Autonomous Navigation in Swarm of UAVs Using Spatio Temporal Data and Constrained-Reinforcement Learning

Abhudaya Shrivastava(✉), Christos Petridis, Marijana Vacic, and Zoran Obradovic

Department of Computer and Information Sciences,
Temple University, Philadelphia, PA 19122, USA
{abhudaya.shrivastava,zoran.obradovic}@temple.edu

Abstract. Obstacle avoidance in aerial robotics remains a critical challenge, particularly in environments with uncertain terrain and weather conditions. This study introduces a Constrained Q-Learning model that leverages spatio-temporal data from LiDAR and OctoMap to achieve Zero-Shot Execution (ZSE) for autonomous Unmanned Aerial Vehicle (UAV) navigation in unseen environments, eliminating the need for iterative training. Experimental evaluations are conducted using a high-fidelity simulator across three environments: random forests, clustered forests, and metropolitan areas, under varying obstacle densities and flight velocities. The proposed model demonstrates a 100% success rate, achieving average flight times of 45 s for slow velocities (below 2.5 m/s) and 34 s for fast velocities (above 2.5 m/s). Comparative analysis with 90 Human-to-Computer (HTC) flights (slow and fast velocities), conducted by three pilots under identical conditions, shows the proposed model reduces flight time by 33% (1.5 times faster) while enhancing path optimization. Additionally, the model matches the path selection efficiency of a standard Q-Learning approach without requiring iterative training, highlighting its robustness and scalability for autonomous UAV navigation in complex environments and GPS-denied locations.

Keywords: Reinforcement Learning (RL) · Constrained Reinforcement Learning (CRL) · Zero-Shot Execution (ZSE) · Human-To-Comuter (HTC) · Human Computer Interaction(HCI) · Autonomous navigation · Computer Vision (CV) · OctoMap

1 Introduction

Aerial robotics, driven by UAV advancements, has expanded into applications like disaster response, environmental monitoring, and urban planning. UAVs leverage spatio-temporal data for enhanced navigation, optimizing flight paths in complex environments. Reinforcement Learning (RL) further improves autonomy by enabling UAVs to adapt in real-time, mitigating the impact of noisy or incomplete sensor data.

L. Iliadis et al. (Eds.): EANN 2025, CCIS 2582, pp. 54–67, 2025.
https://doi.org/10.1007/978-3-031-96199-1_5

This paper introduces a Constrained-RL (CRL) model for swarm UAV path planning and obstacle avoidance using computer vision (3D LiDAR and Octants) to obtain obstacle locations in various environments. The CRL model integrates Q-learning with heuristic constraints, resulting in zero-shot execution (ZSE), which optimizes the swarm UAVs' flight time and path selection to reach the destination without requiring training. Furthermore, a comparative analysis highlights the model's advantages in path optimization, collision avoidance, and flight time efficiency over state-of-the-art RL models and the Human-To-Computer (HTC) approach based on Human-Computer Interaction (HCI).

The proposed framework enables ZSE, reducing computational overhead while improving real-time decision-making due to ZSE-CRL framework. This can be used as an advantage in various applications including but not limited to disaster response, fire detection, and pollution monitoring, where rapid and adaptive UAV navigation is critical, often with limited prior data, making traditional navigation models inefficient and less adaptable [1]. By eliminating dependence on iterative training, this approach outperforms existing RL and human-to-computer (HTC) approaches, ensuring greater reliability, efficiency, and adaptability in unseen environments.

2 Related Work

Autonomous UAV navigation in complex and dynamic environments has gained significant attention, driven by advancements in multi-agent reinforcement learning (MARL), deep reinforcement learning (DRL), and evolutionary algorithms. Shi et al. [2] proposed a MARL approach with Adaptive Dimensionality Reduction (ADR) to enhance spatial data processing and coordination. Sheng et al. [3] improved DRL-based navigation with dynamic reward functions and a novel state-space representation, while Meng et al. [4] introduced a Cooperative Co-Evolutionary Algorithm (CCEA) for scalable swarm operations. Wang et al. [5] addressed real-time collision-free navigation with the APPA-3D algorithm, and Zhang et al. [6] optimized UAV routing in Mobile Edge Computing (MEC) environments.

Chenyuan He et al. [7] developed a spatiotemporal data-driven framework for UAV path planning in wind fields, significantly reducing decision time. Prior studies, including Chao Yan et al. [8], Bingze et al. [9], and Ronglei Xie et al. [10], achieved navigation improvements but required extensive training episodes or were constrained to specific scenarios.

This paper introduces the Zero Shot Execution-Constrained Reinforcement Learning (ZSE-CRL) framework, which integrates Q-learning with constraint-aware heuristic planning to enhance real-time UAV navigation without iterative training. Unlike prior models, ZSE-CRL leverages spatiotemporal data obtained using 3D LiDAR and OctoMap for immediate decision-making, improving adaptability and computational efficiency in unseen environments. Comparative evaluations against HTC and RL approaches highlight ZSE-CRL model's superiority in flight time optimization, path accuracy, and decision-making efficiency, making it a robust solution for UAV swarm autonomous operations.

