one responsible for high-level surveillance and the other for ground-level interception. We describe each rescuer below:

1. **High-Level Rescuer (HLR):** Operates at high altitude with a wide field of view. It maps the environment, infers the evacuee's location, and tracks their movement using

a downward-facing camera. However, it cannot physically intercept the evacuee.

2. **Low-Level Rescuer (LLR):** Operates at low altitude with a narrower field of view. It is the only agent capable of physically intercepting the evacuee. The LLR navigates closer to the ground and detects obstacles that may be hidden from the HLR, using a front-facing camera.

We model the urban environment as a discrete-time, 2D grid world, represented by a tuple $\mathcal{G} = (\mathcal{X}, \mathcal{A}, \mathcal{O}, \mathcal{B})$, as shown in Fig. 1. Our grid design extends the base environment introduced in [17] by incorporating a dynamic fire model where:

- $\mathcal{X} \subseteq \mathbb{Z}^2$ denotes the set of grid cells, divided into accessible and inaccessible regions.
- $\mathcal{A}$ defines the set of possible actions (e.g., move up / down / left / right, wait).
- $\mathcal{O}$ encodes the local observations available to each agent (e.g., visibility, fire, evacuee location).
- $\mathcal{B}_t \subset \mathcal{X}$ denotes the subset of cells affected by fire at time t.

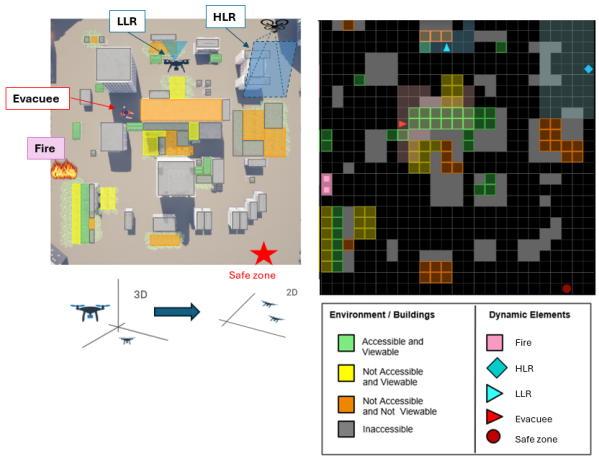

Fig. 1. **Left:** A 3D urban environment from a top-down perspective illustrating a disaster evacuation scenario involving a human evacuee, fire, and two UAV rescuers. **Right:** The same environment is modeled as a 2D grid-based world, where each cell is annotated with accessibility and visibility properties.

Fire propagates over time using a simple stochastic model and can be adapted to other models[1]. At each time step t, each burning cell $(x, y) \in \mathcal{B}_t$ may ignite one of its adjacent neighbors $(x', y') \in \mathcal{N}(x, y)$, where $\mathcal{N}(x, y)$ denotes the neighborhood of the cell (x, y). Each neighbor cell (x', y') ignites independently with probability p_{fire}, provided that it is within the map limits and not already burning. The probability that a given cell ignites is defined as

$$\mathbb{P}\big((x', y') \in \mathcal{B}_{t+1} \,\big|\, (x', y') \notin \mathcal{B}_t\big) = 1 - \prod_{(x,y)\in\mathcal{N}(x',y')} \big(1 - \mathbf{1}_{\{(x,y)\in\mathcal{B}_t\}} \cdot p_{\text{fire}}\big)\,. \quad (1)$$

Each cell may contain static obstacles (e.g., walls, buildings, trees) or dynamic entities (e.g., agents, fire, smoke). The environment transitions based on both agent actions and environmental dynamics. Static obstacles are classified into four types based on two agent-specific properties: accessibility[2] and viewability[3]. These properties vary across the HLR, LLR, and the evacuee:

- **Type I:** Inaccessible and non-viewable by all agents (e.g., solid walls or opaque buildings).
- **Type II:** Inaccessible and non-viewable by the HLR, but accessible and viewable by the LLR and the Evacuee (e.g., tree rows, tall canopies, or partially open structures).
- **Type III:** Accessible and viewable by the Evacuee; viewable but inaccessible by the LLR (e.g., narrow alleys or dense urban corridors).
- **Type IV:** Accessible and viewable by the Evacuee, but both inaccessible and non-viewable by the LLR and HLR (e.g., enclosed indoor areas).

