2024 IEEE 27th International Conference on
Intelligent Transportation Systems (ITSC)
September 24- 27, 2024. Edmonton, Canada

Safe-assured Learning-based Deep SE(3) Motion Joint Planning and Control for UAV Interactions with Dynamic Environments

Yuanyuan Zhang, Weisong Wen* and Penggao Yan

2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC) | 979-8-3315-0592-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ITSC58415.2024.10919536

Abstract— **In the era of low-altitude economy, ensuring a safe and agile flight of Unmanned aerial vehicles (UAVs) through the obstacle environment is significant to expanding their interactive applications. Deep reinforcement learning (DRL) based methods have demonstrated promising performance in achieving reliable navigation and precise task execution for UAVs. However, due to its trial-and-error search characteristic, DRL fails to balance safety robustness while pursuing agile performance, especially during training. Moreover, this problem is exacerbated due to the existence of uncertain observation noise in dynamic environments. To address this issue, this paper proposes a safe-assured learning-based deep SE(3) joint planning and control framework. This framework firstly achieves high-level safety decision-making, online complex motion planning, and control for UAVs by seamlessly integrating DRL with nonlinear model predictive control (NMPC). Secondly, this paper constructs a safe stochastic reachability certificate to calculate the stochastic forward reachable set of planned trajectories under uncertain conditions to perform specific safe collision probability checks. This safety foresight mechanism adaptively selects belief space actions from planned actions to interact with the environment, further improving safety while reducing training time. In the simulation of the agile traversal of a fast-moving gate by UAV, we demonstrate that the proposed method can effectively reduce total collision incidents and training time, thereby enhancing training safety and efficiency to a large extent.**

I. INTRODUCTION

In the coming era of low-altitude space economy (LASE) [1], ensuring a safe and agile flight of Unmanned aerial vehicles (UAVs) through obstacle environments is crucial to expanding their interactive applications, such as parcel delivery [2], pedestrian safety monitoring [3], and etc. This requires UAVs to perform constrained motion planning and control within the special Euclidean group SE(3) space [4] while satisfying safety requirements. Kaufmann et al. [5] demonstrate that deep reinforcement learning (DRL) based UAVs outperform manually controlled UAVs in flying through the gate in drone racing. This is attributed to their seamless integration of planning and control, requiring minimal prior knowledge to address continuous control problems. However, the DRL training for an agile flight is an unconstrained trial-and-error search process, unavoidably causing numerous failures that pose risks to user and system safety. Moreover, the uncertainty of observation noises in dynamic environments exacerbates this impact, a factor that is not considered in racing that merely requires the traversal of static gates. In other words, it is interesting to see how the method in [5] performs when flying through dynamic gates.

End-to-End Safe Navigation Control: Numerous safe DRL methods have been developed to impose constraints on agents to improve policy safety. This involves reducing the number of constraint violations during the training process [6]. Existing methods can be roughly categorized into risk-aware exploration methods and external knowledge-based methods. Kaymaz et al. [7] propose a framework for UAVs to learn how to navigate through a series of moving gates, utilizing both the deep Q-network (DQN) and constrained Markov decision processes (CMDPs). While their method makes the agent more inclined towards safe actions, it adopts hard constraints, making the agent overly conservative. In the second category, external knowledge such as reachability analysis [8] can be leveraged to ensure the safety or stability of the system [9]. For example, an accessibility-based approach is proposed in [10] to compute the reachable set of the system and adjust cost values, ensuring that the system's reachable set falls within a designated safe region. However, these methods involve learning end-to-end black-box control policies that generally lack interpretability.