3 Material and Methods

This section outlines the real-time UAV swarm navigation framework, detailing methodologies for path planning, obstacle avoidance, and computational efficiency in complex and unseen environments.

3.1 Framework: Spatio-Temporal Data, OctoMap, and CRL

The framework combines spatio-temporal data and OctoMap to enhance CRL for UAV swarm autonomous navigation, enabling adaptability, optimized path planning, and efficient operation in complex environments (Fig. 1).

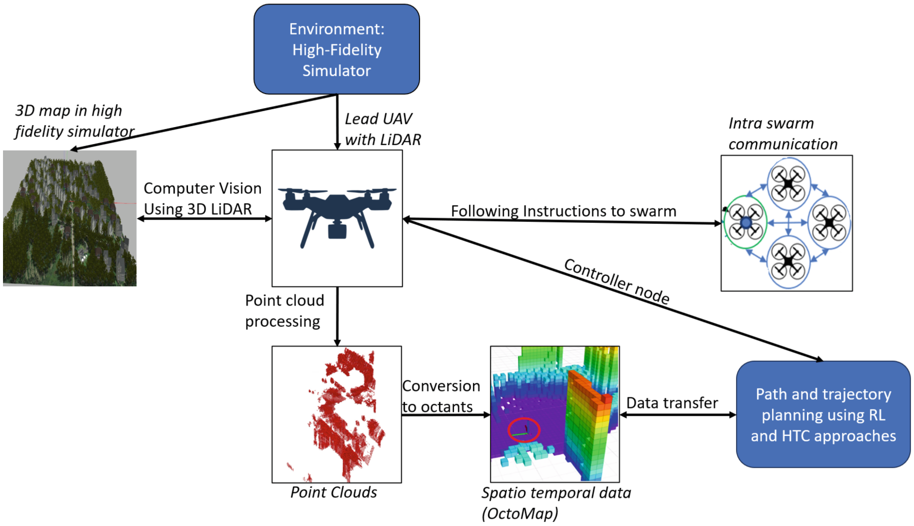

Environment: High-Fidelity Simulator
3D map in high fidelity simulator
Lead UAV with LiDAR
Intra swarm communication
Computer Vision Using 3D LiDAR
Following Instructions to swarm
Controller node
Point cloud processing
Conversion to octants
Data transfer
Path and trajectory planning using RL and HTC approaches
Point Clouds
Spatio temporal data (OctoMap)

Fig. 1. The framework utilizes spatio-temporal data and octo-mapping to develop a reinforcement learning algorithm for autonomous UAV swarm navigation.

Figure 1 shows the framework employed for autonomous navigation. The explanation of each part of the framework is as follows:

Geometry Synthesis and High-Fidelity Simulator: For high-fidelity simulation, 3D environment maps (Tables 1, and 2) are generated in Gazebo. The most common shapes used to construct 3D buildings and trees are shown below, along with their influence on the planned paths.

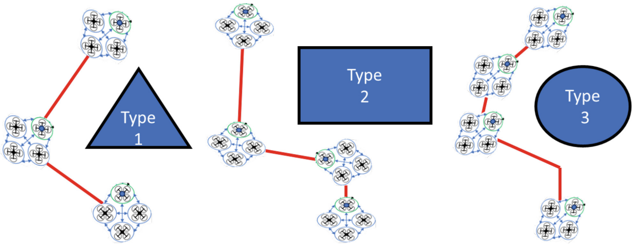


Fig. 2. Influence by shapes on path planning.

Figure 2 emphasizes the significance of spatial awareness in effective path planning, where the geometry of shapes affects the types of paths requiring need for geometry synthesis. This presents a challenge for real-time data processing and mapping in autonomous navigation using the ZSE-CRL model.

Computer Vision, Spatio-Temporal Data, and Path Planning: As shown in Fig. 1, obstacle point clouds are converted into octants to enhance computer vision for the swarm, optimizing path planning. This process is further detailed in Fig. 3.

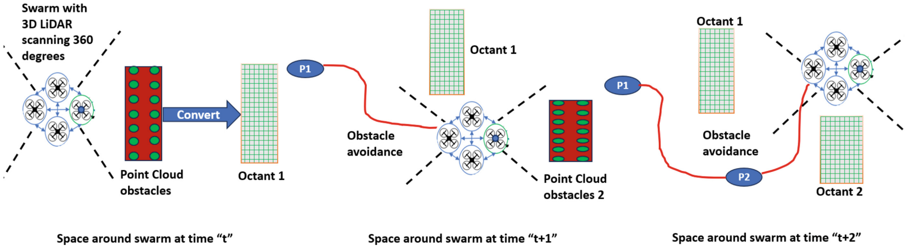

Swarm with 3D LiDAR scanning 360 degrees
Convert
Point Cloud obstacles
Octant 1
Space around swarm at time "t"
P1
Octant 1
Obstacle avoidance
Point Cloud obstacles 2
Space around swarm at time "t+1"
P1
Octant 1
Obstacle avoidance
P2
Octant 2
Space around swarm at time "t+2"

Fig. 3. Computer Vision: Lead UAV based octants mapping for obstacle avoidance and detection in different spaces and times.