Obstacles of Types I–IV occlude an agent's field of view when not viewable by that agent type. In Fig. 1, obstacle types include: Type I (gray, inaccessible), Type II (orange, not accessible and not viewable), Type III (yellow, not accessible but viewable), and Type IV (green, accessible and viewable). Fire sources (pink tiles) dynamically spread through the environment. The human (red triangle) must reach the designated safe zone (red circle in 2D, red star in 3D) while avoiding fire and evading two UAVs: a high-level rescuer (HLR, blue diamond) and a low-level rescuer (LLR, blue triangle). Shaded regions around each UAV indicate their respective fields of view (FOV).

The HLR and LLR share their positions, headings, fire observations, and detected information about evacuees. The HLR communicates any sightings to the LLR to enable coordinated interception. The objective is to intercept the evacuee as quickly as possible.

The evacuee follows a behavior model shaped by panic, uncertainty, and environmental risk, including uncertain responses such as hesitation or rerouting. This creates a dynamic and partially observable environment requiring adaptive strategies from the UAV agents.

In this work, we address the following key challenges:

C1 Human Behavior under Panic. The rescuers must be able to search and intercept evacuee, especially because

[1]Even though we assume this simplistic fire model, our design approach is modular and can incorporate other complex fire models as well.

[2]An obstacle is accessible if an agent can occupy its cell. The HLR is unaffected by accessibility, as it flies above all structures.

[3]An obstacle is viewable if an agent can detect another agent through it when within its FOV.

evacuees can act irrationally under stress with actions influenced by panic, limited information, and environmental signals.

C2 Long-Term Planning under Uncertainty. UAVs must plan over long horizons despite occlusions and evolving conditions to gather information, track, and intercept the evacuee. This requires modeling a partially observable environment in which the rescuer only relies on their observations to plan their actions.

C3 Time-varying environment. The fire evolves stochastically, and its behavior is unknown to all agents, influencing both agent behavior and human decision-making. UAVs must adapt to changing fire conditions with limited foresight.

C4 Collaborative Multi-Agent Coordination: rescuers must coordinate their actions and share partial observations to track and intercept the evacuee effectively. Previous work has shown that naive and uncoordinated deployments of UAVs often fail to guarantee successful search and rescue outcomes in complex environments [2]. Coordinated planning and communication are essential to achieving critical time intervention in dynamic and uncertain settings.

C5 Heterogeneous Agent Capabilities. The HLR and LLR differ in observation and action modules. Coordinating asymmetric agents poses challenges in policy learning.

C6 Unknown evacuee location. The initial position of the evacuee is unknown to both the HLR and LLR agents, requiring them to actively search the environment.

IV. Approach

In this section, we detail the modeling components of our framework and explain how we address challenges **(C1)-(C6)**.

A. Agent-Based Modeling with Panic Behavior

To address **C1**, we implement an agent-based model where a single human agent (the evacuee) interacts with a dynamic environment and makes decisions influenced by cognitive and emotional factors. We incorporate panic as a key psychological variable, following the framework introduced by Trivedi and Rao [6], which is grounded in socio-psychological studies and empirical evidence.

The evacuee follows a rule-based policy with a predefined goal, simulating awareness of a potential safe zone. Under normal conditions, it plans movement using a truncated grid-based path. However, the panic parameter dynamically modulates behavior: elevated panic may override rational planning, leading to irrational responses such as freezing or erratic motion. For simplicity, we currently model social forces as an exogenous factor rather than through explicit agent-to-agent interactions.

The evacuee's observations include its position, goal, fire visibility, and last known rescuer locations, which feed into both panic updates and path decisions. At each time step t, the panic stimulus $\gamma(t) \in [0, 1]$ is computed using four normalized stimulus components:

$$\delta(t) = \frac{1}{4} \sum_{k=1}^{4} \delta_k(t),$$

where

$\delta_1(t) =$ distance to nearest visible exit;
$\delta_2(t) =$ misalignment with neighbors' velocity;
$\delta_3(t) =$ presence of nearby fire;
$\delta_4(t) =$ presence of nearby agents in discomfort.

