2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Explainable Artificial intelligence for Autonomous UAV Navigation

Didula Dissanayaka[1], Thumeera R. Wanasinghe[1] and Raymond G. Gosine[2]

Abstract— **Unmanned Aerial Vehicles (UAVs) with limited computational, perception and power resources face significant challenges when navigating autonomously in unfamiliar environments. While artificial intelligence (AI)-assisted algorithms have been used to address these limitations, transparency of the underlying AI models remains a concern, hindering user trust. To address this limitation, this research study proposes a novel, explainable AI-based navigation approach for UAVs to navigate them through unknown environments autonomously. The soft actor-critic (SAC) algorithm and multilayer perceptron (MLP) policies integrated deep reinforcement learning algorithm is developed to derive control actions. This controller is integrated with a novel moving-window gradient-based explainable artificial intelligence (XAI) framework to shed light on the UAV's decision-making process. The proposed XAI algorithm provides granular insights into how various factors, such as image segments and UAV state features, influence the UAV's actions. It lays the groundwork for a novel visual explanation approach that segments input depth images to highlight critical navigational cues, augmented by a dynamic color map for precise obstacle identification. Additionally, the study introduces comprehensive textual explanations to provide an in-depth understanding of the UAV's decision processes, thereby improving the model's transparency and explainability. The simulation results indicate that the proposed DRL model achieves over 95% success rate. Moreover, evaluations conducted in two distinct environments demonstrate the model's capability to generate effective and reliable explanations.**

I. INTRODUCTION

Deployment of unmanned aerial vehicles (UAVs) has emerged as an important technology in various applications ranging from goods delivery and surveying to emergency response. The ability to navigate autonomously in complex environments has become fundamental for the operational efficiency and effectiveness of these applications. Integration with artificial intelligence (AI) has transformed the landscape of autonomous UAV navigation, enabling UAVs to perform tasks with improved precision and autonomy [1]. Despite the significant progress in developing robust AI models for UAV navigation, particularly those leveraging deep reinforcement learning (DRL) [2], [3], the "black box" nature of such models often leaves users in the dark about how decisions are made. While DRL enables UAVs to autonomously navigate through challenging terrains by learning from interactions with the environment, the complex architectures of deep neural networks (DNNs) can obscure the logic behind their outputs. This complexity not only hampers user trust but also complicates the task of refining and debugging the AI system, prolonging broader acceptance [4], [5].

Consequently, the field of explainable AI (XAI) has gained attention from the autonomous UAV community. XAI algorithms aim to bridge the gap between AI's capabilities and human understanding, ensuring that users and developers can comprehend, trust, and effectively manage AI systems [5]. This is particularly crucial in safety-critical applications like UAV navigation, where understanding the rationale behind autonomous decisions — such as obstacle avoidance maneuvers — is essential for validation, troubleshooting, and iterative development. The need for explainability becomes even more pronounced as UAVs are deployed in increasingly complex and dynamic environments, where they must make split-second decisions that directly impact operational safety and effectiveness [6]. Despite these advantages and necessities, deploying XAI algorithms for autonomous UAV navigation is still in its infancy.

Therefore, our research primarily concentrates on the explainability aspect of AI in the context of UAV navigation. By integrating XAI principles into the development of UAV systems, we aim to clarify the decision-making processes of DRL-based reactive controllers, thereby enhancing transparency, trust, and explainability. The contribution of this work is as follows:

- A novel neural network model that employs SAC and MLP policies for improved accuracy and reduced computational complexity without resorting to conventional convolutional layers. This approach not only streamlines model architecture but also facilitates the extraction and improves the understanding of decision-making factors.
- A novel moving window gradient evaluation method that offers insights into how specific observations — including image segments, distance to the goal, velocity, and yaw metrics — influence the UAV's actions.
- A novel visual explanation technique which segments input images to pinpoint influential areas, enhanced by colormap for precise obstacle identification.
- Advanced textual explanation method that provides detailed rationales behind the UAV's maneuvers, adding a layer of explainability that goes beyond mere directional commands.

In emphasizing the explainability of AI in UAV navigation, our work not only seeks to establish seamless human-machine interaction, encouraging an environment of trust and collaboration but also enhance the performance and

[1]Didula Dissanayaka and Thumeera R. Wanasinghe are with the Faculty of Engineering Applied Science, Memorial University of Newfoundland, St.John's, NL, A1C 5S7, Canada ddissanayaka@mun.ca, thumeerawa@mun.ca

[2]Raymond G. Gosine is with the Faculty of Engineering Applied Science, Memorial University of Newfoundland, St.John's, NL, A1C 5S7, Canada and with the Innovation Policy Laboratory, Munk School of Global Affairs and Public Policy, University of Toronto, Toronto, ON M5S 3K7, Canada rgosine@mun.ca

efficiency of autonomous systems. This focus represents a pivotal step towards realizing the full potential of UAV technologies, ensuring that they can be deployed safely, effectively, and transparently in a wide range of applications.

II. BACKGROUND

A. DRL for UAV Navigation

DRL has been effectively utilized to solve complex navigation challenges by exhaustive simulations. [7] and [8] explore vision-based DRL approaches employing variations of the deep Q-network (DQN) for UAV obstacle avoidance and path planning. Additionally, work done in [9] presents a laser-based DRL strategy with deep deterministic policy gradient (DDPG) algorithms for indoor UAV path planning. This research work demonstrates the potential of DRL in improving UAV autonomous navigation.

