Advances in Artificial Intelligence and Machine Learning; Research 4 (4) 2969-2980 Received 19-09-2024; Accepted 02-12-2024; Published 9-12-2024

RUDE: Fusing Rules and Deep Learning for High-Speed Drone Path Planning

Yan Zheng yanzheng@tju.edu.cn
College of Intelligence and Computing, Tianjin University,
Tianjin, China

Yanran Li lyr@fzu.edu.cn
School of Economics and Management, Fuzhou University,
Fujian, China

Yanan Zhang
Tianjin University of Science and Technology,
Tianjin, China

Shixi Lian
College of Intelligence and Computing, Tianjin University,
Tianjin, China

Yongming Wen
Beijing Institute of Control and Electronics Technology,
Beijing, China

Yue Yu
Beijing Institute of Control and Electronics Technology,
Beijing, China

Corresponding Author: Yanran Li

Abstract

This paper focuses on the problem of emergency restricted zone avoidance for high-speed drones. Existing path planning methods struggle with inefficiency due to large search spaces, slow convergence rates, and difficulties in path planning for high-speed drones. To overcome these challenges, we propose a novel approach that integrates rules with deep learning. This hybrid approach simplifies decision-making by converting temporal decisions into a limited set of middle way-points, significantly reducing the complexity of both the state and solution search space. Additionally, rules are employed to prevent aimless exploration within the solution space. To enhance the algorithm performance, we introduce a situation prediction model, which is trained to capture the relationship between way-points and flight outcomes, such as restricted zone encounters and energy consumption. Experimental results demonstrate notable improvements over purely rule-based methods, with high success rates in avoiding restricted zones and maintaining sufficient kinetic energy to reach the goal. This

Citation: Yan Zheng, et al. RUDE: Fusing Rules and Deep Learning for High-Speed Drone Path Planning. Advances in Artificial Intelligence and Machine Learning. 2024;4(4):172.

approach effectively addresses the challenges posed by high-speed drones operating under complex physical models and dynamic emergency scenarios.

Keywords: High-speed drone, Path planning, Deep learning

1. INTRODUCTION

Path planning algorithms are fundamental to navigation, particularly for the automatic generation of a path from an origin to a destination within a defined area. During this process, it is essential to avoid restricted zones while maintaining low path costs, such as minimizing distance or energy consumption. Path planning is widely recognized as an NP-hard problem [1], and both rule-based and learning-based methods have been explored to address its inherent challenges. However, key challenges remain, including slow convergence and inefficiency exploration in handling large search spaces [1].

Path planning algorithms can generally be divided into rule-based and learning-based approaches, each with its own inherent limitations. Rule-based path planning algorithms, while effective in known environments, require the construction of a global map for decision-making. This separation between perception and decision-making makes them difficult to apply in dynamic or emergency environments, where adaptability is crucial. Moreover, rule-based approaches tend to be computationally complex and highly sensitive to the size of the search space, further limiting their applicability in real-time scenarios [2]. Current intelligent path planning algorithms primarily include evolutionary algorithms (EAs) and reinforcement learning (RL) methods. However, EA often requires numerous iterations, and the path optimization process depends on real-time interaction with the environment, making it challenging to find optimal solutions quickly [3]. On one hand, many EAs have proven ineffective when the search space is too large [2]. On the other hand, RL algorithms struggle to efficiently explore optimal solution sequences in vast search spaces [4].

In this work, we developed a practical simulation focused on restricted zone avoidance for high-speed drones. In the simulation, the drone operates at high velocity, detecting restricted zones while progressing towards a target point. The challenge lies in navigating around these restricted zones while ensuring sufficient kinetic energy to reach the target. Failures occur when the drone either enters a restricted zone or depletes its energy reserves before reaching the destination. Several key challenges emerge in this scenario: the state space and solution space are large due to the amount of mission-related information; the dynamic nature of the environment introduces unseen scenarios during deployment, which were not encountered during training; the vehicle's complex physical model and environmental state information complicate the computation of an optimal solution for every situation; and the drone's high speed makes it difficult to follow pre-planned paths precisely. Given these complexities, existing methods fail to adequately address the needs of high-speed drones in such dynamic and constrained environments.