The panic level is then smoothed over time:

$$\gamma(t) = \frac{1}{2}(\gamma(t-1) + \delta(t)).$$

The evacuee's velocity vector is then updated as follows:

$$v(t) = (1 - \gamma(t)) \cdot v_{\text{optimal}} + \gamma(t) \cdot v_{\text{herd}},$$

where v_{optimal} is the velocity directed towards the goal, while v_{herd} aligns with the average heading of the local neighbors. Finally, the position is updated using:

$$p(t+1) = p(t) + v(t) \cdot \Delta t.$$

In our simulation, δ_1 is computed based on the distance to a known safe zone, assuming the evacuee has some knowledge of its location. δ_2 and δ_4 are modeled as random perturbations drawn from a normal distribution, introducing Gaussian noise at every timestep. δ_3 is deterministically set when fire is within the evacuee's field of view (FOV). The overall panic level is then computed as the average of δ_1 through δ_4.

B. Partial Observable Markov Decision Process (POMDP)

To address **C2-C3**, we model the problem as a POMDP multi-agent environment represented by the tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{O}, T, R, \gamma),$$

where $\mathcal{S}$ is the latent environment state (comprising the location of all agents, location of fire, field of view of all agents), including the true positions of all agents, fire spread, and the evacuee's internal panic level. This state is not directly accessible to the agents. $\mathcal{A} = \mathcal{A}^{\text{HLR}} \times \mathcal{A}^{\text{LLR}}$ is the joint action space and $\mathcal{O} = \mathcal{O}^{\text{HLR}} \times \mathcal{O}^{\text{LLR}}$ is the joint observation space, T is the stochastic transition function capturing UAV and fire dynamics, R is the reward function, and $\gamma \in (0, 1)$ is the discount factor. Each agent $i \in \{\text{HLR}, \text{LLR}\}$ selects an action in polar form, which defines a target point relative to its current position. The environment then computes an A^* path to that point and truncates it according to the agent's speed. The LLR's action space is defined as:

$$a_t = (r, \theta, \psi), \quad r \in [0, R],\ \theta, \psi \in [-\pi, \pi],$$

where (r, θ) defines the target point in polar coordinates within selection radius R, and ψ specifies the heading direction.

a) Observation Space.: At time t, each rescuer receives a partial observation of the environment. Observations for the HLR and LLR are defined as:

$$o_t^{\text{HLR}} = o_t^{\text{LLR}} = \Big\{\mathcal{F}_t^{\text{HLR}},\ p_t^{\text{HLR}},\ p_t^{\text{LLR}},\ \phi_t^{\text{LLR}},\ (p_t^{\text{evac}},\ \phi_t^{\text{evac}}) \cdot \mathbf{1}\left[p_t^{\text{evac}} \in \mathcal{F}_t^{\text{HLR}} \cup \mathcal{F}_t^{\text{LLR}}\right]\Big\},$$

where $\mathcal{F}_t^i \subseteq \mathcal{X}$ denotes the field of view (FOV) of agent $i \in \{\text{HLR}, \text{LLR}\}$ at time t, and p_t^i, ϕ_t^i are its position and orientation. Rescuers observe their own states, each other's positions, LLR's heading, and the evacuee's position p^{evac} and orientation ϕ^{evac}, when visible to either agent. Each agent maintains a history of past observations and actions

$$h_t^i = \{(o_{t'}^i, a_{t'-1}^i) \mid t' \le t\},$$

and acts according to a policy $\pi^i : h_t^i \mapsto a_t^i$, such that:

$$a_t^i = \pi^i(h_t^i).$$

Under partial observability, we train each policy π_θ^i using Proximal Policy Optimization (PPO) with recurrence, where h_t^i is encoded via an LSTM. Each policy is trained to maximize the expected cumulative reward:

$$J(\theta^{(i)}) = \mathbb{E}_{\pi_\theta^{(i)}}\left[\sum_{t=0}^{T} \gamma^t r_t^{(i)}\right].$$