Figure 3 illustrates the use of computer vision by octants mapping [21] instead of point clouds for UAVs internal mapping. The lead UAV performs a 360-degree LiDAR scan and employs a probabilistic approach to convert point clouds into a memory-efficient 3D grid-based OctoMap, as seen in octants 1 and 2. This enhances the efficiency of the ZSE-CRL model by enabling faster obstacle mapping and path computation from time t to t+2 [21]. The next section explores how optimized path planning using the ZSE-CRL framework complements real-time spatiotemporal data.

3.2 Spatio-Temporal Path Planning

Reinforcement Learning (RL) enables agents to optimize decision-making through trial and error, maximizing cumulative rewards. While traditional RL seeks an optimal policy without constraints, Constrained Reinforcement Learning (CRL) incorporates safety, performance, and resource limitations to ensure adherence to predefined boundaries. The ZSE-CRL framework enhances spatio-temporal awareness by utilizing real-time updates from the UAV's LiDAR-based OctoMap. As the lead UAV scans its surroundings, it converts point clouds into octants, improving memory efficiency and obstacle representation over time. These dynamically updated octants provide a structured probabilistic map, aiding informed decision-making for navigation [21]. By integrating spatial and temporal data, CRL enables UAVs to adapt their path planning in real-time while maintaining efficiency and constraint adherence. This ensures optimized trajectories and effective obstacle avoidance, enhancing autonomous navigation. The internal process works as:

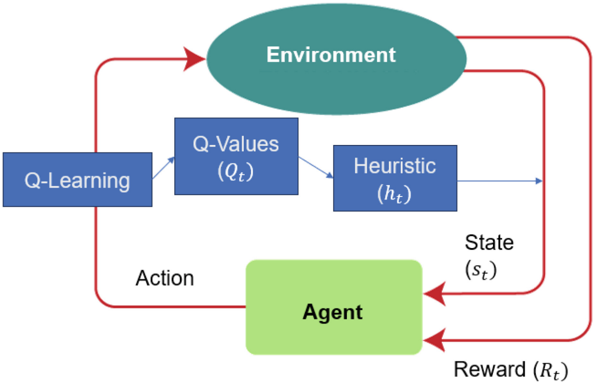


Fig. 4. The proposed reinforcement learning framework for autonomous navigation for a single UAV (one agent) and swarms of UAVs (multi-agent).

Figure 4 illustrates the CRL framework, which combines **Q-learning** with **heuristics**. Agents make decisions using **Q-values**, which are influenced by heuristic, and update their **state** at time t after each **action**. These state updates reflect the agent's position and environment at time t, influencing subsequent decisions. The agents then receive **rewards** based on their performance, which are used to further refine their decision-making process over time. This framework allows the agents to adapt dynamically to changes in their environment at each time step.

The key innovation of how heuristics work and produce a ZSE approach is explained and visualized using Fig. 5 below.

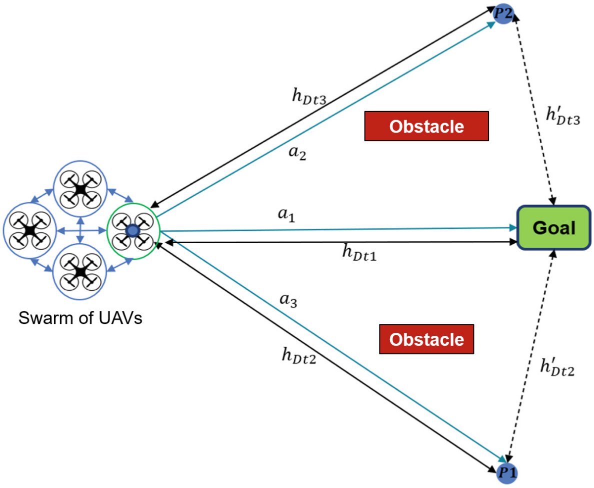


Fig. 5. Visualization of heuristic-based path selection, showing obstacles, points (P_1, P_2), heuristic distances at time step t ($h_{D_{t1}}$, $h_{D_{t2}}$, $h_{D_{t3}}$, $h'_{D_{t2}}$, $h'_{D_{t3}}$), and actions (a_1, a_2, a_3), highlighting the optimal path to the goal.

3.3 Zero Shot Execution (ZSE)

The heuristic at time t guides the swarm's decision-making by modeling path planning as nodes in a graph, evaluating distances D for each node as $h_{D_{t1}}$, $h_{D_{t2}}$, $h_{D_{t3}}$, $h'_{D_{t2}}$, and $h'_{D_{t3}}$ while avoiding obstacles P_1 and P_2. The swarm evaluates three actions: a_1, a_2, and a_3, with a_1 being optimal due to the shortest distance $h_{D_{t1}}$. Actions a_2 and a_3 lead via obstacles P_1 and P_2, respectively. In real-world scenarios, multiple minimum distances are evaluated based on spatiotemporal data obtained from computer vision, connecting different points to form the optimal path.

This method eliminates iterative training, ZSE with CRL, allowing the swarm to adapt autonomously in real-time autonomous navigation (Fig. 1). To explore the ZSE-CRL framework's potential, it is crucial to examine its computation in terms of action space, hyperparameter sensitivity, rewards, and policies.

