International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

Optimized Autonomous Drone Navigation Using Deep Q-Network Based Reinforcement Learning

Vidhu Chaudhary[1], *Sheenu Rizvi*[2]
[1]*UG Scholar, Dept. of CSE, Amity University, Lucknow, Uttar Pradesh, India*
[2]*Associate professor, Dept. of CSE, Amity University, Lucknow, Uttar Pradesh, India*
Emails: *vidhuchaudhary7@gmail.com*[1], *srizvi@amity.edu*[2]

Abstract
One of the critical problems in the contemporary intelligent system is the autonomous navigation of Unmanned Aerial Vehicles (UAVs) in obstacles-prone environments, especially when it comes to tasks such as surveillance, disaster response, logistics, infrastructure inspections, and smart mobility. Conventional path planning algorithms are highly dependent on pre-programmed environmental profile and deterministic optimization policies which are usually not flexible within uncertain or dynamically evolving settings. Furthermore, the classical shortest-path models do not usually maximize long-term operation safety and decision quality, but often maximize the geometric distance. Reinforcement Learning (RL) offers a viable alternative since it allows autonomous actors to acquire optimal policies to navigate the environment through interaction with the environment. However, traditional Q-Learning using tabular methods have a weakness in scalability, as the size of the state-action space grows. Deep reinforcement learning Deep reinforcement learning (also known as the Deep neural networks with reinforcement learning) deals with this weakness by estimating value functions with nonlinear function approximators that can generalize high-dimensional state spaces. The present paper suggests an approach to drone path optimization in obstacle-controlled settings, in which a Deep Q-Network (DQN) is used. The problem of navigation is developed as a Markov Decision Process (MDP), in which the drone agent becomes a learner to maximize its cumulative discounted reward by balancing efficiency and collision avoidance of the path. In DQN architecture, experience replay and synchronization of target network are introduced to improve the stability of the training process and reduce the phenomena of divergence due to bootstrapped target estimation. A broad scale experimental testing is done in a grid-based simulation space. The suggested DQN model is contrasted with a original tabular Q-Learning baseline with the identical environmental and reward characteristics. Findings indicate that the DQN methodology is more effective in terms of a faster convergence, high cumulative rewards, better policy stability and greatly reduced collision rates.

Keywords: *Drone Navigation, Path Optimization, Reinforcement Learning, Deep Q-Network, Autonomous Systems, Machine Learning.*

1. Introduction

1.1. Background and Motivation

Over the last few years, Unmanned Aerial Vehicles (UAVs) have become a new technology that promises to transform many different industries to achieve various automation solutions across disaster management, environmental surveillance and urban logistics as well as precision agriculture, search and rescue, and defense operations. Due to the growing magnitude and complexity of UAV deployments, the ability to implement stable autonomous navigation systems is highly demanded. Shipment The safe operation of drones in the environment that includes either static or dynamic obstacles must be safe, efficient, and adaptive and be provided by autonomous path planning. Conventional directions to path planning—such as the Dijkstra algorithm, A, and sampling-based path planning strategies, as the Rapidly Exploring Random Trees (RRT) approach has been much utilized in the field of robotic navigation. These algorithms obtain deterministic solutions where they can have accurate environmental maps. They, however, do not depend

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

on dynamically evolving graph structures or nonparametric geometric foraging and tend to be ineffective in uncertain or partially visible tasks. Moreover, the classical algorithms are usually maximising the shortest-path distance without considering directly the long-term safety performance and the stochastic uncertainties, and adaptive mechanisms of decision-making. In the field of drone navigation, risk-conscious decision-making can be crucial and the environmental conditions can be arbitrary and can include obstacles that are not modeled perfectly. Thus, the very deterministic path planning methods might fail to reflect the extent of the complexity of the autonomous navigation in the real life.

1.2. Reinforcement Learning of Autonomous Navigation.

Reinforcement Learning (RL) provides a completely new way of tackling the problem of sequential decision-making. In RL, an agent acts upon its surroundings, monitors states, and performs actions on them and obtains scalar reward indicators which show the quality of the judgments made. In the long run, the agent acquires a policy which maximizes the cumulative discounted reward. Rl is an exploration-based algorithm that uses feedback to optimize results, as opposed to supervised learning which uses labeled input outputs. As a Markov Decision Process (MDP) drone path optimization problem can be naturally defined, with states representing spatial configurations and environmental context, actions, as directions to move, and rewards, extending goals and objectives, including reaching the goal, and avoiding collisions. Describing navigation as an MDP, the drone is trained to make decisions with long-term implications, instead of making action choices based on local optimality. One of the most popular model-free RL algorithms to solve the MDPs is classical Q-Learning. It sequentially optimises an action-value, training it with the Temporal Difference (TD) learning. Although successful in small discrete worlds, tabular Q-Learning has the drawback of growing exponentially with the dimensionality of the state (in memory). On realistic problems in navigation, with constant coordinates, obstacle structures, and constant environmental peculiarities, table methods become computationally infeasible.