1) Reward Design: To guide the LLR and HLR agents toward the evacuee, we implement a reward function that promotes visibility, proximity, and successful capture. The rescuers share a reward and are trained jointly to encourage coordination. The reward is computed as:

$$R_{\text{LLR/HLR}}(s, a, s') = \begin{cases} c_{capture} & \text{if } c_{capture} > 0 \\ \alpha(c_1 + c_2 + c_3 + c_4) & \text{otherwise,} \end{cases}$$

where:

- $c_1, c_2 = \mathbf{1}[\text{id}_{\text{evac}} \in \mathcal{F}_i]$: small reward if the evacuee is in LLR and HLR's FOV, respectively.
- $c_3 = \mathbf{1}[\text{evac seen in } s \text{ and } s'] \cdot (||\hat{p}_{\text{LLR}} - \hat{p}_{\text{evac}}||_2 - ||p_{\text{LLR}} - p_{\text{evac}}||_2)$: small reward for approaching the evacuee
- $c_4 \cdot t$: time-based penalty scaled by the current timestep t, where $c_4 < 0$.
- α: $\frac{1}{[\text{LLR Max Speed}]}$: scaling coefficient for speed
- c_{capture}: large terminal reward for capturing the evacuee.

We set $c_1 = 1$, $c_2 = 1$, $c_3 = 1$, $c_4 = -0.05$, and $c_{\text{capture}} = 10$ in our simulations.

Using a POMDP and the reward structure, we encourage the agents to explore the environment when the evacuee's location is unknown. This addresses **C6**, as the rescuers must rely on their partial observations to locate the evacuee. The HLR learns to maximize aerial visibility, while the LLR navigates through accessible regions to increase ground-level coverage. The designed reward incentivizes coordinated movement that balances exploration and pursuit.

C. Centralized Multi-Agent Reinforcement Learning

To address **C4-C5**, we adopt a centralized training and execution framework from [17] in which the two rescuer agents share observations during both training and deployment. The team is controlled by a joint policy based on a recurrent neural network to leverage the history of observations, and is optimized using Proximal Policy Optimization (PPO). The choice of centralized training is motivated by the fact that the agents operate with asymmetric roles: the High-Level Rescuer (HLR) is responsible for wide-area search and tracking, while the Low-Level Rescuer (LLR) focuses on close-range interception. This heterogeneity introduces coordination and optimization challenges due to differing behavioral objectives. These factors make it difficult to attribute success or failure to individual actions, often leading to unstable policy updates and increased sample complexity during training. Thus, a joint policy utilizing each other's observations simplifies challenges **C4-C5** since each agent learns to coordinate with each other under the shared framework.

V. Analysis

We conducted simulations to investigate the following three research questions:

- **Q1** What is the effect of UAV support on an evacuee's ability to reach the safe zone more quickly? Specifically, we assume that once a UAV reaches the evacuee, the evacuee follows the UAV to the destination via an $A^\star$-computed path (i.e., a rational path).
- **Q2** How effective is the deep reinforcement learning based approach to learn a policy for UAVs?
- **Q3** How robust is our framework to variations in environmental factors, such as the start locations of the evacuee, the safe zone, and the fire's origin?

To address these questions, we consider a training environment in which the fire can randomly originate in one of four distinct locations spread around the map symmetrically, and the evacuee start location and the safe zone is spread around the perimeter of the environment with 40 different start-end pairs, forcing the evacuee to traverse all regions of the map during training. This information is initially unknown to the rescue team. Further details about the algorithm and training parameters are provided in Appendix A.

To address **Q1**, in Figure 2, we show three sample evacuee trajectories: (1) a baseline trajectory representing rational behavior without panic (green trajectory), (2) a trajectory where the evacuee always experiences panic (black trajectory), and (3) a trajectory where the evacuee initially panics but is intercepted by the LLR (denoted by red dot), which is then guided to the safe zone (blue trajectory). As expected, sustained panic increases the time required for the evacuee to reach the safe zone. Our autonomous UAV algorithms are designed to reduce the duration of panic by enabling early detection and interception. Once located, the UAVs guide the evacuee toward safety, reducing panic behavior.