3.4 Computation Under ZSE-CRL

Since the ZSE-CRL model is part of RL structure, the key configurations are:

- **Actions and Hyperparameters:** The ZSE-CRL model exhibited no sensitivity to hyperparameter changes due to its use of a greedy deterministic policy (discussed in the results section). The comparison RL model, hyperparameters selection followed experimentation within the range 0.0001 to 1.5, with the testing values chosen due to the negative impact of others on performance. The learning rate was tested at 0.0001, 0.001, and 0.01, while other hyperparameters were fixed: $\gamma = 0.95$, $\epsilon_{\text{init}} = 0.8$, $\epsilon_{\text{min}} = 0.05$, and $\epsilon_{\text{decay}} = 0.98$ (s). Actions were defined in eight cardinal and diagonal directions.

- **Rewards and Policies:** The reward system consists of multiple components to guide UAV navigation. **Destination Reward** grants rewards of +10 upon reaching the destination. **Move Penalty** assigns -1 if not exploring, while **Collision Penalty** applies -1 for obstacle collisions. Conversely, **No Collision Reward** grants +1 for safe movement. To promote vertical navigation, **Y-Coordinate Change Reward** provides +1 for changing the y-coordinate. The **Avoid Points Penalty** imposes -1 for remaining at the same x and y coordinates. Moving closer to the goal is +1 reward with **Closer to Goal Reward**, whereas **Penalty for Not Getting Closer** deducts -1. Lastly, **Bubble Rebound Penalty** assigns -1 for having an obstacle in the defined bubble radius also called a dynamic safety radius around UAV/UAVs (Figs. 1, 2, 3, and 4).

  Deriving from the general Q-learning equation, the ZSE-CRL approach incorporates a heuristic term, yielding the following result at time step t:

$$Q(S_t, a_t) = Q(S_t, a_t) + \alpha \cdot \left(r - \gamma \cdot \min_{a'_t} Q(S'_t, a'_t) - Q(S_t, a_t) \right) \tag{1}$$

  From Eq. 1 and Fig. 5, the heuristic is defined as:

$$\min Q_{(S'_t, a'_t)} = h_{D_t} \tag{2}$$

  Thus, as the distance to the goal increases, the Q-value decreases, leading to action selection based on:

$$a_{selected_t} = \text{argmax}_{a_t}(Q_{(S_t, a_t)}) \tag{3}$$

  Selecting the highest Q-value for action a_t in state S_t at time t ensures optimality, but obstacles and safety constraints may lead to invalid options. To address this, three policies are introduced to improve decision-making.:

  The **Rule-Based Policy with Constraint** filters actions based on distance to obstacles, ensuring $d(s', \text{obstacle}) > \text{threshold}$. The **Filtered Stochastic Policy** allows stochastic action selection from valid options using heuristics. The **Heuristic-Guided Deterministic Policy** selects the optimal action $a^* = \arg\max_{a \in A_{\text{valid}}} Q(s, a)$. The **Reactive Policy with Constraint (Bubble Rebound)** triggers a **Rebound Action** when the UAV's distance to an obstacle $d(s_t, \text{obstacle}) \leq r_{\text{safe}}$, enabling evasive maneuvers for collision avoidance of the swarm (Figs. 3 and 5).

3.5 Lead UAV and Intra-swarm Communication

The following section explains the leader-offset UAV communication for path computation, obstacle avoidance, and mapping. Using the Leader-Follower Model [14], the lead UAV communicates with Relay UAVs (three offset UAVs in this study), which then relay information to the follower UAVs, optimizing communication and reducing the load on the lead UAV for efficient swarm coordination [14].

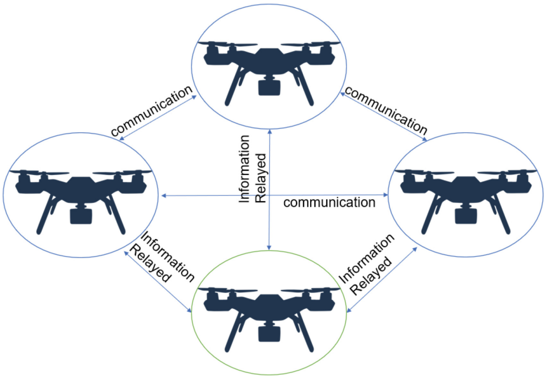


Fig. 6. A resilient robust network for communication between the UAVs in the swarm and relaying information from the lead to the offset UAVs.

As shown in Fig. 6. This communication framework also enhances fault tolerance by providing redundant communication paths and dynamic reassignment of relay roles, ensuring consistent updates in complex environments. The framework in the UAV swarm operates through a hierarchical structure for efficiency and scalability. The *Lead UAV* transmits critical state information to the selected *Relay UAVs* [15], which then forward it to neighboring *Follower UAVs* within their range [16]. This reduces the communication load on the lead UAV by avoiding direct communication with every follower. Redundancy is ensured through multiple *Relay UAVs*, enhancing fault tolerance as others can take over in case of failure [18]. Furthermore, the lead UAV can dynamically reassign relay roles as needed, optimizing mobility and system reliability [19]. This approach minimizes bandwidth usage and scales efficiently with larger swarm sizes [20].