We propose a method called RUDE, fusing **RU**les and **DE**ep Learning to obtain our optimal policy. Our policy outputs optimal way-points for the drone to track. We begin by collecting data and training a model to establish the correspondence between way-points and flight outcomes, such as entering a restricted zone and energy consumption. During testing, we generate 1000 sets of way-points randomly around the geometrically calculated way-points and use the trained deep learning model, i.e. the Situation Prediction Model (SPM) to predict the flight outcomes for all way-point sets. The best set of way-points is selected as the policy output.

RUDE offers several advantages. First, it transforms the complex temporal decision-making process into a simplified decision based on a small number of trajectory way-points, drastically reducing the complexity of the solution space. Second, by incorporating rule-based methods, the approach avoids aimless searches across the entire solution space. Additionally, the deep learning component captures the relationship between way-points and real-world flight outcomes, leveraging the generalization capabilities of deep learning to address variability between training and deployment scenarios. This hybrid approach bypasses the need for computing optimal theoretical solutions in real-time and mitigates difficulties related to tracking pre-planned paths at high speeds. Furthermore, RUDE decouples data collection from model training, enabling parallelized data collection and rapid model training. This results in a more lightweight and flexible solution compared to evolutionary algorithms and reinforcement learning, which require continuous real-time interaction with the environment during the optimization process.

Our contributions can be summarized as: (1) We propose a practical scenario of evading restricted zones for high-speed drone; (2) We transform the complex sequential decision-making path planning problem into a decision based on a small number of way-points on the trajectory, reducing the state space and action space through state reduction; (3) We combine deep learning to bypass challenges related to the complex physical model of the drone, difficulties in computing the optimal solution in emergency scenarios, and challenges related to tracking the planned path for high-speed drone; (4) Experimental results demonstrate that our method RUDE has high success rate and outperforms pure rule-based methods.

2. RELATED WORK

Various approaches have been proposed to tackle the path planning problem for drones, leveraging both rule-based and learning-based techniques.

Rule-based path planning. Dijkstra-based algorithms, such as in [5], employ a greedy strategy that iteratively selects the node with the lowest cost and updates the costs of neighboring nodes. In contrast, several works [6–11], utilize the A* algorithm [12], which incorporates heuristics to guide the search towards the goal. These methods evaluate nodes based on the sum of the cost-so-far and the estimated cost-to-go, using heuristic functions. For dynamic environments, D* Lite [13], is an incremental search algorithm that efficiently updates paths when the environment changes. It starts with a search similar to Dijkstra's algorithm and uses a priority queue to manage cost updates.

Learning-based path planning. To improve adaptability in new environments, learning-based approaches have been developed. One such approach is supervised learning. For example, [14] proposed a novel solution using Support Vector Machine (SVM) to solve unmanned aerial vehicle (UAV) path planning. In [15], Partial Differential Equations (PDE) were incorporated to optimize paths for multiple UAVs. More recently, RL has been applied to optimize path planning under various constraints. An enhanced Deep Q-Network (DQN) model, introduced in [16], integrates Lazy Training into the Deep Q-Network [17], to improve path planning. An advanced version, the Dueling Double Deep Q-Networks (D3QN) algorithm [18], has been developed to further optimize path planning, though its computational complexity increases with environmental complexity. Another approach, the Deep Reinforcement Learning Echo State Network (Deep-RL-ESN), was proposed in [19]. This model combines a Deep Echo State Network (DESN), a deep recurrent neural network, with RL to solve multi-UAV and online path planning problems. Additionally, the Deep-Sarsa algorithm [20], based on the reinforcement learning Sarsa approach, was introduced for optimizing UAV path planning. Several studies also focus on Q-learning-based methods. In [21], an improved Q-learning algorithm integrates ϵ-greedy and Boltzmann strategies to enhance UAV path planning, demonstrating better performance in generating shorter paths with fewer steps. Another novel Q-learning approach, MARER Q-learning, was proposed in [22], to address UAV path planning by incorporating an Experience Relay function to improve algorithm performance.