To study **Q2** and **Q3**, we introduce the following four environments:

- **Env-I**: Identical to the training environment. The evacuee's start and destination locations, as well as the fire sources, remain fixed.
- **Env-II**: The evacuee's start and destination locations are randomly generated within a radius r of those in **Env-I**, while fire locations remain fixed as in **Env-I**.
- **Env-III**: The evacuee's start and destination locations are the same as in **Env-I**, but fire sources are randomly generated within a radius r of those in **Env-I**.
- **Env-IV**: Both the evacuee's start/destination locations and the fire sources are randomly generated within a radius r of the corresponding positions in **Env-I**.

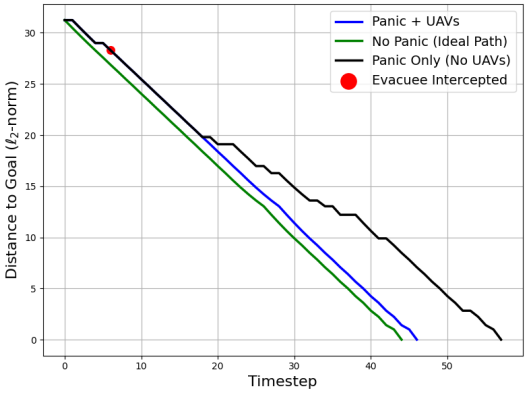

Fig. 2. Number of steps taken by the evacuee to reach the safe zone under varying panic levels

We evaluate the following performance metrics to assess the robustness and effectiveness of our approach:

- **Rescuer Capture Rate:** The percentage of trajectories in which the LLR successfully captures the evacuee before the episode ends.
- **Time to Capture:** The number of steps it takes for the rescuer team to intercept the evacuee.
- **First Seen:** The amount of time it takes for the evacuee to be seen by the rescuer team for the first time within a scenario.
- **Time in FOV:** The percentage of timesteps during which the evacuee is within the FOV of either rescuer. This reflects the rescuers' ability to track panic-influenced behavior. This metric also considers only trajectories where the evacuee is observed.

Table I shows that the rescuer team achieves its highest overall win rate in the environment it was the RL training objective, meaning it most effectively locates and intercepts the evacuee in that setting. Furthermore, the performance remains comparable across perturbed environments, demonstrating robustness to changes in initial conditions and addressing **Q3**.

Notably, in **Env-III**, the learned policy yields the best performance in terms of capture time, time to "first seen", and time spent in the rescuer's FOV. The location of the fire changes the trajectory of the evacuee, which leads to better performance in terms of these metrics.

Furthermore, we define **Percent Improvement** as the ratio of the difference between the time it takes for the evacuee to reach the safe zone without rescuers and with rescuers, to the difference between the time it takes for the evacuee to reach the safe zone without rescuers and that of the optimal $A^\star$ trajectory (evacuee with no panic). A higher value of this metric indicates that introducing the rescuer helps in evacuation.

Figure 3 shows the relationship between the capture time and the percent improvement when the UAV rescuers are deployed, evaluated in **Env-IV** with a capture radius $r = 5$. The plot includes only scenarios in which the evacuee was successfully captured, accounting for 77% of all cases. We observe that, in most of these captured instances, the percent improvement exceeds 60%, showing the effectiveness of the rescuers in aiding evacuation.

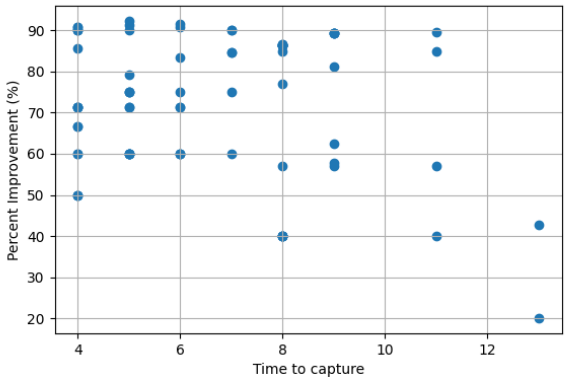

Fig. 3. Time to capture vs. percent improvement in evacuation time

Next, we study the robustness of our approach by varying the values of r in **Env-IV**. Figure 4 compares two key metrics: (1) the percentage of episodes in which the rescuers successfully intercepted the evacuee, and (2) the average percent improvement, computed over successful rescues. We vary r from 0 to 10. When $r = 10$, the evacuee can originate from any grid point and travel to any other grid point. As expected, the rescue rate decreases as r increases, since the trained policy encounters scenarios that deviate more from the training distribution. However, even when $r = 10$, the evacuee is rescued in over 40% of the runs. Moreover, the average percent improvement remains above 70%. These results suggest that the learned policy remains effective in diverse environments that deviate substantially from the training environment, highlighting the robustness of our approach. To further improve performance in such environments, one promising direction is to increase the diversity of origin-destination grid points and fire locations used during training.