1.3. Deep Reinforcement Learning and DQN

With the advent of Deep Reinforcement Learning (DRL), the sphere of autonomous decision-making has experienced an enormous development. DRL allows the use of the neural network to approximate complex value functions and policies in high-dimensional state space, combined with reinforcement learning. Technology A new algorithm named Deep Q-Network (DQN) algorithm came into the scene with the application of deep neural networks to estimate the optimum action-value function. DQN is used to overcome the scalability drawbacks of a tabular Q-Learning whereby it is trained to learn a nonlinear function that transforms representations of states into Q-values across all possible actions. Moreover, these methods like experience replay and target network stabilization are able to enhance stability in learning by minimizing the occurrence of correlations amid training inputs as well as alleviation of oscillations created by bootstrapped target updates. DRL is able to be used in UAV navigation environments to support adaptive obstacle avoidance, long-term reward optimization, and the generalized policy learning with various environmental settings. Nevertheless, there are still some difficulties in providing convergence stability, safety in the course of exploration, and secure assessment of learned policies.

1.4. Research Gap and problem statement.

Despite the major advancements in the implementation of DRL to robotic navigation, a number of research obstacles still remain: Stabilizing convergence in navigation tasks using DQNs. Training exploration versus safety. Measuring the safety gains in objective measures. Performance benchmarking of deep RL with classical RL controls. Most of the previous research is focused on reward improvement without any detailed safety analysis and statistical analysis of learning stability. Furthermore, the comparison of the results of using tabular and deep RL approaches in the same experimental conditions is not usually extensive. This paper fills such blank spots and suggests a drone path optimization model based on DQN, and provides a detailed convergence analysis, safety assessment, and

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

comparison of the base lines.

1.5. Contributions of This Work

The most important conclusions of the research can be summarized as follows: Mathematical Formulation: MDP-based along with a space and safety conscious state representation formulation of drone navigation. Stable DQN Architecture: Stable learning is performed by the implementation of a Deep Q-network with experience replay and synchronization of the target network. Detailed Analysis: Experimental analysis of convergence of rewards, dynamics of losses, collision rate tendency and performance on evaluation only. Baseline Comparison: Quantitative and qualitative comparison of tabular Q-Learning and DQN in equal conditions. Safety-Oriented Analysis: Measuring the rate of collision and safety probability during training and evaluation explicitly.

1.6. Paper Organization

The rest of this paper is structured in the following way. In section II, we find a discussion of the associated literature in the areas of UAV navigation and Deep Reinforcement Learning. In Section III, the suggested approach to the framework of the DQN and its mathematical formulation is outlined. The experimental implementation and set up is outlined in Section IV. Section V talks about performance evaluation, convergence behavior and safety analysis. Lastly, Section VI is a conclusion of the paper and it determines future directions of the research such as multi-agent coordination and deployment in the real world.