4 Setup of the Study

The setup comprises of deploying UAVs type x500 in high fidelity simulator. Using the UAVs type x500 on described environments in the Table 1, and 2 for swarm of UAVs. The visualization of these environments is as follows:

Table 1. 3D Views of Random Forests maps

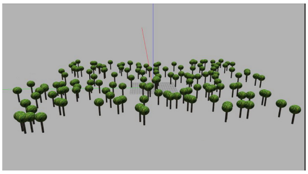
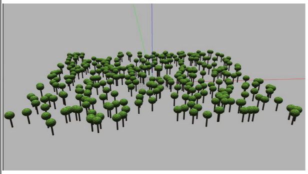
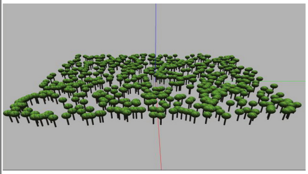
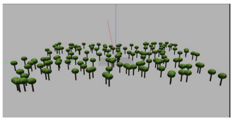
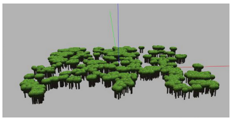
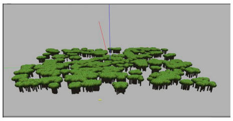
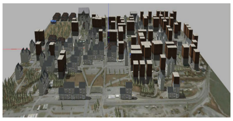
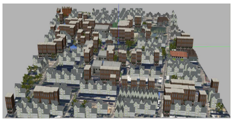
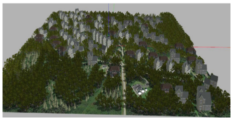
Low-Density	Medium-Density	High-Density
Random Forests	**Random Forests**	**Random Forests**
		

Table 2. 3D Views of Clustered Forests and Metropolitan maps

Low-Density	Medium-Density	High-Density
Clustered Forests 	**Clustered Forests** 	**Clustered Forests** 
Anchorage Downtown Alaska 	**North Central Philadelphia Pennsylvania** 	**Eagle River Suburbs Alaska** 

5 Results

The results are divided into two subsections: (5.1) Comparison of CRL and HTC approaches, considering psychological factors in decision-making, and (5.2) Evaluation of flight time optimization vs. cost, and the reliability of ZSE-CRL compared to state-of-the-art models.

5.1 ZSE-CRL and HTC Approaches

The ZSE-CRL is a non-iterative RL model for autonomous navigation, while the HTC approach involves human-computer interaction. To compare, three pilots performed 90 flights at slow (speed < 2.5 m/s) and fast (speed > 2.5 m/s) speeds across the environments in Tables 1 and 2. One example of decision-making from these flights is as follows:

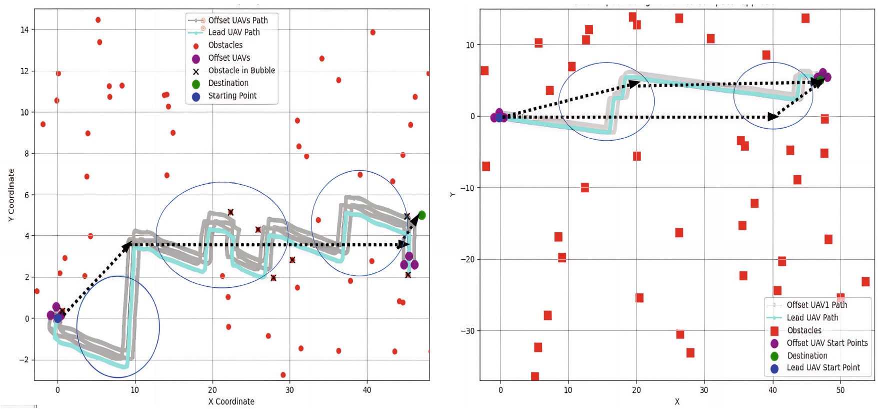
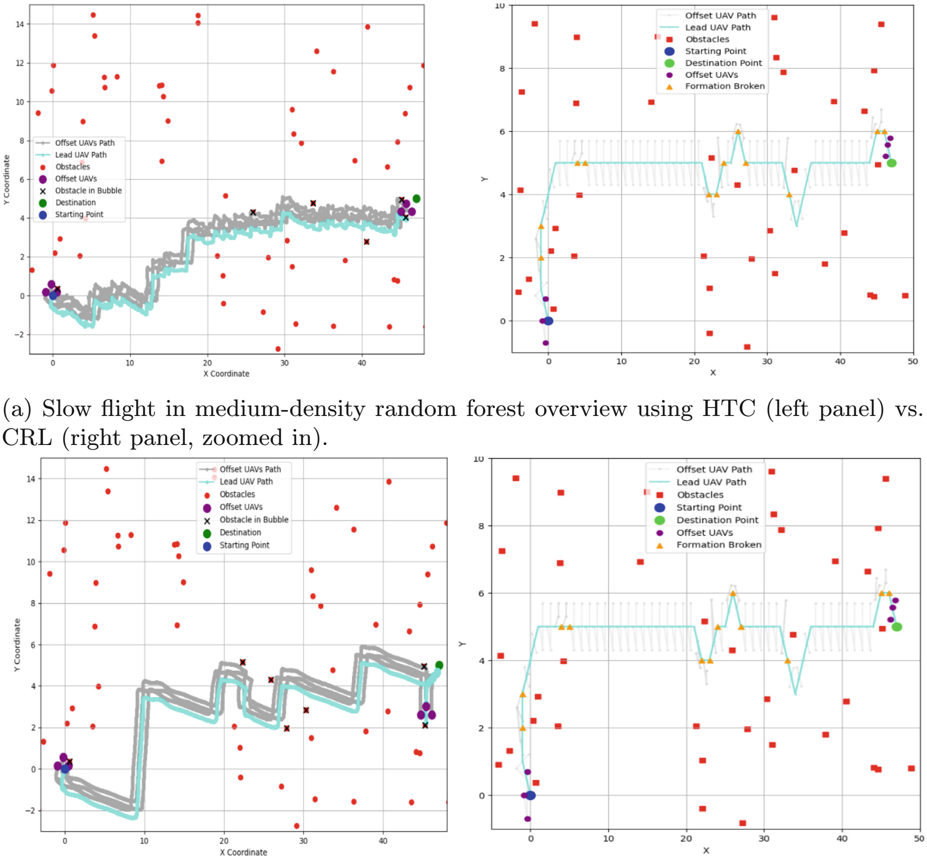