DQN [10], proximal policy optimization (PPO) [11], DDPG [12] and soft actor-critic (SAC) [13] represent leading DRL algorithms, each offering unique benefits for decision-making. DQN introduces experience replay and target networks, making it effective in environments with discrete action spaces. PPO, a policy gradient method, emphasizes stable and efficient training through a novel objective function that mitigates the risk of harmful large policy updates. DDPG, aimed at continuous action spaces, lays the groundwork for SAC by combining off-policy data with an actor-critic framework, enhancing precision in control tasks. SAC further refines this by integrating entropy maximization into the reward structure, encouraging exploratory behavior and leading to more robust and adaptive strategies.

Policy architectures in DRL serve as the decision-making core of an agent, translating observation states into actions. To this end, multi-layer perceptrons (MLPs) provide a balance between complexity and performance, making them ideal for UAV navigation. Unlike more complex policy networks like convolutional neural networks (CNNs) or recurrent neural networks (RNNs), which excel in processing spatial and sequential data, respectively, MLPs provide a straightforward, fully connected network architecture [14]. This simplicity allows for faster training and inference, making it more suitable for scenarios where computational resources are limited and rapid decision-making is critical.

B. XAI

In the field of XAI, feature attribution methods are integral for explaining the decisions made by machine learning models. These methods are divided into general-purpose techniques, applicable across various models and data types, and direct visualization techniques, ideal for visual data. General-purpose methods like Shapley additive explanations (SHAP) [15], local interpretable model-agnostic explanations (LIME) [16], and gradient analysis [17] stand out for their versatility. Using game theory, SHAP quantifies each feature's impact on predictions, aiming for a fair attribution reflective of their true influence. LIME simplifies explanations by approximating the model around specific predictions, focusing on the impact of individual features. However, these methods can be computationally expensive [18]. Gradient analysis is particularly noted for its application in straightforward data formats, avoiding the complexities inherent in models like CNNs. Its computational efficiency is a significant advantage for applications with limited processing power. Even though variants like DeepSHAP [19] have been designed to be more efficient than their original SHAP counterpart, they still involve a more complex computation process than gradient-based techniques.

On the other hand, direct visualization techniques address the need for intuitive and immediate insights into models, especially those dealing with visual data like images. These techniques include methods like class activation mapping (CAM) [20], gradient-weighted-CAM (grad-CAM) [21] and saliency maps [22]. However, their application is primarily confined to visual data, and they need to provide comprehensive insights into non-visual inputs.

Moreover, general feature attribution methods can be extended to provide visualizations. [23] introduces a visual explanation method combining the SHAP feature attribution method and CNN. Despite its innovative integration, this method sometimes produces features that are not distinctly defined, requiring manual interpretation and adjustments to derive meaningful insights. Therefore, our work takes a distinct yet complementary path by leveraging gradient analysis for feature attribution, specifically applying it to segmented depth images. This approach enables the precise identification of obstacles. The proposed method automatically generates a color-map for visualization, directly identifying and highlighting obstacles. This auto-generation of distinct visual markers ensures clarity and immediacy in explaining the model's decisions.

III. DRL BASED RESPONSIVE UAV CONTROL APPROACH

The work presented in this study addresses the problem of UAV navigation in unknown environments using a DRL model that uses a responsive UAV controller. This approach models the UAV as an agent in an environment, framed within a Markov decision process (MDP). At each time step t, the agent in state S_t executes an action a_t transitioning to the next state S_{t+1} and earns a reward R_t. The SAC algorithm was employed to train the proposed DRL-based UAV navigation system. The SAC algorithm distinguishes itself by focusing not only on maximizing cumulative rewards, calculated as $R_t = \sum_{i=t}^{T} \gamma^{i-t} r(s_i, a_i)$ with γ as the discount factor, but also on increasing the policy's entropy. Policy entropy quantifies the degree of uncertainty associated with the current state. As the entropy value increases, the system tends to explore more extensively. This dual objective of reward maximization and entropy enhancement ensures that the UAV's decision-making process is robust and adaptable [13]. This approach allows the UAV to respond quickly to environmental changes while enhancing computational efficiency, a vital attribute for small UAVs with limited resources.

A. Problem Formation

This work considers a scenario where a UAV navigates in a three-dimensional environment. The observation space S_t at any time t consists of depth image features $S_{t_{\text{image}}}$ and UAV state features $S_{t_{\text{state}}}$. Depth image features are derived from raw images processed into a simplified vector of 25 features. This is achieved by dividing the image into a 5×5 grid, as shown in Figure 1, and extracting the pixel value of the closest object to the UAV within each grid section. The threshold for this extraction is set at $20\,\text{m}$ based on the ZED 2 depth camera's capabilities [24]. The UAV state features consist of the UAV's relative position in the $x-y$ plane $(d_{t_{xy}})$, the relative angle towards its destination (θ_t), its current planar velocity $(v_{t_{xy}})$, and its yaw rate (ω_t). The action space is defined by two actions: the planar velocity command (v_{xy}^{cmd}) and the yaw rate command (ω^{cmd}), which are determined by a policy network $(\Pi(S))$ that takes the UAV's state as input and outputs these actions for navigation.