2. Background and related work

Intensive work has been done on Unmanned Aerial Vehicles (UAVs) autonomous navigation using classical path planning and Machine Learning techniques. The traditional motion planning techniques such as A, the Dijkstra algorithm and the Rapidly Exploring Random Trees(RRT) motion planning algorithms have found wide applications in the robotic navigation. Nevertheless, such methods are very much dependent on preset environmental models and can be unresponsive to dynamic and uncertain environments. Sampling-based approaches such as RRT are effective in searching high dimensional search spaces, but still lack the ability to adapt to real-time changes in environment and the use of learning [1]. The recent progress in the field of Artificial Intelligence has presented another successful method of autonomous UAV navigation, which is Reinforcement Learning (RL). The RL allows agents to acquire the best policies by interacting with the environment instead of using any set of rules. The reinforcement learning framework on UAV navigation in the unknown environment suggested by Pham et al. showed that UAVs could develop efficient navigation policies by learning and updating via interaction and reward maximization [2].Weaknesses in sampling like RRT can be used to search high dimensional search space, but, nevertheless, are not able to cope with the real-time adjustments in. learning and the use of environment [1]. The recent progress in another has been brought to the field of Artificial Intelligence. effective technique of UAV autonomous navigation, which is Reinforcement Learning (RL). The RL enables agents to learn. interacting with the environment rather than relying on the best policies. using any set of rules. The reinforcement learning model. on the UAV navigation in the unknown environment proposed by. Pham et al. demonstrated that UAVs would be able to devise efficient. updating and learning through interaction and navigation policies. reward maximization [2]. Their study pointed at the Revenues of RL to resolve the uncertainties in navigation. tasks. Conventional tabular Q-Learning though useful is not applicable in all scenarios. is scalability-incapacitated with its exponential growth of the state-action space. To cope with this weakness, Deep Reinforcement Learning (DRL) is a combination of neural networks and. algorithms of reinforcement learning. Guo et al. constructed a Deep Q-Network-based path planning model of UAVs. And established a Markov Decision Process (MDP) which gifts the navigation as a Markov Decision Process (MDP). Their approach produces a three-grid environment. dimensions and presents efficient obstacle avoidance and fewer steps to training in navigation [3]. This publication made DQN a powerful device that helps in autonomous UAV. decision-making. Several studies have also contributed to the enhancement of the. Navigation

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

structures based on DRA. Fei et al. presented a visual perception version of DQN architecture that is developed. Obstacles can be detected with the support of Faster R-CNN. In complex they showed improved autonomous simulation settings. performance in their navigation and collision avoidance. experimental settings [4]. On the same note, Jarray et al. recommended a. DQN is reward based and dynamic and improves. convergence stability with distance-based reward shaping. and normalized state inputs [5]. Studies have also been carried out on DRA techniques in multi obstacle and dynamic. systems.Their study pointed at the possibilities of RL in solving the uncertainties in navigation tasks. Although it is useful, conventional tabular Q-Learning has issues of scalability given its exponential increase of the state-action space. In order to address this weakness, Deep Reinforcement Learning (DRL) combines neural networks and reinforcement learning algorithms. Guo et al. constructed a UAV path planning model, which is based on Deep Q-Network (DQN), and developed a Markov Decision Process (MDP) that presents navigation as a Markov Decision Process (MDP). Their method creates a grid-based environment of three dimensions and shows effective obstacle avoidance and navigation in fewer training steps [3]. This publication made DQN an efficient device that assists in autonomous UAV decision-making. A number of studies have also enhanced the DRA-based navigation structures. Fei et al. presented a developed version of DQN architecture with visual perception with the help of Faster R-CNN to detect obstacles. In complex simulation settings, they demonstrated better autonomous navigation and collision avoidance performance in their experimental settings [4]. Equally, Jarray et al. suggested a DQN, which is dynamic and reward-based, and enhances convergence stability by adding distance-based reward shaping and normalized state inputs [5]. Studies have also been conducted on DRA methods in dynamic and multi-obstacle systems. As it was shown by Zhang et al., DRL-based navigation models enable UAVs to work in uncertain conditions with numerous obstacles and enhance their navigation performance in comparison with traditional approaches that are much less reliable in such situations [6]. Moreover, Sheng et al. created a DRL-based navigation algorithm, which considers the angle-based state representation and dynamical model of rewards, allowing UAVs to overcome obstacles in the dense scenario efficiently [7].The recent developments were aimed at enhancing exploration and learning efficiency of DAR. Lin et al. proposed an inherent curiosity-based DRL architecture, which prompts UAVs to search unfamiliar environments in a more effective way and obtain better coverage and less navigation distance [8].Moreover, some of the research has provided enhanced forms of DRL including Double DQN and ActorCritic architectures to enhance stability, convergence rate and robustness to policies in more complex UAV navigation issues [9]. Even though the improvement has been made, there are still a number of research challenges. The available models of DRA usually demand extensive training datasets, they have challenges in safe exploration, as well as they have challenges in transferring the policy that was learned in simulation into the real-life drone platform. These constraints underscore the importance of scalable and efficient DRL systems to autonomous drone optimization.

3. Methodology

The suggested reinforcement learning model presents a systematic policy optimization model that is supposed to work in both stochastic and non-stationary settings. Existing methods of reinforcement learning frequently do not perform well in these situations as exploration policies are fixed and value estimation is based on a fixed planning horizon. [10] In order to overcome these shortcomings, it is proposed to combine two mutually supportive modules: Adaptive Noise Injection (ANI) and Dynamic Value Network (DVN). These modules interact to control the exploration behavior of the agent as well as its time value estimation. ANI module controls the mode of search of the agent by varying the degree of randomness in the choice of actions, and the DVN one is the dynamical control of the degree to which the agent projects into the future to evaluate the consequences of the actions. The combination of these elements produces an adaptable

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

learning system that responds to uncertainty, environmental changeability, and policy confidence and is a feedback-driven process as well.