Fig. 7. HTC in random forest (left panel) and North Philadelphia (right panel).



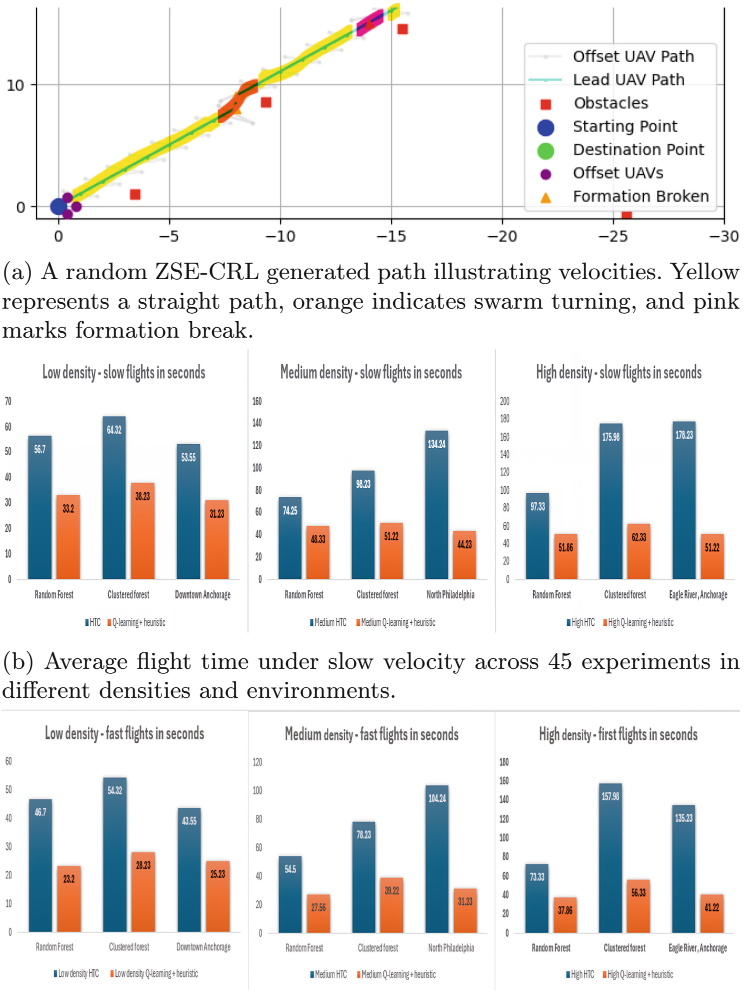
(a) Slow flight in medium-density random forest overview using HTC (left panel) vs. CRL (right panel, zoomed in).

(b) Fast flight in medium-density random forest overview using HTC (left panel) vs. CRL (right panel, zoomed in).

Fig. 8. Comparison of HTC and CRL in different environments and flight conditions.

Figure 7 illustrates the decision-making process in path selection, where the left and right panels reveal alternative, more optimal routes overlooked by pilots. When comparing the ZSE-CRL framework to the HTC approach (left panel), the decision-making strategy becomes evident. Figures 8a and 8b further emphasize the advantages of the ZSE-CRL model, which exhibits straight paths with sharp, well-defined turns, in contrast to the erratic turns observed in the HTC model. This trend remains consistent across various environments, demonstrating the ZSE-CRL's superiority in autonomous navigation and optimal path selection. A comparative analysis of flight time optimization with the HTC approach follows.

Time Optimization: The time optimization analysis of the ZSE-CRL model is explained and compared as described in the figures.



(a) A random ZSE-CRL generated path illustrating velocities. Yellow represents a straight path, orange indicates swarm turning, and pink marks formation break.