B. Reward Function

To effectively achieve the dual objectives of the proposed navigation system — i.e., reaching a predefined goal and robust obstacle avoidance behavior — the reward function of the proposed DRL framework includes two distinct components: a continuous reward and constant reward values. This approach was necessary due to the expansive state space and complex dynamics encountered in a 3D environment, demanding an advanced model to facilitate efficient learning and reliable performance. The reward function used in this study is as follows:

$$R(S_t) = \begin{cases} 10, & \text{if goal reached} \\ -20, & \text{if crashed} \\ -10, & \text{if outside workspace} \\ R_c, & \text{otherwise} \end{cases} \tag{1}$$

In the proposed navigation system, a specific boundary for UAV operation is established, tailored to each environment. Corresponding boundary values for each environment will be provided in section IV. Deviations beyond this designated area incur a penalty of -10. Additionally, collisions with obstacles are penalized more severely, with a -20 penalty, underscoring the system's emphasis on safe navigation. Successfully reaching the goal is positively reinforced with a reward of 10 points. Furthermore, this model incorporates a continuous reward function (R_C) composed of multiple components to finely tune the UAV's behavior. These components include

- Distance reaching reward (R_{dist}): This component encourages the reduction of the distance to the goal location, reinforcing progress towards the goal.
- Position punishment (R_{pos}): Aimed at discouraging undesirable positional deviations, this penalty is crucial for maintaining course fidelity.
- Obstacle punishment (R_{obs}): This factor intensifies the consequence of approaching obstacles, promoting spatial awareness.
- Action punishment (R_{act}): Designed to optimize the efficiency of control actions, this component mitigates excessive or erratic maneuvers.
- Yaw error punishment (R_{yaw}): This element minimizes orientation deviation of the UAV, ensuring that it maintains proper heading relative to the target.

Each of these components plays a vital role in shaping the UAV's learning process, ensuring that the navigation strategy is not only goal-oriented but also safe, efficient, and aligned with operational constraints. R_C is defined as a weighted sum of these components, (2), where α_1 to α_5 are the weights given to each component.

$$R_c = \alpha_1 R_{\text{dist}} - \alpha_2 R_{\text{pos}} - \alpha_3 R_{\text{obs}} - \alpha_4 R_{\text{act}} - \alpha_5 R_{\text{yaw}} \tag{2}$$

The reward R_{dist} and punishments R_{pos} and R_{obs} calculated using (3)-(5), where $d(S_t)$ is the Euclidean distance from current position to goal position at time t and β and ι serve as parameters for the normalization of position and obstacle punishments, respectively. These values were determined through comprehensive simulations and in-depth analysis.

$$R_{\text{dist}} = \frac{\delta \cdot d(S_{t-1}) - d(S_t)}{d(S_0)} \tag{3}$$

$$R_{\text{pos}} = \max\left(0, \min\left(1, \frac{d(S_t) - d(S_0)}{\beta}\right)\right) \tag{4}$$

$$R_{\text{obs}} = 1 - \max\left(0, \min\left(1, \frac{d_{obs} - d_{crash}}{\iota}\right)\right) \tag{5}$$

DJI Matrice 300 quadcopter [25] is considered as the targeted UAV when selecting numerical parameters related to the UAV. R_{obs} is calculated only when the UAV is within $10\,\text{m}$ of an obstacle, and it is normalized to be between 0 and 1. This $10\,\text{m}$ threshold is set to align with $8\,\text{m/s}$ maximum velocity of the DJI Matrice 300 drone, ensuring adequate reaction time for safe maneuvering while accounting for sensor accuracy and computational delays. If the minimum distance to the obstacle is less than $1\,\text{m}$, it is considered crashed. Thus, parameter d_{crash} is set to $1\,\text{m}$. Considering the DJI Matrice 300 drone's dimensions of $81 \times 67 \times 43\,\text{cm}$, this crash threshold allows for a safety buffer around the drone, reflecting a realistic margin for avoiding physical contact given the drone's size. The action punishment is calculated as $R_{\text{act}} = |\omega_{\text{curr}}| / \omega_{\text{max}}$, where, ω_{curr} is the current yaw rate action and ω_{max} is the maximum allowable yaw rate change. In our training process, ω_{max} is set to $80°/s$ to match the DJI Matrice 300 specifications with a 20% safety margin with respect to its maximum yaw rate. The yaw error punishment is calculated as $R_{\text{yaw}} = |\omega_{\text{error}}| / 90$, where ω_{error} is the relative angle error between the agents heading and direction towards the goal. This error could be between $180°$ and $-180°$. However, the heading error is divided by 90 instead of 180 to intensify the impact of heading corrections by doubling the penalty, emphasizing the importance of alignment with the goal direction.