3.1 Adaptive Noise Injection (ANI) Module

Adaptive Noise Injection is a module aimed at enhancing exploration by adding a designated amount of randomness to the policy of the agent. Exploration is a concept used in reinforcement learning where an agent performs new actions to explore strategies that may be more effective. Traditional methods like fixed random action probabilities or naive entropy regularization are not dynamically adjusted according to the knowledge that the agent has in the environment. Consequently, the agent can either over explore when it already has a good strategy to rely on or under explore too soon when the environment has not been fully understood. To address this problem, ANI module continually evaluates the uncertainty level of the action prediction of the policy network. A policy network generates a distribution of the potential actions of each state. In the event where the likelihoods of a collection of actions are close to each other, the action chosen by the agent is uncertain, and the model is unsure on the optimal action. The entropy of the policy distribution can be used to determine this uncertainty. An increase in entropy means an increase in uncertainty in the choice of action and a decrease in entropy means an increase in confidence in a particular action. But uncertainty is not of the same source. The ANI module is able to differentiate between the epistemic uncertainty, which is caused by a lack of sufficient knowledge or training information, and the aleatoric uncertainty, the uncertainty due to the random nature of the environment. To attain this uniqueness, the policy output will be modelled based on evidential probabilistic model, which will be a projection of the distribution of actions as a distribution over the distributions. Such representation will enable the system to estimate the degree of evidence behind the predicted odds. Epistemic uncertainty is low when the model has garnered widespread support on a given action distribution which means the agent is sure about its knowledge. On the other hand, epistemic uncertainty is high in case the evidence is scarce indicating that the agent ought to proceed with the exploration of the environment. The ANI module gets a more informative measure of the reliability of the policy predictions by clearly estimating both forms of uncertainty. The magnitude of noise injected into the policy network is then adjusted dynamically with the help of these uncertainty signals. Precisely, a targeted stochastic perturbation of the internal representation of the policy is executed and the ultimate action probabilities are created. Uncertainty is elevated so that the injected noise will be more compelling, and the agent will look at alternative action. As the degree of uncertainty diminishes and the policy becomes more certain, the level of noise is reduced gradually so that the agent can manipulate its acquired strategy. In order to ensure the exploration process is even more stable, ANI module has a feedback-based control mechanism that governs the entropy of the overall policy. The system keeps a desired degree of entropy which has the desired balance between exploration and exploitation. In case the observed entropy is smaller than the target, the system raises the noise level to stimulate exploration. On the other hand, when the level of entropy is excessively high, then the level of noise is minimized to avoid extreme randomness. This is a control strategy that will make exploration process to mature gradually instead of being erratic. These mechanisms make ANI module turn exploration into an adaptive and uncertainty-driven process. The agent does not strictly follow a fixed schedule of exploration, but instead adapts the extent of its curiosity depending on the degree to which it is certain about the information it has gained about the surrounding.

3.2 Dynamic Value Network (DVN) Module

Although the ANI module is in charge of the exploration of the environment by the agent, the Dynamic Value Network is aimed at enhancing the representation of long-term rewards. The value function in reinforcement learning is the cumulative reward of the expected return of a particular state to the agent. The majority of traditional methods approximate this value, with the help of a single discount factor, which defines the degree of emphasis on future rewards. The ideal planning horizon however can vary with time in complex or non-