However, none of these methods specifically addresses the unique challenges of path planning for high-speed drones. Our method differs from these approaches by combining rule-based techniques with deep learning, transforming the complex sequential decision-making process of path planning into a more manageable task based on a small set of way-points. This significantly reduces the complexity of the solution space while maintaining adaptability and efficiency.

3. METHODOLOGY

In this section, we provide a detailed description of our method, RUDE consisting of three parts: data collection, SPM, and the SPM-guided controlling. In RUDE, a set of way-points can be regarded as a solution, and our policy determines the optimal set of way-points for effective drone navigation.

3.1 Data Collection

The purpose of data collection is to gather a diverse set of trajectory information to provide a robust foundation for training the SPM. To generate a wide range of scenarios, we randomly vary the drone's initial positions, as well as the locations and sizes of the restricted zones. By expanding the variety and coverage of scenarios, we enhance the generalization capability of the deep learning model. In each scenario, we establish a coordinate system with the center of the restricted zone as the origin, and define the direction from the restricted zone center to the target point as the positive x-axis.

FIGURE 1 (left) illustrates the high-level idea. In each scenario, we first generate a lower-bound solution via the rule-based algorithm. Specifically, using the aforementioned coordinate system, way-points are set at $(-0.8r, \pm 1.0r)$, $(0, \pm 1.2r)$, and $(0.5r, \pm 0.9r)$, forming the lower-bound performance, where r represents the radius of the restricted zone. The vertical coordinates are assigned positive or negative values based on the sign of the drone's (blue circle in the FIGURE 1, initial vertical position, ensuring that the drone flies around the restricted zone on the same side it starts from, thereby reducing the difficulty of avoiding the zone. Our experiments indicate that when the drone's trajectory passes through the restricted zone, tracking these three way-points increases the likelihood of successfully reaching the target.

Next, we generate a broad set of candidate way-points. Specifically, around the previously mentioned lower-bound solution, we create numerous way-point groups, with each group consisting

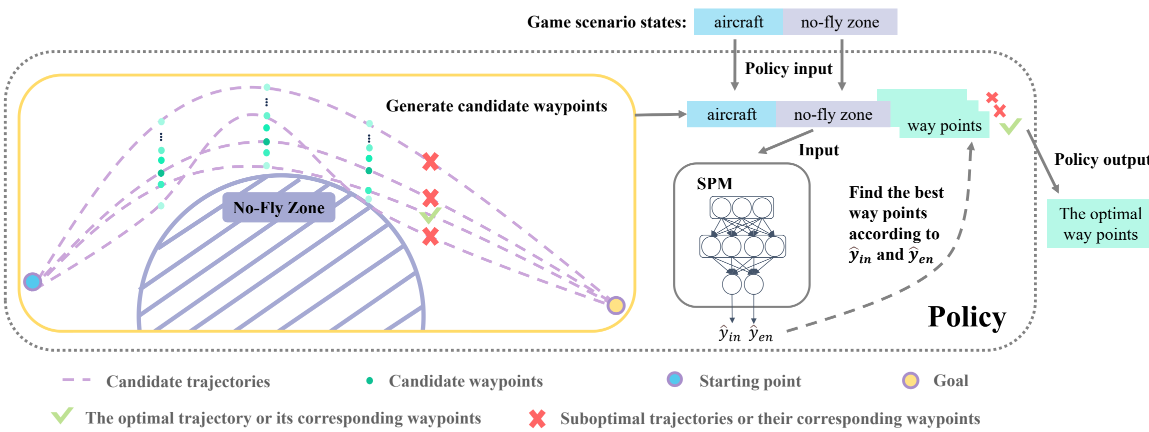


Figure 1: Intuition example of the RUDE algorithm. Light green way-points represents the candidates randomly generated near the lower-bound solution (darkest green way-points). The purple dotted line represents the trajectory derived from each set of way-points. Blue and Yellow circles represent the start point and the goal. The dashed line with the check mark represents the optimal way-points which avoiding the restricted zone with the minimal kinetic energy.

of three points corresponding to the three way-points in the lower-bound solution. This results in a large set of potential candidate trajectories. The distance between these candidate way-points and the respective way-points in low-bound solution is limited to $0.2r$ to prevent exploration of ineffective solutions. An illustration of these way-points, shown as green points, is provided in FIGURE 1.