TABLE I
POLICY PERFORMANCE SUMMARY

Env	Win Rate (%) ↑	Capture Time ↓	First Seen ↓	FOV Time (%) ↑
I (0, 0)	**71.0 ± 4.54**	13.20 ± 1.17	8.48 ± 0.97	30.77 ± 1.66
II (5, 0)	61.0 ± 4.88	12.28 ± 0.59	7.46 ± 0.35	31.60 ± 1.70
III (0, 5)	68.0 ± 4.66	**10.91 ± 0.51**	**6.75 ± 0.30**	**33.77 ± 1.83**
IV (5, 5)	66.0 ± 4.74	12.83 ± 1.17	8.77 ± 1.14	32.79 ± 1.88

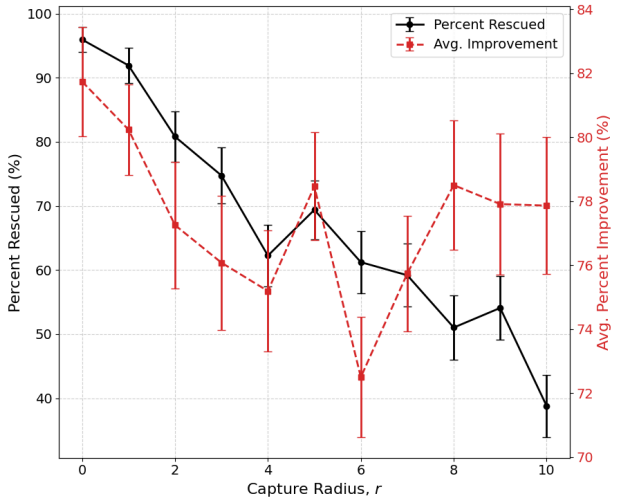

Fig. 4. Impact of capture radius r on evacuation outcomes. The left axis shows the percentage of evacuees successfully intercepted by UAV rescuers. The right axis shows the average percent improvement in capture time relative to the panic-only baseline.

VI. CONCLUSION

In this work, we proposed a novel framework for autonomous fire evacuation support using a coordinated team of UAVs operating in a partially observable and dynamically evolving urban environment. Our approach addresses several key challenges in real-world evacuation scenarios, including uncertainty in the evacuee's location, limited observability due to occlusions, and the impact of panic on human decision-making.

We introduced a team-based strategy comprising two UAVs with complementary roles: a high-level rescuer for wide-area search and a low-level rescuer for local guidance. To model realistic human behavior under stress, we leveraged an agent-based model that incorporates a panic-based motion heuristic, grounded in sociopsychological findings. We trained UAV policies using a recurrent deep reinforcement learning architecture to enable memory-driven planning in the absence of full state observability.

Our simulation results demonstrate that the use of UAV support significantly improves evacuation outcomes by reducing time to safety and compensating for suboptimal evacuee behavior under panic. Additionally, our recurrent policy architecture generalizes well across a range of environmental conditions, showing robustness to variations in fire origin, evacuee state, and map topology. Some limitations of our work include the assumption that the evacuee knows the location of the safe zone a priori, and that social forces influencing evacuee behavior are artificially introduced rather than explicitly modeled. Future work includes several promising directions, such as scaling to multi-human scenarios with N evacuees and M rescuers. While the current framework assumes $M = N + 1$, future extensions could explore more general team configurations and coordination strategies for $M \neq N$. Additional directions include integrating heterogeneous robot teams (e.g., UAVs and ground robots), and extending to 3D environments.