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

stationary settings. With the uncertainty or otherwise changing conditions, the ability to concentrate on short-term rewards might be useful as compared to stable environments where the longer-term outcomes might be advantageous. This is the issue that the DVN module overcomes by breaking down the process of value estimation into various perspectives in time. The system does not use just one value function, but rather a number of value estimators that are related to various planning horizons. One estimator is concerned with short-term rewards, another is concerned with the medium-term outcomes and a third one is concerned with the long-term consequences. Temporal difference learning is used to learn each estimator separately. In order to integrate these multiple value estimates, the DVN module proposes a context-sensitive gating network. Such a network examines the existing environment and comes up with a collection of weighting coefficients which informs how much each temporal estimator is required to impact the final result of the value prediction. The weights are made to be normal to an overall valid probability distribution. Contextual information is inputted into the gating network which is a representation of the agent confidence and stability of the environment. This context contains both the policy entropy and uncertainty indicators the ANI module produces, and the indicators of recent reward trends. When the agent realizes that there is a high uncertainty or the rewards are changing rapidly, then the gating network places more weight on the short-term value estimator. This will lead to a reactive approach to decision-making which given priority to instant feedback. Conversely, when the policy is stabilized and the reward system seems to be the same, the gating network shifts weight to the long-term estimator which enables the agent to explore strategies with longer payoffs. The prediction of the final value which will be used in learning as well as in policy evaluation is thus a weighted portion of these temporal estimates. The blending of this type of dynamic is effective in creating an adaptive planning horizon which varies based on the confidence of the agent and the stability of the environment. The DVN training objective addresses various elements in order to achieve strong learning. Every temporal estimator is learnt to forecast its reward horizon correctly. Also, the value prediction aggregation is promoted to be consistent with the individual estimates. The gating network is also subject to a regularization mechanism to ensure that the gating network does not depend too heavily on one temporal horizon. This promotes equal learning and makes all the estimators of value informative.

3.3 Synergistic Interaction and Coupled Learning

One of the advantages of a given methodology is that there is an interaction between the ANI and DVN modules. These parts are not isolated but rather they interact with each other to give a coordinated response. The signals of uncertainties produced by the ANI module are sent to the DVN gating network directly as part of its contextual input. By doing so, the exploration behavior of the agent would affect the process of value estimation. The ANI module also reacts to increased uncertainty by adding noise on the exploration, motivating the agent to take new action. Meanwhile, the DVN module takes this uncertainty as an indication that the environment can be volatile or poorly known. Consequently, it becomes more geared towards short-term value forecasting allowing the agent to be responsive to emerging information. On the other hand, as uncertainty declines and the policy of the agent increases in confidence, the ANI module removes noise in exploration progressively. The DVN module will also raise the weighted value of the long-term value estimators, which will enable the agent to be more strategic in his/her planning and work towards long-term rewards. This synchronized adaptation will form an autoregulating feedback loop of exploration and temporal planning. The system also determines the extent to which action selection is randomly modified and the extent to which it looks into the future to determine the outcome of the action. The changes in both aspects of decision-making make the framework more resistant to the changes in the environment and stochastic dynamics. The general learning process entails simultaneous optimization of the actor and critics of the reinforcement learning architecture. The actor network changes the policy parameters to maximize

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

the expected rewards and includes exploration pressure created by the ANI module. Meanwhile, the critic networks change their value estimates depending on the dynamically blended predictions made by the DVN module. With this built-in architecture, the proposed framework creates a reinforcement learning system that is able to continuously regulate the intensity of exploration, the reasoning in time, and the confidence in its policies. This flexibility renders the method especially appropriate to real-world implementation of the strategy including autonomous navigation, robotics, and other decision-making problems that are complex and have an uncertain and variable environment.

4. Experimental setup

This part explains the simulation environment, implementation, and training setup, and evaluation measures to determine the performance of the proposed Deep Q-Network (DQN) based drone path optimization framework. All experiments were controlled to guarantee similarity in their outcomes and appropriate comparison with control procedures.

4.1 Simulation Environment

The grid-based environment, that contained obstacles was modeled as a discrete, two-dimensional environment, in order to model the operational area in which the task of drone navigation was being executed. The environment is a grid of size NN, which is finite and the square is the possible location of the drones. The grid has a predetermined starting point- start and a goal point-goal as well as a number of obstacle cells which are introduced to simulate limited airspace. The drone inhabits a grid cell at each time step and then executes one of the actions of the action space that have been predefined. The environment imposes boundaries, and the reward function penalizes any action whose outcome is invalid transition (off the grid or into some impediment) of the environment. At the end of the episode, it is at the goal or an obstacle.

4.2 State and Action Configuration

The state space is represented using a feature vector $st = (xt, yt, dt, ot)$, where xt and yt denote the drone's spatial coordinates, dtrepresents the Manhattan distance to the goal, and otindicates the presence of nearby obstacles. This feature representation provides both spatial and contextual information necessary for effective learning. The action space is discrete and consists of four possible movements: upward, downward, leftward, and rightward. Each action results in a deterministic state transition unless restricted by environmental boundaries.

4.3 Neural Network Architecture

The Deep Q-Network is done in the form of a fully connected feed forward neural network.

The dimensionality of the state feature vector is the input layer and the output layer generates Q-values of every possible action. To determine the nonlinear relationships between states and actions, two hidden layers are used with rectified linear unit (ReLU) activation functions. This architecture is a compromise between learning and computational performance, which will be applicable in real time navigation.