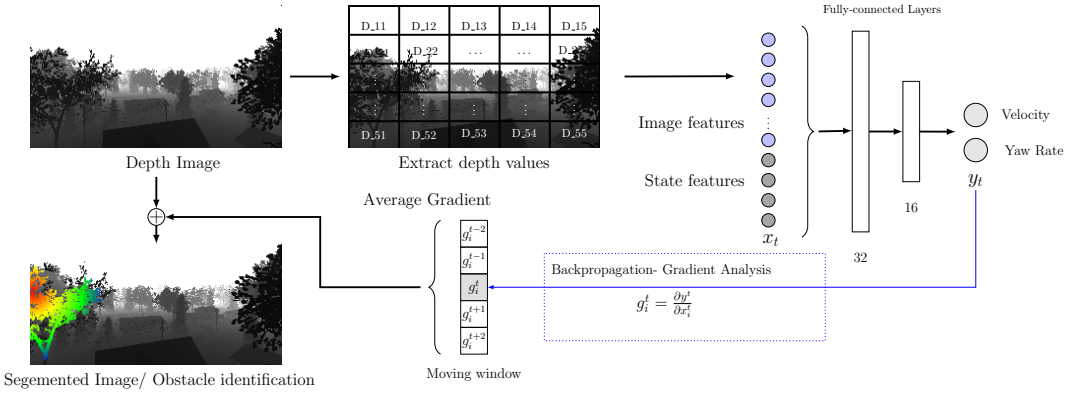


Fig. 1: System overview of the proposed method. The inputs are depth image and UAV states. The depth image is segmented into a 5×5 grid and the closest depth values from each image segment is extracted. These features are then fed to a fully-connected network to obtain two output control commands, velocity and yaw rate. Based on that, gradient values are calculated for each input feature. These gradient values are averaged over a 5 time step window to find three most influential features. If one of them is a image feature, corresponding pixel location is identified in the depth image and visualization is performed.

IV. EXPLANATION METHOD

In this work, as shown in Figure 1, a novel application of gradient analysis to enhance the explainability of UAV control actions is introduced. Gradient analysis computes the gradient of the output function with respect to each input feature, capturing how slight modifications in the input could affect the model's prediction. If $f(x)$ represents the output of a model for an input vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$, the significance of the k-th feature in influencing the model's output is quantified by the partial derivative $\partial f / \partial x_k$.

Additionally, while traditional methods focus on post-hoc explanations of entire iteration or single time step [23], [26], the proposed method employs a moving window technique to analyze decision-making processes involved in DRL-based autonomous UAV navigation. Providing a global, post-hoc explanation of a UAV's entire flight path can be overwhelming and less informative due to the aggregation of all actions. On the other hand, explaining control actions in each time step may lead to misinterpretations, as scattered variations or jitters in the UAV's actions might be misconstrued as deliberate maneuvers. Additionally, this approach could be computationally intensive and produce information at a rate too rapid for adequate human comprehension. Therefore, explanations are provided if the UAV has taken distinguishable actions like accelerating, decelerating or turning within a specific window. It allows for more dynamic and situation-specific explanations.

Our study sets the moving window size to 5 after considering temporal resolution, cognitive load and interpretability of a human, computational efficiency and empirical validation. The depth images are captured at 10 Hz; hence, the window size of 5 translates into 0.5 s window, making it a significant time frame in dynamic environments. This frequency matches human cognitive speeds [27] and balances computational demands with timely analysis. Moreover, empirical validation with exhaustive simulation trials indicates that the window size of 5 yields the most accurate and useful explanations in practice.

In this work, gradient values for all 29 observation states (25 depth image features and four UAV state features) are computed and averaged over the moving window. This average gradient evaluation, combined with UAV action values, is employed to generate both textual and visual explanations.

A. Textual Explanations

In the proposed approach, textual analysis of the action space and explanations for the particular action based on the gradient values of features are provided. As introduced previously, the proposed model features two continuous action outputs: the horizontal speed (v_{xy}^{cmd}) and the steering angular speed (ω^{cmd}). To analyze these actions, they are segmented into five distinct categories as shown in Figure 2. In this analysis, each time step in the window is examined to assess changes in velocity. Initially, the velocity at the beginning of the window is noted as a baseline. As we move through the window, we monitor the velocity at each step to determine whether it increases or decreases. For example, if the velocity consistently increases throughout the window for all time steps, it is determined as velocity increases. After that, the nature of the increase—whether gradual or rapid—is determined by comparing the velocity at the start of the window with the velocity at the end.

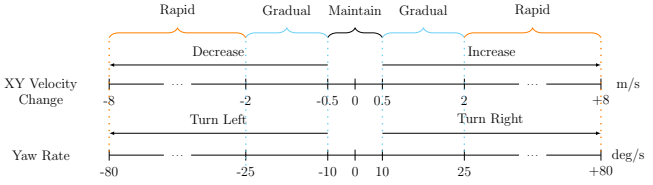


Fig. 2: Action space. Actions are divided into five categories.

Leveraging the Principle of Pareto Efficiency, which suggests that approximately 80% of effects come from 20% of the inputs [28], initially, six features were selected based on their gradient values to explain corresponding UAV actions. However, through empirical validation with the aim of resource optimization and computational efficiency, this was reduced to the three most influential features. If velocity increases or the yaw rate is positive, only the features with positive gradients are considered. Conversely, only negative gradients are considered when velocity decreases or the yaw rate is negative. Additionally, if one of the three influential factors is an image feature with a feature value greater than zero, it indicates the presence of an obstacle. In such cases, the textual explanation will include information on obstacle location. Notably, an image feature with a value of zero also carries significant meaning, indicating a vacant area where the UAV can safely traverse.