We then have the drone follow each set of way-points in the simulator environment, generating a large amount of interaction data. From this data, we extract key information to create structured trajectory data. This includes the way-point sets defining the trajectory, the drone's initial information (such as its initial position relative to the center of the restricted zone and initial speed), the restricted zone details (including its radius and the center position), and the flight result information (e.g., whether the trajectory enters the restricted zone and the change in kinetic energy from the start to the end of the flight). The way-points, drone's initial information, and restricted zone data are used as input features, denoted as $\vec{x}$.

The flight result information, including whether the drone enters the restricted zone and its kinetic energy consumption, is represented by y_{in} and y_{en}, respectively. Finally, we store the structured trajectory data in dataset D. It is worth noting that we record the drone's initial position relative to the center of the restricted zone to simplify the state space and enhance the generalization ability of the situational prediction model.

3.2 Situation Prediction Model

The Situation Prediction Model (SPM) comprises two components: the restricted zone entry discrimination component and the energy consumption prediction component. These components predict whether the drone will enter a restricted zone and estimate the drone's energy consumption relative to its starting point after completing a flight, respectively. Importantly, the models are trained to simulate flight outcomes during testing, prior to any actual interaction with the game environment. Both components take the game scenario state and way-points as inputs, with the former predicting restricted zone entry and the latter forecasting kinetic energy consumption. The structure of the SPM is depicted in the SPM section of FIGURE 1.

The restricted zone entry discrimination model is a binary classifier (denoted as ω) . The input to this model is a concatenation of the drone's initial state, the restricted zone information, and the set of way-points. The model outputs $\hat{y}_{in} = P_{\omega}(y|x)$, which represents the probability that the drone will enter the restricted zone while tracking the given set of way-points in the scenario. To train the discriminator ω , we calculate the cross-entropy loss using the predicted probability and the true labels, as shown in Equation 1.

$$L_{\omega} = -E_{x,y_{in}\in D}\left(y_{in}\log(\hat{y}_{in}) + (1 - y_{in})\log(1 - \hat{y}_{in})\right) \tag{1}$$

The energy consumption prediction model (denoted as ϕ) serves as a function approximator. It takes as input a concatenation of the drone's initial state, restricted zone information, and the set of way-points, and outputs $\hat{y}_{en} = P_{\phi}(y|x)$, representing the predicted energy consumption of the drone after completing its flight around the restricted zone. The model's performance is optimized using mean squared error between the predicted and actual energy consumption, as shown in Equation 3.

$$L_{\phi} = E_{x,y_{en}\in D}(\hat{y}_{en} - y_{en})^2 \tag{2}$$

We sum up two loss function items as the ultimate loss L to train the entire policy as follows:

$$L = L_{\omega} + L_{\phi} \tag{3}$$

Besides, the restricted zone entry discrimination model and the energy consumption prediction models are two independent models with the same network architecture but do not share parameters. Both models have a structure consisting of a multilayer perception (MLP) with two hidden layers of size 32 and use the ReLU activation function. During training, a weight decay of 0.01 is applied to prevent over-fitting, and each model is trained for 10,000 gradient descent steps.

3.3 SPM-guided Controlling

The drone's control policy takes as input the current state, which includes the drone's initial information and the restricted zone details, and determines the optimal set of way-points for the drone to follow. The process of identifying the optimal way-point set is guided by the SPM, as illustrated in FIGURE 1. When a restricted zone is near the drone, the first step is to assess whether it will impact the flight. If the straight line between the drone and the target point does not intersect the restricted zone, the zone is considered irrelevant, and the drone proceeds directly toward the target in a straight path. However, if the restricted zone poses an obstacle, the go-around process is triggered, which activates the control policy based on the SPM.