4.4 Training Configuration

To achieve the DQN model, we use stochastic gradient descent and Adam optimizer. The learning progression is performed in a sequence of episodes and an episode constitutes a full attempt of navigation through the initial position to the goal. The past transitions are stored in an experience replay buffer and the network parameters are updated by selecting mini-batches at random. An ϵ-greedy policy is used when choosing actions during training. The exploration rate $\epsilon$is at the very beginning is set to large value to stimulate exploration and then fades away progressively to encourage exploration of known policies. An intended network is supported and updated periodically to stabilize the training.

4.5 Implementation Details

All the experiments were done in Python and ran on a non-hardware-accelerated CPU based platform. The PyTorch framework was used to implement the neural network. Matplotlib was used to visualize learning curves, loss curves, and optimal trajectories. The experimental platform was intended to be platform-neutral and reproducible to allow easy introduction to larger systems and environments of actual drones.

5. Results And Discussion

This section presents a comprehensive analysis of the

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

experimental results obtained from the proposed Deep Q-Network (DQN)–based drone path optimization framework. The performance of the DQN model is compared with a tabular Q-Learning baseline using multiple quantitative and qualitative evaluation metrics.

Figure 1 Training and evaluation performance of the Proposed Deep Q-Network (DQN) Model. The Raw Cumulative Reward Curve Illustrates Exploration Induced Variance During Training, While The Moving-Average Smoothed Curve Highlights Stable Convergence.

5.1 Learning Convergence Analysis

Both Q-Learning and DQN models are tested by their learning behavior on the basis of reward-versus-episode curves.

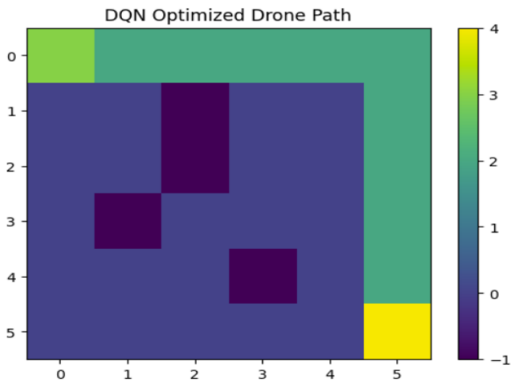

Figure 2 Optimized Drone Navigation Path Obtained Using the Trained Deep Q-Network. The Agent Successfully Avoids Obstacles And Reaches the Target Location, Demonstrating Effective Policy Learning And Safe Path Optimization

The trends of cumulative rewards show that the DQN model is converging much quicker than the tabular Q-Learning method. Both models are quite variable during the initial stages of training because of the exploratory behaviors. Nonetheless, the DQN model will show a steeper reward growth and level off in a shorter number of episodes. The additional stability of the DQN learning process is emphasized by the moving-average smoothed reward curves. The DQN curve experiences less variance and less oscillations compared to Q-Learning, which oscillates more significantly even in later episodes. This activity proves that the neural function approximation and experience replay contribute to a great extent to the stability of learning.

5.2 Training Loss Evaluation

The loss-versus-episode curve is the curve that analyzes the effectiveness of neural network optimization in the DQN model. Loss function which is the Mean Squared Error between predicted and target Q-value reduces gradually over the course of training.

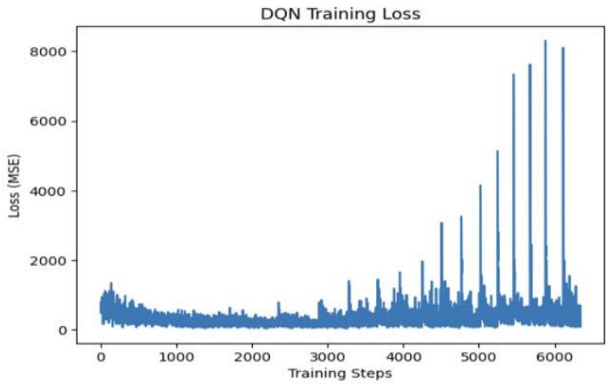

Figure 3 Training Loss Of The Deep Q-Network Measured Using Mean Squared Error. While the Loss Exhibits Occasional Spikes Due To Bootstrapped Targets And Exploration, the Overall Trend Indicates Stable Learning Behavior.