TABLE II
HYPERPARAMETERS (PPO)

Hyperparameter	Value
Frames per batch	1024
Sub-batch size	256
Number of epochs	10
Discount factor (γ)	0.99
GAE parameter (λ)	0.95
Clip range (ϵ)	0.2
Critic coefficient (c_0)	0.5
Entropy coefficient (c_1)	0.005
1-2 Optimizer	Adam
Learning rate	3×10^{-4}
Betas	$(0.9, 0.99)$
Eps	1×10^{-8}
Weight Decay	0

APPENDIX A
TRAINING PARAMETERS AND ALGORITHM

We provide a detailed description of the algorithm used in our implementation, along with various environments and algorithms.

A. Policy Optimization

To train the policies, we used Proximal Policy Optimization (PPO) with Recurrence for POMDPs, the exact variant of which we used is below, as applied to a POMDP.

In our implementation, we use a Tanh Normal distribution for the sampled action (line 12 in Algorithm 1), combined with action normalization.

B. Hyperparameters

Table II defines the values we used during training of the rescuer team policy.

Algorithm 1 PPO with Recurrence (POMDP)

Input: POMDP $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, T, R, Z, \rho_0 \rangle$
Output: Policy parameters θ for π_θ
Data: `total_timesteps`, `frames_per_batch`, `num_epochs`

Initialize θ, ϕ for π_θ, V_ϕ randomly;
`collected_timesteps` $\leftarrow 0$;
while `collected_timesteps` $<$ `total_timesteps` **do**
  $D \leftarrow \{\}$;
  `batch_collected_timesteps` $\leftarrow 0$;
  while `batch_collected_timesteps` $<$ `frames_per_batch` **do**
    $\tau \leftarrow \{\}$;
    Sample $s_0 \sim \rho_0$, $o_0 \sim Z(s_0)$;
    for $t \leftarrow 0$ to $T-1$ **do**
      $h_t \leftarrow [o_0, \ldots, o_t]$;
      $a_t \sim \pi_\theta(h_t)$;
      $s_{t+1} \sim T(s_t, a_t)$, $o_{t+1} \sim Z(s_{t+1})$;
      $r_t \leftarrow R(s_t, a_t, s_{t+1})$;
      $\tau \leftarrow \tau \cup (h_t, a_t, r_t)$ to τ;
      `batch_collected_timesteps` $\leftarrow$ `batch_collected_timesteps` $+1$;
    end for
    $D \leftarrow D \cup \tau$;
  end while
  $\theta' \leftarrow \theta$;
  for $i = 1$ to `num_epochs` **do**
    $\delta_t \leftarrow r_t + \gamma V_\phi(h_{t+1}) - V_\phi(h_t)$;
    $\hat{A}_t \leftarrow \sum_{l=0}^{T-t} (\gamma\lambda)^l \delta_{t+l}$;
    $\hat{V}_t \leftarrow \hat{A}_t + V_\phi(h_t)$;
    $e_t \leftarrow \text{entropy}(\pi_\theta(h_t))$;
    $\xi_t \leftarrow \frac{\pi_{\theta'}(a_t|h_t)}{\pi_\theta(a_t|h_t)}$;
    $L_t^{\text{CLIP}} \leftarrow \mathbb{E}_t \left[\min \left(\xi_t \hat{A}_t,\ \text{clip}(\xi_t,\ 1-\epsilon,\ 1+\epsilon) \hat{A}_t \right) \right]$
    $L^{VF} \leftarrow \mathbb{E}_t \left[\text{SmoothL1} \left(V_\phi(h_t),\ \hat{V}_t \right) \right]$
    $L \leftarrow L_t^{\text{CLIP}} - c_0 L_t^{VF} + c_1 \mathbb{E}_t[e_t]$;
  end for
  $\theta \leftarrow \theta'$;
end while
return θ

C. Environment Parameters

Table III defines the environment parameters for the experiments.

TABLE III
PARAMETERS (ENVIRONMENT)

Parameters	Value
LLR FOV size	5
HLR FOV size	9
LLR selection radius	10
HLR selection radius	10
LLR Max Speed	5

REFERENCES

[1] R. Hoffman, L. Sarnoff, M. Kekatos, and W. Mansell, "La fires aftermath: How people are rebuilding after losing almost everything," Mar 2025.