The goal of the drone's control policy is to generate an optimal set of way-points for emergency scenarios. Here, we use the rule-based expert system as a backbone algorithm to generate numerous navigation paths, then applying SPM to evaluate the quality of each path, ultimately arriving at the optimal solution. In specific, in the beginning, the expert system (i.e., a set of predefined rules) is used for generating way-points (refereed to as the lower-bound solutions). These way-points are positioned around the periphery of the region at coordinates $(-0.8r, \pm 1.0r)$, $(0, \pm 1.2r)$, and $(0.5, \pm 0.9r)$, forming a flight trajectory. The positive or negative vertical coordinates are set as described in Section 3.1 to simplify avoiding the restricted zone.

Around each lower-bound solution, 1000 sets of way-points are generated, with each set comprising three points corresponding to the original rule-based way-points. This creates a total of 1000 candidate trajectories. The distance between these generated way-points and their corresponding rule-based way-points is restricted to within $0.2r$, ensuring that irrelevant solutions are not considered.

The control policy is then derived, with the goal of identifying the optimal solution. Specifically, the rule-based lower-bound solution is combined with the generated trajectories, resulting in 1001 candidate trajectories. Each candidate trajectory is concatenated with the current state and input into the SPM, which predicts the energy consumption and the likelihood of entering the restricted zone for each trajectory. By integrating these two factors, the trajectory that minimizes energy consumption while avoiding the restricted zone is selected as the optimal policy for the current state.

4. EXPERIMENT

In this section, we evaluate RUDE using a simulation platform with a realistic physics engine. We compare RUDE with a classical rule-based baseline, and the experimental results demonstrate the effectiveness and efficiency of our approach. First, we outline the environmental setup of our newly proposed scenario in Section 4.1. Then, we address the following research questions (RQs) regarding our approach:

RQ1: Can our approach achieve a high success rate, and is this success driven by more than just a rule-based approach?

RQ2: Can our situational discrimination model converge effectively, and how well does it generalize to complex scenarios?

RQ3: Does the accuracy of the situational discrimination model contribute to higher success rates?

We answer RQ1 in Section 4.2, and address RQ2 and RQ3 in Section 4.3.

4.1 Environmental Settings

In our simulation platform, we conduct a strategic game that involves an drone, restricted zones, and a target point. The positions of these elements, along with the drone's initial speed, the visual range of the restricted zone, and its radius, can all be manually adjusted. The platform accepts way-points entered sequentially based on their distance from the target and automatically guides the drone to follow them. The target point is automatically set as the final way-point, so the platform directs the drone to fly toward it. Our policy focuses on selecting the intermediate way-points.

In some scenarios, the drone must navigate around the restricted zone to reach the target. If the drone enters the restricted zone or runs out of kinetic energy before reaching the target, the game is failed. The challenge lies in balancing proximity to the restricted zone. If the drone flies too close, it risks entering the zone at high speed. Conversely, staying too far results in excessive energy loss from taking unnecessary turns, preventing the drone from reaching the target. Additionally, poorly chosen way-points (such as adjacent way-points that cause sharp turns) can lead to excessive energy consumption or prevent the drone from accurately tracking the way-points.

4.2 Main Results

In this section, we evaluate the effectiveness of RUDE by comparing it to two pure rule-based approaches designed by domain experts. The reason is that existing methods like learning-based methods suffer from finding a feasible solution due to large searching space and the physical constraints in this particular high-speed drone navigation domain.

Rule1: Set the way-points along the edge of the restricted zone at the coordinates: $(-0.8r, \pm 0.6r)$, $(0, \pm r)$, and $(0.5, \pm\sqrt{0.75}r)$.

Rule2: Set the way-points based on the rule-based lower-bound solution without using the SPM at: $(-0.8r, \pm 1.0r)$, $(0, \pm 1.2r)$, and $(0.5r, \pm 0.9r)$.