The state variables are also essential to the explanation model. If these state variables are among the top three influential factors in the gradient analysis, they are explicitly included in the textual explanation. This method enables an improved understanding of the UAV's actions by considering both temporal changes in action values and the most influential factors behind these actions. By combining continuous action output analysis with gradient values from both image and state features, this approach offers a comprehensive explanation of the UAV's decision-making process.

B. Visual Explanations

In addition to textual explanations, the proposed approach includes a unique visualization component. To this end, a segmented image is derived from the original depth image. When one of the top three gradient values is an image-based feature indicating an obstacle, the corresponding pixel location is identified using the image feature value within the specific image segment. For instance, if 'Depth_34' corresponds to an obstacle, using the value of 'Depth_34', the corresponding pixel value can be calculated and be located within the image segment in the 3rd row, 4th column of the image grid. Subsequently, the neighboring pixels are iteratively examined to determine if they are part of the same structure based on their depth values. After identifying the entire structure, it is then highlighted/colored to visually represent the obstacle. This visual aid complements the textual explanations and offers a clear, intuitive understanding of the UAV's navigation space. This approach provides effective visual explanations despite the use of a simplified, vectorized form of the image data. This dual approach of textual and visual explanations enhances the explainability and transparency of the UAV's decision-making process.

V. EXPERIMENTS AND RESULTS

A. Simulation Environment and Experimental Setup

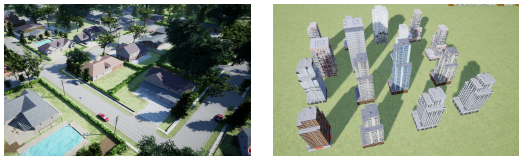
The DRL model training and evaluating experiments were carried out in AirSim simulation environment [29] with Unreal Engine 4.25 [30]. The environment used for training is shown in Figure 3a. This environment depicts a suburban setting with houses, trees, and utility poles as obstacles. At the beginning of each iteration, the UAV start position is set to $(0, 0, 0)$ coordinate, which is the middle of the environment. The goal location is randomly placed $120\,\mathrm{m}$ away from the start position with a boundary size of 200 m by 200 m. The altitude of the UAV was set to $20\,\mathrm{m}$. All the experiments were performed on a desktop computer with Nvidia GeForce 3090 GPU and 64GB RAM.

B. DRL Model Results

The proposed DRL model (SAC+MLP) is evaluated based on success rate - the percentage of instances where UAV successfully reached a given goal location without collisions in a specified time frame. The evaluation was carried out in two distinct environments: (i) a suburban area, which was the training environment for the model, and (ii) an urban setting characterized by tall buildings. As shown in Figure 3b, the urban setting contains features noticeably different from those in the training environment. However, the task remains an obstacle avoidance problem, with buildings being the only type of obstacle. The model underwent 100 iterations for each environment with randomly assigned goal locations. Figure 4 shows a sample set of trajectories followed in two environments. The results of this evaluation are presented in Table 1. In both environments, the model achieved a success rate of over 95%, indicating its effectiveness in guiding the UAV to its destination without colliding with any obstacles and its robust performance across varied scenarios.

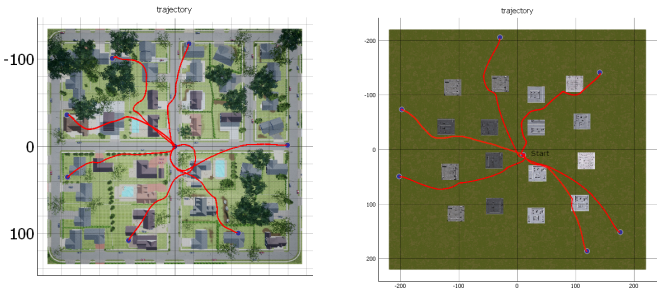
C. Model Explanation Results

In this work, textual and visual model explanations are generated for every 5-time step window, corresponding to a duration of 0.5 seconds, given a continuous change in velocity or yaw rate within that window. Four different instances from three different evaluation episodes are chosen



(a) Suburban Environment (b) Urban Environment

Fig. 3: Aerial view of training/evaluation environments



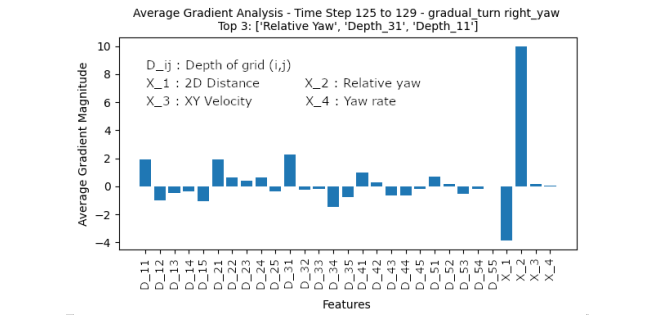
(a) Suburban Environment (b) Urban Environment

Fig. 4: Top view with UAV trajectories

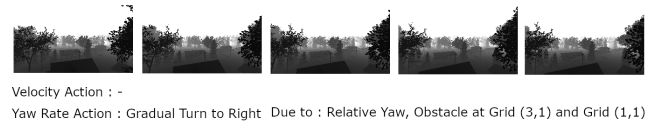
Environment	Success	Fail: Obstacle	Fail: Time-limit	Success Rate
Suburban-NH Center	95	2	3	95%
Urban - City	97	3	0	97%

TABLE I: Success rate of DRL model for two different environments.