Combining Learning and Model-based Navigation Control: In recent years, there has been an increasing trend to combine the strength of model-based controllers [11] and DRL [12], allowing for the management of large-scale inputs, reducing human intervention in system design and tuning, and ultimately achieving adaptive and optimal control performance. Song [13] introduces a hybrid approach that utilizes a deep neural network (DNN) to predict traversal time for the decision variables of NMPC, demonstrating the effectiveness of flying through pendulum gates. However, the quadcopter's attitude in this method is fixed to align with the gate's direction, resulting in UAVs' conservative actions with limited agility. To solve this issue, Wang [14] develops a reinforcement-imitation learning (RIL) framework and employs a binary search algorithm to enhance the dynamic adaptability of NMPC. However, these methods do not consider the safety exploration issue during training, affecting both training efficiency and the safety of the system. [15] utilizes a convolutional neural network (CNN) based policy to predict the cost map of the track, which is employed for online optimization of control. However, this method assumes overly ideal states which may not be applicable in dynamic scenarios, such as parcel delivery tasks, and consequently, it may not provide sufficient safety guarantees.

To enhance the safety of optimization problems and policy execution, this paper proposes a safety-assured

Yuanyuan Zhang, Weisong Wen and Penggao Yan are with the Department of Aeronautical and Aviation Engineering, the Hong Kong Polytechnic University, Hong Kong, yuan-yuan.zhang@connect.polyu.hk, peng-gao.yan@connect.polyu.hk

Corresponding author: Weisong Wen. (E-mail: welson.wen@polyu.edu.hk)

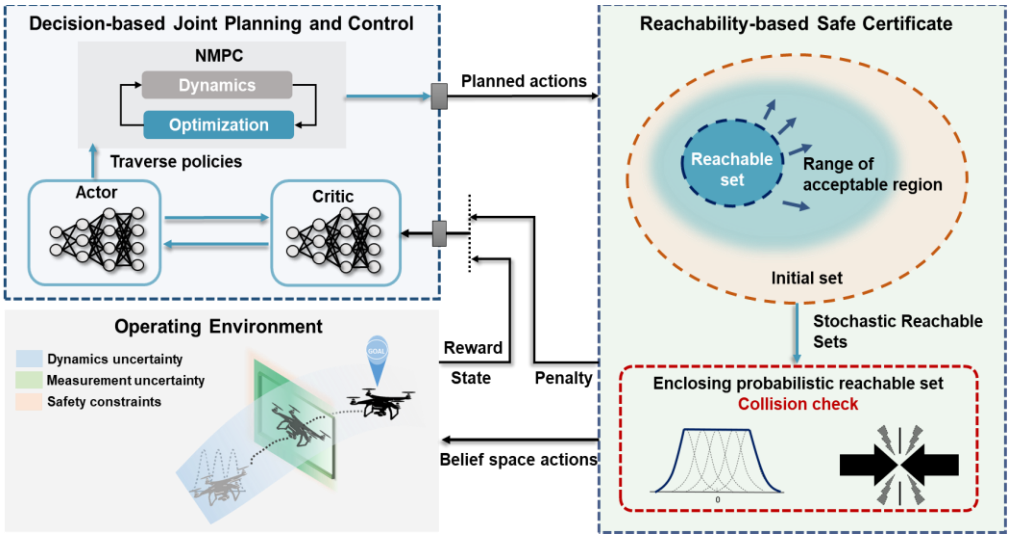


Fig. 1. Schematic diagram of the proposed method. The inputs of the framework are the observation state from the environment and the reward, and the outputs are the belief space action and proactive collision penalty.

framework of the proposed method is depicted in Fig. 1. The proposed method consists of two layers: a decision-based joint planning and control layer and a reachability-based safe certificate layer. In the decision-based joint planning and control layer, the DRL produces high-level traversal decisions, which are subsequently adopted by the NMPC to achieve optimal control and output planned actions based on constrained optimization. Then, a reachability-based safe certificate layer is deployed to expand these actions into a probabilistic region and generate a stochastic reachable set. After conducting collision safety detection on the closed probability reachable set considering noise uncertainty, this certificate adjusts the proactive collision penalty in the reward function while allowing belief space actions to interact with the environment, thus achieving a balance between agility performance and safety during the DRL training. The proposed method is compared with the state-of-the-art RIL method [14] on the task of flying through a moving gate, demonstrating the effectiveness of the proposed method.