The positive or negative vertical coordinates are set as described in Section 3.1 to simplify avoiding the restricted zone.

First, to validate RUDE's effectiveness, we created a variety of game scenarios for testing, where the drone needs to avoid the non-fly zone and reach the goal. These scenarios feature different drone starting positions (randomly between 600-1300 km from the target), different initial speeds (randomly between 4-6 km/s), varying restricted zone radii (100-250 km), random restricted zone positions, and varying visual distances for the restricted zone (200-600 km). The effectiveness of RUDE and the pure rule-based baseline is tested on all the constructed scenarios. We run RUDE 500 times on each game scenario. The results are tabulated in TABLE 1. The results show that our approach achieves a significantly higher success rate (i.e., ratio that the drone successfully reaching to goal), improving performance by more than twofold compared to the purely rule-based approach, addressing **Q1**.

Table 1: Effectiveness of related baselines controlling the drone to reach the goal. Rule 1&2 are control policy manually designed by experts and RUDE achieves the best result.

RUDE	Rule1	Rule2
99.6%	41.8%	47.7%

Since RUDE incorporates a randomized policy, we also evaluated its stability by testing the success rate across multiple runs with different random seeds on five distinct game scenarios. We compared this with a method that does not use the SPM and randomly selects way-points near the rule-based performance lower-bound way-points described in Section 3.3. The outcomes are presented in TABLE 2.

Table 2: Average success rate with different seeds on five different game scenarios.

	Game1	Game2	Game3	Game4	Game5
Ours	**100%**	**100%**	**100%**	**100%**	**100%**
No model	50%	40%	**100%**	**100%**	50%

In the table, the restricted zone radius for Game3 and Game4 is only 100 km, which is small enough to easily bypass without significant loss of kinetic energy. As a result, success can be achieved even without using the SPM. The experimental results demonstrate that, despite the randomness in our policy, its performance remains consistently stable.

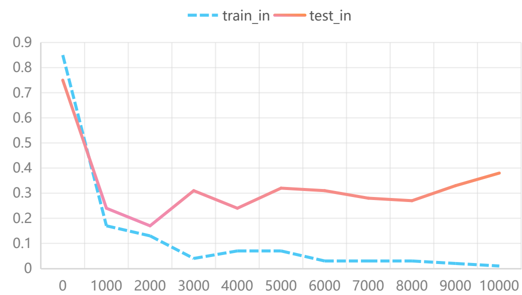
4.3 Additional Results

To gain deeper insights into how RUDE operates, we conducted additional experiments. To verify that both the energy consumption prediction model and the restricted zone entry discrimination model can converge effectively on complex scenario data, we present the training loss curves for these models.

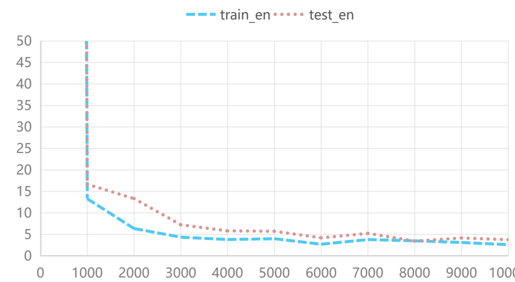
The blue curves in FIGURE 2 (a) and (b), show that the training error decreases steadily as the number of training steps increases, ultimately converging. Moreover, the convergence of RUDE is rapid, with both models stabilizing within just 1000 steps, addressing **Q2**.

Regarding the generalization capability of our SPM, the red curves in FIGURE 2 (a) and (b), represent the test error. The absence of significant over-fitting demonstrates that our SPM generalizes well and accurately predicts outcomes for scenarios that appear during testing but were not encountered during training.