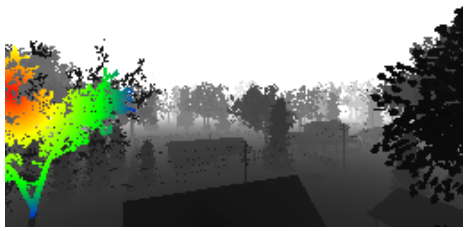
to demonstrate the textual and visual explanations of the proposed method. During the time step window 125 to 129 of the first iteration, yaw rate action is gradual turn to the right and gradient analysis shown in Figure 5c identifies the most influential features, 'Relative yaw', 'Depth_31' and 'Depth_11'. In that, features with positive influence are considered because the turn to the right is indicated by a positive yaw rate. As shown in Figure 5b, the associated text explanations are 'Relative yaw', 'Obstacle at Grid (3,1)' and 'Obstacle at Grid (1,1)'. Moreover, the visual explanation generated for the period, presented in Figure 5a, highlights an obstacle on the left of the UAV. This observation supports the need for a rightward adjustment in the UAV's heading to align with the intended direction and avoid the detected obstacle.



(a) Yaw rate gradient analysis



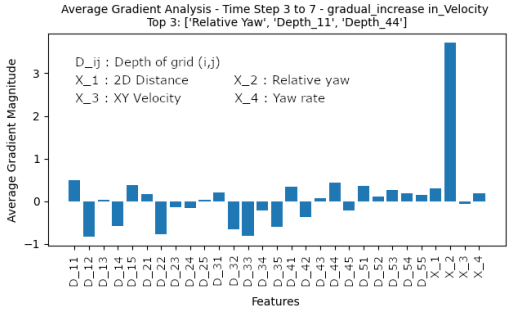
(b) Textual explanations



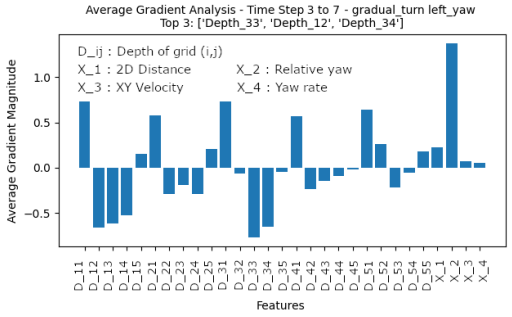
(c) Visual explanation

Fig. 5: Results for the time step 125 to 129 of iteration 1

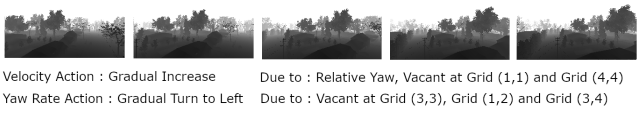
During the interval from time steps 3 to 7 in the second iteration, actions include a gradual increase in velocity and a gradual turn to the left. Therefore, positively influential values for velocity and negatively influential values for yaw rate are considered from gradient plots shown in Figure 6a and Figure 6b. As shown in Figure 6c, textual explanations include 'Relative yaw' and 'Vacant at Grid (1,1)' and 'Grid (4,4)' for velocity, and 'Vacant at Grid (3,3)', 'Grid (1,2)', and 'Grid (3,4)' for yaw rate, indicating no obstacles nearby due to zero-valued depth measures. This observation is expected as it is the start of this second iteration, and the UAV is facing a completely different direction from the goal location. Consequently, the UAV takes in-place leftward rotation to align with the goal direction while gradually increasing the velocity to head toward the goal. Additionally, this illustrates that depth features not only indicate the presence of obstacles but also highlight the absence of obstacles, i.e., vacant spaces that UAVs could traverse.



(a) Velocity gradient analysis



(b) Yaw rate gradient analysis

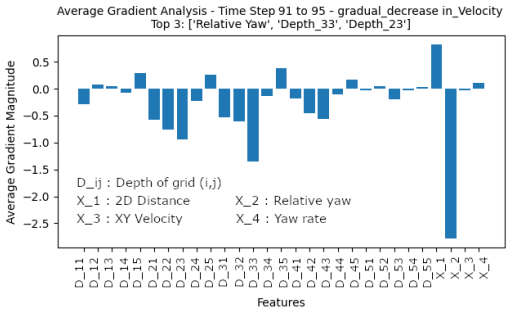


(c) Textual explanations

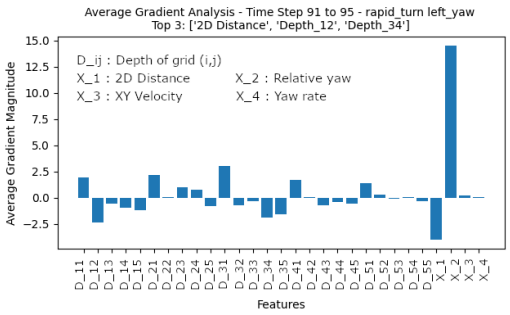
Fig. 6: Results for the time step 3 to 7 of iteration 2