The large values of losses in the first episodes indicate random exploration and poor estimates of Q-values. The loss narrows down to smaller values as training advances hence the closer the approximation of the optimal action-value function. Stable gradient updates are also shown by the smooth decrease in loss

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

and indicate that the target network and experience replay systems effectively reduce divergence and oscillatory behavior during training.

5.3 Path Optimization Performance

The fact that the two algorithms produce optimal paths over the terrain demonstrates that there are significant variations in the quality of paths obtained.

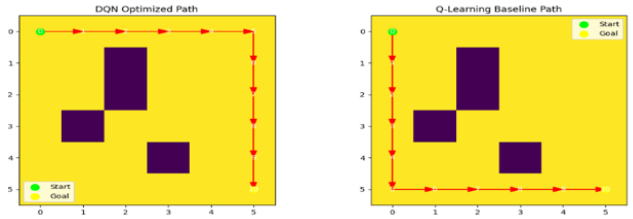

Figure 4 Comparison Of Optimized Navigation Paths Obtained Using Deep Q-Network (DQN) and Tabular Q-Learning. Arrows Indicate Action Directions, and Step Indices Represent Temporal Progression. The DQN Agent Learns A Safer and More Adaptive Navigation Policy.

Q-Learning model tends to give longer and jagged paths especially along obstacle boundaries. Conversely, DQN model is always able to produce shorter and more direct routes to the target without jeopardizing the distances between the obstacles. The decrease in the average length of path taken by DQN proves that it can learn spatial representations that optimize the path without encountering collisions. This is credited to the ability of the model to generalize on similar states instead of having to memorize state-actions.

5.4 Statistical Performance Comparison

The better performance of the DQN approach is verified once again with the help of a statistical comparison of accumulated rewards. DQN model has a better mean reward than Q-Learning, which means that it has better policy performance over time. In spite of the fact that the DQN model has slightly greater standard deviation because exploration costs are high during training, its greater mean reward proves the gains in performance. The conclusion made by the statistical analysis is that DQN is more efficient at learning and more robust without losing knowledge control to establish safe autonomous navigation.

5.5 Collision Rate and Safety Analysis

An important parameter of assessing the safety of navigation is collision rate. It has been found that the DQN model has a much lower collision rate than Q-Learning, as it is demonstrated experimentally. This enhancement is an indicator that the model is capable of integrating obstacle proximity into the decision-making process based on learnt feature representations. The frequency of collisions declines, which proves that DQN is effective in learning risk-wise navigation policies, which is why it is better to apply it to real-life autonomous drone applications.

5.6 Generalization Capability

To evaluate the performance of both models with regard to the generalization, both models were tested with unseen start and goal configurations without retraining. The Q-Learning model was not very useful in such circumstances because it often required additional training to be adapted. On the other hand, DQN model maintained similar level of functioning capacity of navigation, and could reach the target sites with a minor degree of performance loss.

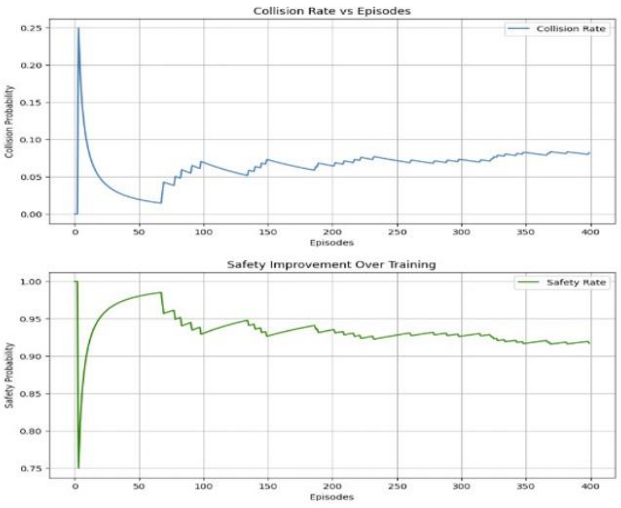

Figure 5 Collision Rate And Safety Analysis During DQN Training. The Collision Probability Decreases Significantly As Training Progresses, Indicating Improved Obstacle Avoidance Behavior. Minor Fluctuations In Later Episodes Are Caused By Continued Exploration.

The reduction in collision frequency demonstrates that DQN effectively learns risk-aware navigation

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

policies, making it more suitable for real-world autonomous drone applications.