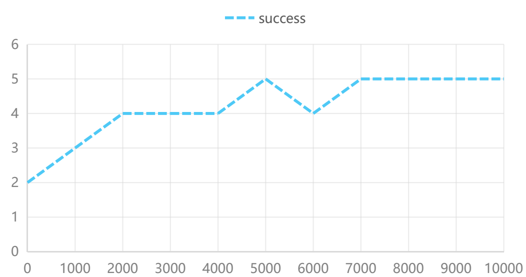
We also examined the critical role of SPM accuracy in the success rate by tracking how the success rate evolves with further training across five different game scenarios, as shown in FIGURE 2 (c). The results indicate that the success rate gradually increases and stabilizes with additional training, confirming that SPM accuracy is crucial to the success rate of the policy, answering **Q3**. Besides, we study the restricted zone entry discrimination model and visualize the Precise-Recall curve in FIGURE 2 (d), where we can find that the area under curve (AUC) is not small, meaning that the model has good ability in predicting whether entering the restricted zone or not. Besides, as shown in FIGURE 2 (d), we chose 0.82 as a reasonable threshold for the discrimination model to perform a binary prediction.



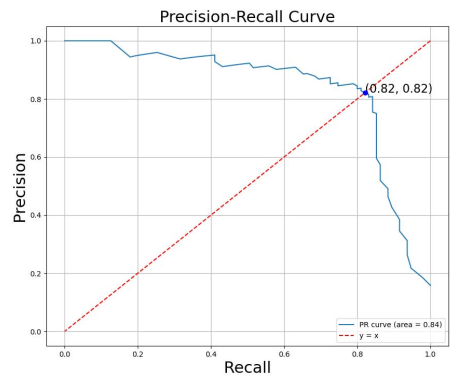
(a) Learning curve of the restricted zone entry discrimination model



(b) Learning curve of the energy consumption prediction model



(c) Number of successful games



(d) P/R curve of the discrimination model

Figure 2: Loss and success rate during training, where the x-axis represents training steps. (a) Training and testing loss of the restricted zone entry discrimination model. The train_in and test_in represent the training loss and testing loss respectively. (b) Training and testing loss of the energy consumption prediction model. The train_en and test_en represent the training loss and test loss of the energy consumption prediction model, respectively. (c) Success times among 5 game scenarios. (d) The P/R curve of the discrimination model.

5. CONCLUSION

In this paper, we proposed a novel approach that integrates rule-based methods with deep learning to tackle the challenge of high-speed drone emergency avoidance of restricted zones. Our method streamlines the decision-making process by converting temporal decisions into a limited number of trajectory waypoints, thereby reducing the complexity of both the state and solution spaces. By incorporating rule-based strategies, we minimize aimless exploration in the solution space, resulting in faster convergence and improved avoidance of restricted zones. Furthermore, we trained a deep learning-based situational prediction model to capture the relationship between waypoints and flight outcomes, enhancing performance in avoiding restricted zones while preserving sufficient energy levels. Experimental results demonstrated the effectiveness of our approach, achieving high success rates in both avoiding restricted zones and reaching target destinations. Our research advances the field of path planning for high-speed drone operations, offering a practical solution and paving the way for future developments in efficient and reliable path planning algorithms.

6. ACKNOWLEDGEMENTS

This work is supported by the National Project X of China (Grant No. JCKY2021204B104), and the Science and Technology on Information Systems Engineering Laboratory (Grant No. WDZC20235250409).

References

[1] Ait Saadi A, Soukane A, Meraihi Y, Benmessaoud Gabis A, Mirjalili S, et. al. UAV Path Planning Using Optimization Approaches: A Survey. Arch Comp Methods Eng. 2022;29:4233-4284.

[2] Zafar MN, Mohanta JC. Methodology for Path Planning and Optimization of Mobile Robots:a Review. Procedia Comput Sci. 2018;133:141-152.

[3] Zheng Y, Xie X, Su T, Ma L, Hao J, et. al. Wuji: Automatic Online Combat Game Testing Using Evolutionary Deep Reinforcement Learning. In: 34th IEEE/ACM International Conference on Automated Software Engineering. IEEE PUBLICATIONS. 2019:772-784.

[4] Ladosz P, Weng L, Kim M, Oh H. Exploration in Deep Reinforcement Learning: A Survey. Inf Fusion. 2022;85:1-22.