(b) Average flight time under slow velocity across 45 experiments in different densities and environments.

(c) Average flight time under fast velocity across 45 experiments in different densities and environments.

Fig. 9. Comparison of HTC (blue) and ZSE-CRL (orange) across different environments and flight conditions. (Color figure online)

To calculate the total flight time, we use the relationship between distance, velocity, and time, where time is the ratio of distance to velocity. Specifically, the time taken for a UAV to travel a given path is determined by:

$$\text{Time (t)} = \frac{\text{Distance (d)}}{\text{Velocity (V)}}$$

Figure 9a illustrates how straight paths maintain velocities of 2.0 m/s (fast) and 1.5 m/s (slow), while turns and formation breaks are assigned a velocity of 0.5 m/s. By applying this structured velocity assignment, the swarm is able to avoid high-speed banking collisions and internal interactions. Performance is then compared to the HTC approach vs the ZSE-CRL model.

Furthermore, as seen in Figs. 9b and 9c, the ZSE-CRL model (which combines Q-learning with heuristic) completes the flights in approximately 1.5 times less time than the HCI based HTC approach, showcasing its enhanced efficiency in path selection, decision-making, and overall flight time optimization.

5.2 ZSE-CRL Vs State-Of-The-Art Models

Based on the path selection and time optimization results from Figs. 8a, 8b, 9a, and 9b, ZSE-CRL excels in selecting optimal paths and optimizing flight time. To further validate its effectiveness, we compare the proposed model with an RL model using the framework in Figs. 1 and 6, as well as other popular approaches from similar case studies as described in related work section.

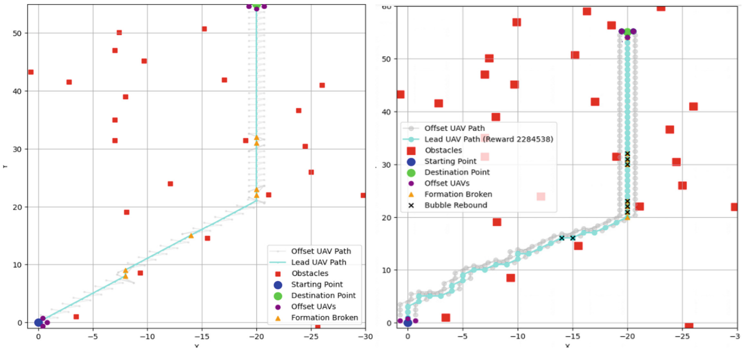


Fig. 10. Reliability for optimal path selection: ZSE-CRL model (left) vs. Q-Learning with 2500 learning episodes (right).

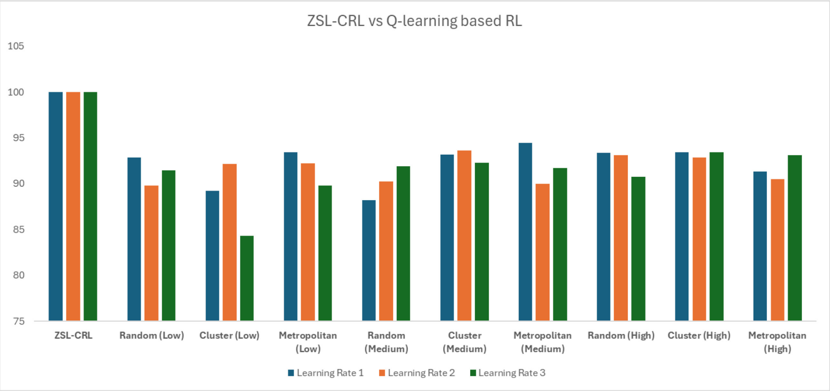


Fig. 11. Success rates for ZSE-CRL vs. Q-learning model (2500 learning episodes).

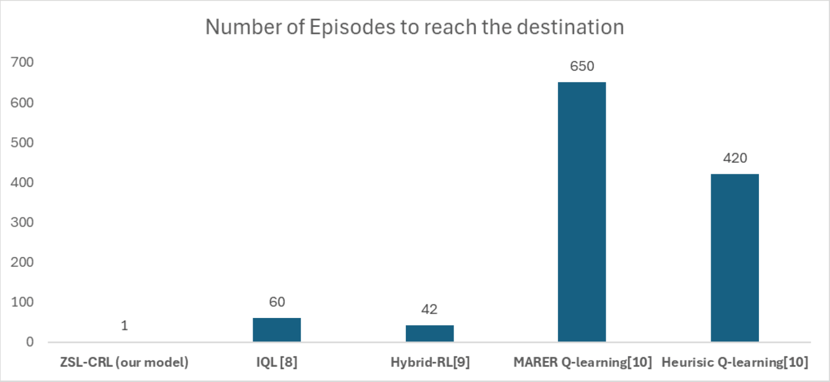


Fig. 12. ZSE-CRL: Number of episodes to reach the destination vs. other researched models.