[2] M. Erdelj, E. Natalizio, K. R. Chowdhury, and I. F. Akyildiz, "Help from the sky: Leveraging uavs for disaster management," *IEEE Pervasive Computing*, vol. 16, no. 1, pp. 24–32, 2017.

[3] R. Y. Aldahlawi, V. Akbari, and G. Lawson, "A systematic review of methodologies for human behavior modelling and routing optimization in large-scale evacuation planning," *International Journal of Disaster Risk Reduction*, vol. 110, p. 104638, 2024.

[4] E. Bakhshian and B. Martinez-Pastor, "Evaluating human behaviour during a disaster evacuation process: A literature review," *Journal of Traffic and Transportation Engineering (English Edition)*, vol. 10, no. 4, pp. 485–507, 2023.

[5] Y. Xenidis and G. Kaltsidi, "Prediction of humans' behaviors during a disaster: The behavioral pattern during disaster indicator (bpdi)," *Safety Science*, vol. 152, p. 105773, 2022.

[6] A. Trivedi and S. Rao, "Agent-based modeling of emergency evacuations considering human panic behavior," *IEEE Transactions on Computational Social Systems*, vol. 5, no. 1, pp. 277–288, 2018.

[7] H. Bendea, P. Boccardo, S. Dequal, F. Giulio Tonolo, D. Marenchino, M. Piras *et al.*, "Low cost uav for post-disaster assessment," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 37, no. B8, pp. 1373–1379, 2008.

[8] G.-J. M. Kruijff, F. Pirri, M. Gianni, P. Papadakis, M. Pizzoli, A. Sinha, V. Tretyakov, T. Linder, E. Pianese, S. Corrao, F. Priori, S. Febrini, and S. Angeletti, "Rescue robots at earthquake-hit mirandola, italy: A field report," in *2012 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, 2012, pp. 1–8.

[9] M. Nayyar, G. Paik, Z. Yuan, T. Zheng, M. Zhu, H. Lin, and A. R. Wagner, "Learning evacuee models from robot-guided emergency evacuation experiments," in *arXiv preprint arXiv:2306.17824*, 2023.

[10] M. Nayyar and A. Wagner, "Effective robot evacuation strategies in emergencies," in *2019 28th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2019*, ser. 2019 28th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2019. United States: Institute of Electrical and Electronics Engineers Inc., Oct. 2019, 28th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2019 ; Conference date: 14-10-2019 Through 18-10-2019.

[11] I. Kruijff-Korbayová, F. Colas, M. Gianni, F. Pirri, J. de Greeff, K. Hindriks, M. Neerincx, P. Ögren, T. Svoboda, and R. Worst, "Tradr project: Long-term human-robot teaming for robot assisted disaster response," *KI-Künstliche Intelligenz*, vol. 29, pp. 193–201, 2015.

[12] V. A. Jorge, R. Granada, R. G. Maidana, D. A. Jurak, G. Heck, A. P. Negreiros, D. H. Dos Santos, L. M. Gonçalves, and A. M. Amory, "A survey on unmanned surface vehicles for disaster robotics: Main challenges and directions," *Sensors*, vol. 19, no. 3, p. 702, 2019.

[13] S. K. R. Moosavi, M. H. Zafar, and F. Sanfilippo, "Collaborative robots (cobots) for disaster risk resilience: a framework for swarm of snake robots in delivering first aid in emergency situations," *Frontiers in Robotics and AI*, vol. 11, p. 1362294, 2024.

[14] Y.-D. Kim, Y.-G. Kim, S.-H. Lee, J.-H. Kang, and J. An, "Portable fire evacuation guide robot system," in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 2789–2794.

[15] R. R. Murphy, S. Tadokoro, and A. Kleiner, "Disaster robotics," in *Springer handbook of robotics*. Springer, 2016, pp. 1577–1604.

[16] C. Nieto-Granda, J. G. Rogers III, and H. I. Christensen, "Coordination strategies for multi-robot exploration and mapping," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 519–533, 2014.

[17] A. Kalanther, D. Bostwick, C. Maheshwari, and S. Sastry, "Evader-agnostic team-based pursuit strategies in partially-observable environments," 2025, in submission to conference.