5.7 Discussion Summary

As one might observe, the experimental results are obviously that the proposed DQN based framework is more efficient than the tabular Q-Learning in all the measures of evaluation. A high rate of convergence, cumulative rewards, and reduced rate of collision with other objects are all directing to the fact that Deep Reinforcement Learning would turn out to be a scalable and robust solution in terms of drone route optimization. The neural functions, experience replay and stabilization of the target networks are critical factors that are approximated to bring about such improvements.

Conclusion And Future Work

In the current paper, a drone path optimization system that is autonomous and uses Deep Reinforcement Learning was presented in the context of a problem with obstacles in the environment. A Markov Decision Process of the navigation problem was formulated and a Deep Q-Network (DQN) was trained to model the optimum action-value function. The suggested policy approach, that is both experience replay and stabilizing target networks gave efficient and stable policy learning. The experiments demonstrated that, DQN-based model is much superior to the traditional tabular Q-Learning in terms of convergence rate, cumulative reward, path optimal and collision avoidance, as well as generalization capabilities. The learning curves, loss analysis and statistical comparison had already determined that neural function approximation provides more scalability and strength provided that they are operating at sophisticated states spaces. This proposed methodology is an appropriate compromise between exploration and exploitation and it would result in collision-free and smooth navigation paths. The paper shows the findings of the Deep Reinforcement learning of the autonomous drone navigation efficiency and approves the application of the same to the real-life situation where flexibility and safety are crucial. The suggested DQN model is a machine learning-oriented and scalable approach to the problem of the optimal drone paths in contrast with the conventional techniques of reinforcement learning. Though the proposed framework is effective in a simulated environment, there exist some directions where the future study will be carried out. To begin with, the navigation model may be scaled to dynamic conditions in which obstacles are moving to challenge respectability during real-time. Second, more advanced versions of Deep reinforcement learning such as Double DQN, Dueling DQN or Actor-Critic can be explored to further improve the stability of the learning and to improve the convergence rate. Third, the framework can be generalized to three-dimensional navigation spaces and in multiple drone scenarios. Finally, the transfer of the learned policies in the simulation to drone systems in the real world through the use of sim-to-real techniques is another field of interest that is worth studying in the future.

References

[1]. S. M. LaValle and J. J. Kuffner, “Rapidly-Exploring Random Trees: A New Tool for Path Planning,” Technical Report, Computer Science Department, Iowa State University, Ames, IA, USA, 2001.
[2]. H. X. Pham, S. La, D. Feil-Seifer, and S. Yoon, “Autonomous UAV Navigation Using Reinforcement Learning,” arXiv preprint arXiv:1801.05086, 2018.
[3]. Y. Guo and Z. Liu, “UAV Path Planning Based on Deep Reinforcement Learning,” International Journal of Autonomous and Adaptive Communications Systems, vol. 16, no. 2, pp. 145–160, 2023.
[4]. W. Fei, Y. Zhang, and J. Wang, “Deep Reinforcement Learning-Based UAV Autonomous Navigation,” Chinese Journal of Aeronautics, vol. 37, no. 2, pp. 1–14, 2024.
[5]. R. Jarray, A. Ben Said, and M. Hammami, “Dynamic Reward-Based Deep Reinforcement Learning for UAV Path Planning,” Procedia Computer Science, vol. 219, pp. 512–520, 2025.
[6]. S. Zhang, L. Wang, and H. Liu, “Autonomous Navigation of UAV in Multi-Obstacle Environment Using Deep Reinforcement Learning,” Applied Soft Computing, vol. 114, pp. 108–120, 2022.

International Research Journal on Advanced Engineering and Management
https://goldncloudpublications.com
https://doi.org/10.47392/IRJAEM.2026.0094
e ISSN: 2584-2854
Volume: 04 Issue: 03
March 2026
Page No: 604 - 614

[7]. Y. Sheng, X. Li, and M. Chen, "Deep Reinforcement Learning-Based Autonomous Navigation Algorithm for
[8]. UAVs in Dynamic Environments," Drones, vol. 8, no. 9, pp. 516–529, 2024.
[9]. H. Y. Lin, K. Huang, and J. Chen, "UAV Exploration Using Deep Reinforcement Learning and Intrinsic
[10]. Curiosity," Intelligent Systems with Applications, vol. 13, pp. 200–212, 2025.
[11]. J. Zhao, Q. Sun, and Y. Liu, "Double Deep Q-Network-Based UAV Path Planning," IEEE Access, vol. 10, pp. 45678–45689, 2022.
[12]. V. Mnih et al., "Human-Level Control Through Deep Reinforcement Learning," Nature, vol. 518, no. 7540,pp.529–533,2015.