During the time window 91 to 95 in the second iteration, the actions included a gradual decrease in velocity and a rapid turn to the left. The relevant gradient analysis is shown in Figures 7a and 7b. The textual explanations are 'Relative yaw', 'Obstacle at Grid (2,3)' and 'Depth Grid (3,3)' for velocity action and '2D Distance', 'Vacant at Grid (1,2)' and 'Obstacle at Grid (3,4)' for yaw rate action as shown in Figure 7c. In this scenario, Figure 7d reveals an obstacle on the right, a tree branch, influencing the UAV to reduce

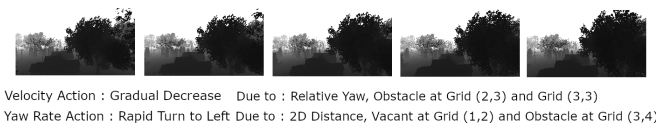
velocity and turn left sharply for avoidance. Moreover, the velocity deduction is influenced by relative yaw, facilitating a safer turn. The yaw rate decision being influenced by the 2D distance can be understood as a means of optimizing the path and ensuring obstacle avoidance. The UAV is already close to the goal location in this time step. If the UAV is far from the target, a slight adjustment in yaw rate might be sufficient. However, as it gets closer, especially in scenarios requiring precise navigation, the control system may opt for more significant yaw rate adjustments to ensure accuracy in reaching the goal or avoiding obstacles without compromising safety.



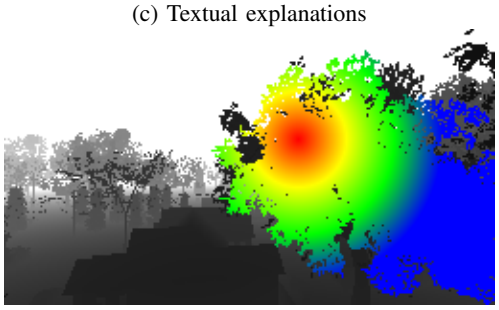
(a) Velocity gradient analysis



(b) Yaw rate gradient analysis



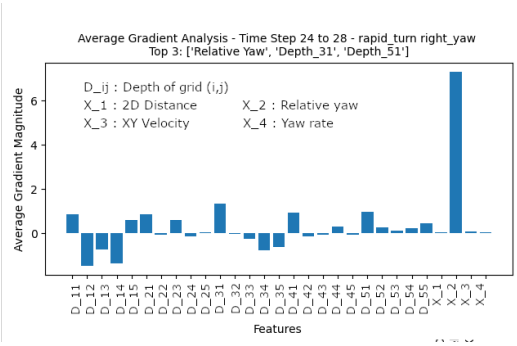
(c) Textual explanations



(d) Visual explanation

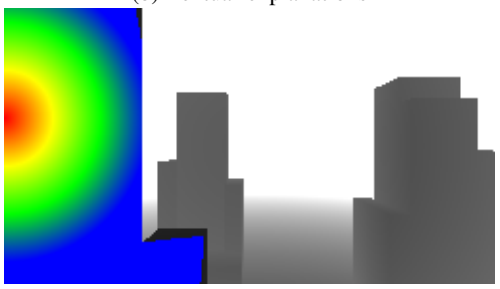
Fig. 7: Results for the time step 91 to 95 of iteration 2

Similar results were obtained from the city environment analysis. One such scenario is presented below, where the UAV took a rapid turned to the right. The gradient analysis indicates 'Relative Yaw', 'Depth_31' and 'Depth_51' as the most influential positive features. Figure 8b presents the associated textual explanations, which are 'Relative yaw', 'Obstacle at Grid (3,1)' and 'Grid (5,1)'. The visual explanation highlights an obstacle on the left side of the UAV, which is expected from the 'Depth 31' and 'Depth 51' image features. That clearly explains the rapid right turn as well. Even though the right turn moves the UAV further from its goal, making the relative yaw between the UAV's heading and the goal direction increase, relative yaw remains a significant factor in the decision-making process as it continuously measures the UAV's alignment with the goal direction, which is essential for eventual course correction.



(a) Yaw rate gradient analysis

(b) Textual explanations



(c) Visual explanation

Fig. 8: Results for the time step 24 to 28 of iteration 1 of city environment

Conclusion and Future Direction

In this work, we presented a novel, explainable DRL approach to solve the UAV autonomous navigation problem. The proposed approach streamlines the DRL architecture and enhances computational efficiency by integrating SAC and MLP policies and forgoing traditional convolutional layers. Two explanation approaches, i.e., textual explanations and visual explanations, were integrated with the DRL-based navigation algorithm to derive human comprehensible explanations for UAV actions in real-time.

The XAI algorithm was based on a novel moving window gradient analysis, which offers insights into the UAV's decision-making process by evaluating the impact of various observations, including image segments, distance to goal, velocity, and yaw metrics, on the UAV's actions. The moving window approach allowed the generation of textual and visual explanations at an adequate rate without overwhelming onboard computational resources and eliminating misinterpretations due to outliers. The visual explanation portion of the proposed XAI algorithm segments input images, identifying crucial areas of influence through a color-map application. The textual explanations augmented these visual explanations, prompting a brief rationale for control actions executed by the UAV. This dual approach bridges the gap between complex AI decision-making processes and end-user comprehension, fostering greater confidence in AI systems.