Fig. 10 demonstrates that the ZSE-CRL model outperforms Q-learning by successfully completing the task in a single attempt with minimal formation breaks and a straighter path selection, eliminating the need for multiple learning episodes. This efficiency allows it to effectively avoid bubble rebounds and unnecessary turns. Figures 11 and 12 further validate its superiority, highlighting a 100% success rate and the ability to reach the goal within a single episode outperforming other comparable models.

6 Conclusion

The ZSE-CRL model, developed using Q-learning and heuristics, enables UAV swarms to autonomously navigate with 100% success rates. The study highlights the importance of environmental context in navigation and emphasizes the need for advancements in reinforcement learning and intra-swarm communication. A key shortcoming is the model's current limitation to static environments. Future work is focused on incorporating dynamic obstacles, environmental disturbances, and reducing reliance on leader UAVs. Expanding the model's adaptability will be essential as UAV swarm applications evolve in more complex terrains, which we are actively working towards.

Acknowledgments. This research was sponsored by the U. S. Army Engineer Research and Development Center (ERDC) and was accomplished under Cooperative Agreement Number W9132V-23-2-0002. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Engineer and Development Center (ERDC) or the U.S. Government.

References

1. Chen, X., et al.: Wildland fire detection and monitoring using a drone-collected RGB/IR image dataset. IEEE Access **10**, 121301–121317 (2022)
2. Shi, H., Zhao, Z., Chen, J., Zhou, M., Liu, Y.: Enhancing unmanned aerial vehicle path planning in multi-agent reinforcement learning through adaptive dimensionality reduction. Drones **8**, 521 (2024)
3. Sheng, Y., Liu, H., Li, J., Han, Q.: UAV autonomous navigation based on deep reinforcement learning in highly dynamic and high-density environments. Drones **8**, 516 (2024)
4. Meng, Q., Qu, Q., Chen, K., Yi, T.: Multi-UAV path planning based on cooperative co-evolutionary algorithms with adaptive decision variable selection. Drones **8**, 435 (2024)
5. Wang, J., Zhao, Z., Qu, J., et al.: APPA-3D: an autonomous 3D path planning algorithm for UAVs in unknown complex environments. Sci. Rep. **14**, 1231 (2024)
6. Zhang, S., Songwen, G., Zhou, Y., Shi, L., Jin, H.: Energy efficient resource allocation of IRS-assisted UAV network. ERA **32**(7), 4753 (2024)
7. He, C., Wan, Y., Xie, J.: Spatiotemporal scenario data-driven decision for the path planning of multiple UASs. SCOPE: ACM, pp. 7–12. NY, USA, New York (2019)
8. Yan, C., Xiaojia, X.: A path planning algorithm for UAV based on improved Q-learning. In: 2nd ICRAS 2018, pp. 1–5 (2018)
9. Xia, B., Mantegh, I., Xie, W.-F., Framework, H.: For UAV motion planning and obstacle avoidance: integrating deep reinforcement learning with fuzzy logic: CoDIT. Vallette, Malta 2024, 2662–2669 (2024)
10. Xie, R., Meng, Z., Zhou, Y., Ma, Y., Wu, Z.: Heuristic Q-learning based on experience replay for three-dimensional path planning of the unmanned aerial vehicle. Sci. Progress **103**(1) (2020)
11. Yan, L., Dai, J., Zhao, Y., Chen, C.: Real-Time 3D mapping in complex environments using a spinning actuated LiDAR system. Remote Sens. **15**, 963 (2023)
12. Fischer, C., Sester, M., Schön, S.: Spatio-temporal research data infrastructure in the context of autonomous driving. ISPRS Int. J. Geo-Inf. **9**, 626 (2020)
13. Hornung, A., Wurm, K.M., Bennewitz, M., et al.: OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton. Robot. **34**, 189–206 (2013)
14. Clerigues, D., Wubben, J., Calafate, C.T., et al.: Enabling resilient UAV swarms through multi-hop wireless communications. JWCN **2024**, 39 (2024)
15. Burdakov, O., Doherty, P., Holmberg, K., Kvarnström, J., Olsson, P.-M.: Relay positioning for unmanned aerial vehicle surveillance*. IJRR **29**(8), 1069–1087 (2010)
16. Tao, C., Liu, B.: Online autonomous motion control of communication-relay UAV with channel prediction in dynamic urban environments. Drones **8**, 771 (2024)
17. Zhang, H., Zhang, G., Yang, R., Feng, Z., He, W.: Resilient formation reconfiguration for leader-follower multi-UAVs. Appl. Sci. **13**, 7385 (2023)
18. Ning, N., Zhou, S., Bao, W., Li, X.: A study on the maximum reliability of multi-UAV cooperation relay systems. Sensors **24**, 2886 (2024)
19. Duan, T., Wang, W., Wang, T., Li, X.: Role assignment mechanism of unmanned swarm organization reconstruction based on the fourth directed motif. Sensors **22**, 8799 (2022)
20. Chen, X., Tang, J., Lao, S.: Review of unmanned aerial vehicle swarm communication architectures and routing protocols. Appl. Sci. **10**, 3661 (2020)
21. Baca, T., et al.: The MRS UAV system: pushing the frontiers of reproducible research real-world deployment, and education with autonomous unmanned aerial vehicles. J. Intell. Robot Syst. **102**, 26 (2021)