[5] Maini P, Sujit PB. Path Planning for a UAV With Kinematic Constraints in the Presence of Polygonal Obstacles. In: international conference on unmanned aircraft systems (ICUAS). IEEE PUBLICATIONS. 2016:62-67.

[6] Zhang Z, Wu J, Dai J, He C. A Novel Real-Time Penetration Path Planning Algorithm for Stealth Uav in 3D Complex Dynamic Environment. IEEE Access. 2020;8:122757-122771.

[7] Dong Z, Chen Z, Zhou R, Zhang R. A Hybrid Approach of Virtual Force and a Search Algorithm for UAV Path Re-Planning. In: 2011 6th IEEE conference on industrial electronics and applications. IEEE PUBLICATIONS. 2011:1140-1145.

[8] Wang Z, Liu L, Long T, Yu C, Kou J. Enhanced Sparse a Search for UAV Path Planning Using Dubins Path Estimation. In: Proceedings of the 33rd Chinese control conference. IEEE. 2014:738-742.

[9] Geng L, Zhang YF, Wang J, Fuh JY, Teo SH. Cooperative Mission Planning With Multiple UAVs in Realistic Environments. Unmanned Syst. 2014;02:73-86.

[10] Chen T, Zhang G, Hu X, Xiao J. Unmanned Aerial Vehicle Route Planning Method Based on a Star Algorithm. In: 13th IEEE conference on industrial electronics and applications (ICIEA). IEEE PUBLICATIONS. 2018:1510-1514.

[11] Zhang G, Hsu LT. A New Path Planning Algorithm Using a Gnss Localization Error Map for UAVs in an Urban Area. J Intell Robot Syst. 2019;94:219-235.

[12] Hart PE, Nilsson NJ, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. IEEE Trans Syst Sci Cybern. 1968;4:100-107.

[13] Koenig S, Likhachev M. D* Lite. In: Eighteenth national conference on Artificial intelligence. 2002:476-483.

[14] Chen Y, Zu W, Fan G, Chang H. Unmanned Aircraft Vehicle Path Planning Based on SVM Algorithm. In: Foundations and practical applications of cognitive systems and information processing. Proceedings of the first international conference on cognitive systems and information processing Beijing China. Springer. 2014:705-714.

[15] Radmanesh R, Kumar M, French D, Casbeer D. Towards a PDE-based large-scale decentralized solution for path planning of UAVs in shared airspace. Aerosp Sci Technol. 2020;105:105965.

[16] Wu J, Shin S, Kim CG, Kim SD. Effective Lazy Training Method for Deep Q-Network in Obstacle Avoidance and Path Planning. In: IEEE international conference on systems man and cybernetics (SMC). IEEE PUBLICATIONS. 2017:1799-1804.

[17] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, et. al. Human-Level Control Through Deep Reinforcement Learning. Nature. 2015;518:529-533.

[18] Yan C, Xiang X, Wang C. Towards Real-Time Path Planning Through Deep Reinforcement Learning for a UAV in Dynamic Environments. J Intell Robot Syst. 2020;98:297-309.

[19] Challita U, Saad W, Bettstetter C. Deep Reinforcement Learning for Interference-Aware Path Planning of Cellular-Connected UAVs. In: IEEE international conference on communications (ICC). IEEE PUBLICATIONS. 2018:1-7.

[20] Luo W, Tang Q, Fu C, Eberhard P. Deep-Sarsa Based Multi-Uav Path Planning and Obstacle Avoidance in a Dynamic Environment. In: Advances in swarm. Intelligence. Proceedings, Part II: 9th International Conference ICSI 2018 Shanghai China. 2018;10942:102.

[21] Yan C, Xiang X. A Path Planning Algorithm for Uav Based on Improved Q-Learning. In: 2nd International Conference on Robotics and Automation Sciences (ICRAS). IEEE PUBLICATIONS. 2018:1-5.

[22] 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 Prog. 2020;103:36850419879024.