The contributions of this paper are listed as follows:

(1) This paper proposes a learning-based deep SE(3) joint planning and control framework for UAVs by combining the deep learning capabilities of DRL for policy search with the advantages of model-based NMPC online constrained optimization. This framework achieves high-level decision-making, online-constrained motion planning, and control within SE(3) in highly dynamic environments.

(2) This paper proposes a safe foresight certificate based on stochastic reachability analysis. This certificate adjusts the proactive collision penalty to obtain a safe policy and allows belief space actions to interact with the environment. The effectiveness of the proposed method is verified in the task of traversing a fast-moving (dynamic) gate by comparing it with the RIL method.

II. Methodology

This section explains the method of safe-assured learning-based deep SE(3) motion joint planning and control for UAVs. We start with a concise description of the dynamic model used throughout this paper in Section II.A. In Section II.B, DRL is employed to handle high-dimensional observations and search for a traversal method. Subsequently, the learned policy by DRL is integrated into the NMPC as high-level decision variables for joint planning and control optimization. In Section II. C, we conduct the stochastic reachability certification with uncertainty awareness, ensuring that DRL can achieve robust yet non-conservative safety during training. Meanwhile, a shielding policy is produced for problem optimization and policy execution, mitigating conservativeness and potential hazards resulting from approximation errors and insufficient learning. Finally, Section II.D illustrates the next episode of the DRL policy search and update process.

A. Dynamics Model

Quadrotor Dynamics Model: In this study, the UAV's continuous-time kinematics model can be written as [16]:

$$\dot{\mathbf{x}}_q = \begin{bmatrix} \dot{\mathbf{p}}_b^w \\ \dot{\mathbf{q}}_b^w \\ \dot{\mathbf{v}}_w \\ \dot{\boldsymbol{\omega}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{v}_w \\ \mathbf{q}_b^w \cdot \begin{bmatrix} \mathbf{0} \\ \frac{\boldsymbol{\omega}_b}{2} \end{bmatrix} \\ \frac{1}{m}(\mathbf{q}_b^w \odot (f_r \mathbf{e}_z) - g_w \mathbf{e}_z) \\ \mathbf{J}^{-1}(\boldsymbol{\tau}_r - \boldsymbol{\omega}_b \times \mathbf{J}\boldsymbol{\omega}_b) \end{bmatrix} \tag{1}$$

where $\mathbf{p}_b^w = [x, y, z]^T$, $\mathbf{q}_b^w = [q_0, q_x, q_y, q_z]$ and $\mathbf{v}_w = [v_x, v_y, v_z]^T$ are the position vector, the unit attitude quaternions, and the inertial velocity vector of the quadrotor's center-of-mass (COM) in the world frame, respectively. $\boldsymbol{\omega}_b = [\omega_x, \omega_y, \omega_z]^T$ is the angular velocity in body frame; $\odot$ represents quaternion rotation, $\mathbf{e}_z = [0,0,1]^T$, and g_w is the gravity constant. The collective thrust and the three-dimensional torque produced by the rotor forces

f_i, $\forall i \in [1,4]$ are denoted as f_r and $\boldsymbol{\tau}_r = [\tau_x, \tau_y, \tau_z]^T$, respectively. The matrix $\mathbf{J}$ is the inertia of the UAV. The full state and control input of the quadcopter are defined as $\mathbf{x}_q = [\mathbf{p}_b^w, \mathbf{v}_w, \mathbf{q}_b^w, \boldsymbol{\omega}_b]^T$ and $\mathbf{u} = [f_1, f_2, f_3, f_4]^T$, respectively.

Gate Dynamics Model: The dynamic rectangular gate is modeled with constant velocity nominal dynamics:

$$\dot{\mathbf{p}}_g^w = \mathbf{v}_g^w, \quad \dot{\omega}_g = \ddot{\theta}_g = 0 \tag{2}$$

where $\mathbf{p}_g^w = [x_g, y_g, z_g]^T$ and $\mathbf{v}_g^w = [v_{xg}, v_{yg}, v_{zg}]^T$ are the position vector and linear velocity vector of the gate's COM in the world frame, respectively. Assuming the gate is perpendicular to the ground, it can rotate about an angle θ_g in the $x_g o_g z_g$ plane. ω_g is the angular rate and its rate of change is zero. Thus, gate states are defined as $\mathbf{x}_g = [\mathbf{p}_g^w, \mathbf{v}_g^w, \theta_g]^T$.