As future work, we propose to extend the proposed DRL algorithm to detect and mitigate environmental disturbances (e.g., wind) and uncertainty in optical data, enhancing the robustness of the proposed DRL algorithm against noise and disturbances. We plan to extend the explanation component of the proposed algorithm to explain UAV's actions when features are strongly correlated. Additionally, we plan to evaluate how different image grid sizes affect computational complexity and explanation results, as well as compare the impact of varying moving window sizes on the results.

ACKNOWLEDGMENT

The authors would like to thank Memorial University of Newfoundland for their financial and intellectual support for this research project.

REFERENCES

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

[2] G. Tong, N. Jiang, L. Biyue, Z. Xi, W. Ya, and D. Wenbo, "Uav navigation in high dynamic environments: A deep reinforcement learning approach," *Chinese Journal of Aeronautics*, vol. 34, no. 2, pp. 479–489, 2021.

[3] S. Zhang, Y. Li, and Q. Dong, "Autonomous navigation of uav in multi-obstacle environments based on a deep reinforcement learning approach," *Applied Soft Computing*, vol. 115, p. 108194, 2022.

[4] O. Banimelhem and B. Al-khateeb, "Explainable artificial intelligence in drones: A brief review," in *2023 14th International Conference on Information and Communication Systems (ICICS)*. IEEE, 2023, pp. 1–5.

[5] M. Mehta, V. Palade, and I. Chatterjee, *Explainable AI: Foundations, methodologies and applications*. Springer Nature, 2022, vol. 232.

[6] G. Bathla, K. Bhadane, R. K. Singh, R. Kumar, R. Aluvalu, R. Krishnamurthi, A. Kumar, R. Thakur, and S. Basheer, "Autonomous vehicles and intelligent automation: Applications, challenges, and opportunities," *Mobile Information Systems*, vol. 2022, pp. 1–36, 2022.

[7] J. Roghair, A. Niaraki, K. Ko, and A. Jannesari, "A vision based deep reinforcement learning algorithm for uav obstacle avoidance," in *Intelligent Systems and Applications: Proceedings of the 2021 Intelligent Systems Conference (IntelliSys) Volume 1*. Springer, 2022, pp. 115–128.

[8] L. Xie, S. Wang, A. Markham, and N. Trigoni, "Towards monocular vision based obstacle avoidance through deep reinforcement learning," *arXiv preprint arXiv:1706.09829*, 2017.

[9] C. Sampedro, H. Bavle, A. Rodriguez-Ramos, P. de La Puente, and P. Campoy, "Laser-based reactive navigation for multirotor aerial robots using deep reinforcement learning," in *2018 IEEE/RSJ international Conference on intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 1024–1031.

[10] H. Van Hasselt, A. Guez, and D. Silver, "Deep reinforcement learning with double q-learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 30, no. 1, 2016.

[11] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[12] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *International conference on machine learning*. Pmlr, 2014, pp. 387–395.

[13] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[14] G. Bachmann, S. Anagnostidis, and T. Hofmann, "Scaling mlps: A tale of inductive bias," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[16] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[17] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 3319–3328.

[18] A. Holzinger, A. Saranti, C. Molnar, P. Biecek, and W. Samek, "Explainable ai methods-a brief overview," in *International workshop on extending explainable AI beyond deep models and classifiers*. Springer, 2022, pp. 13–38.

[19] H. Chen, S. Lundberg, and S.-I. Lee, "Explaining models by propagating shapley values of local components," *Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability*, pp. 261–270, 2021.

[20] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[21] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 618–626.

[22] N. Douglas, D. Yim, B. Kartal, P. Hernandez-Leal, F. Maurer, and M. E. Taylor, "Towers of saliency: A reinforcement learning visualization using immersive environments," in *Proceedings of the 2019 acm international conference on interactive surfaces and spaces*, 2019, pp. 339–342.

[23] L. He, A. Nabil, and B. Song, "Explainable deep reinforcement learning for uav autonomous navigation," *arXiv preprint arXiv:2009.14551*, 2020.

[24] Stereolabs, "Zed 2 - 3d camera for spatial understanding," https://www.stereolabs.com/products/zed-2 [Online], accessed: 10-Mar.-2024.

[25] DJI, "Matrice 300 rtk," https://enterprise.dji.com/matrice-300 [Online], accessed: 10-Mar.-2024.

[26] T. Hickling, N. Aouf, and P. Spencer, "Robust adversarial attacks detection based on explainable deep reinforcement learning for uav guidance and planning," *IEEE Transactions on Intelligent Vehicles*, 2023.

[27] M. Lobier, M. Dubois, and S. Valdois, "The role of visual processing speed in reading speed development," *PloS one*, vol. 8, no. 4, p. e58097, 2013.

[28] G. E. Box and R. D. Meyer, "An analysis for unreplicated fractional factorials," *Technometrics*, vol. 28, no. 1, pp. 11–18, 1986.

[29] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics: Results of the 11th International Conference*. Springer, 2018, pp. 621–635.

[30] E. Games, "Unreal engine version 4.25," https://www.unrealengine.com [Online], 2020, accessed: 15-Oct.-2023.