B. Decision-based Joint Planning and Control

In our framework, a DRL agent is trained to handle vast high-dimensional observation data, forecasting low-dimensional state information tailored for NMPC. The details of the outline in this section are given in Algorithm 1.

The interaction process between DRL and the dynamic environment can be described by a Markov decision process (MDP), denoted as $MDP = (S, A, P, r, \gamma)$. Here, the state space S and action space A are both bounded and continuous. P represents the state transition probability, r is the reward, and $\gamma \in (0,1)$ serves as the discount factor. DRL computes the cumulative reward at any time step k by constructing an action-value function $Q^\pi(s_t, a_t)$ which can be expressed using the Bellman equation [17]:

$$Q^\pi(s_t, a_t) = r(s_t, a_t) + E\left[\gamma \max_a E[Q^\pi(s_{t+1}, a_{t+1})]\right] \tag{3}$$

where s_t and a_t are the state and action variables at time step t. The learning policy π can be parameterized using θ, which is randomly initialized by θ_0. The search and update process of optimal policy will be illustrated in Section II.D.

DRL State Variable: During each training iteration of the DRL agent, the initial position $\mathbf{p}_{b,init}^w$, yaw angle φ_{init} of the UAV, goal position $\mathbf{p}_{b,goal}^w$, and the observed state of the gate $\mathbf{x}_g$ are selected as the state variables, which will be reset randomly.

DRL Action Variable: The key to the integration of DRL with NMPC lies in utilizing the action a_t of DRL as a high-level decision variable $\mathbf{z}$ for the NMPC. In the challenging task of learning to find a trajectory that passes through the center of a moving gate, it is necessary to determine the desired traversal time t_{tra} and traversal state $\mathbf{r}_{tra} = [\mathbf{p}_{b,tra}^w, \boldsymbol{q}_{b,tra}^w]^T$ of the UAV. Therefore, we consider $\mathbf{z} = [\mathbf{r}_{tra}, t_{tra}] \in \mathbb{R}^7$ as the actions output of the DRL for this task.

NMPC Objective Function: After obtaining the traversed high-level decision variable $\mathbf{z} = [\mathbf{r}_{tra}, t_{tra}] \in \mathbb{R}^7$, we employ NMPC for constrained optimal control of the system dynamics. Let $\boldsymbol{\tau} = [\mathbf{x}_k, \mathbf{u}_k]_{k\in 1,\dots,N}$ be the trajectory generated by NMPC and the NMPC controller can be considered as $\boldsymbol{\tau} = \mathrm{NMPC}(\mathbf{z})$. To represent the final states of the UAV hovering at a goal point behind the gate, we use the notation $\mathbf{x}_{goal}$. NMPC is capable of generating an optimal state trajectory $\{\mathbf{x}_k^* | \forall\, k \in [0, N]\}$ that leads towards $\mathbf{x}_{goal}$. Additionally, NMPC determines an optimal sequence of control commands $\{\mathbf{u}_k^* | \forall\, k \in [0, N-1]\}$ over a horizon N that encompasses the planned trajectory $\boldsymbol{\tau}^*$. In the NMPC framework, the location of the gate is regarded as an intermediate waypoint, and the optimal goal is minimizing the sum of five quadratic components as follows:

$$\min_{\boldsymbol{\tau}} J = \min_{\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}} \sum_{k=0}^{N-1} \left(\left\| \mathbf{x}_k - \mathbf{x}_{goal} \right\|_{\mathbf{Q}_x}^2 + \| \mathbf{u}_k - \mathbf{u}_{k-1} \|_{\mathbf{Q}_{\Delta u}}^2 + \| \mathbf{u}_k \|_{\mathbf{Q}_u}^2 + \| \mathbf{x}_k - \mathbf{r}_{tra} \|_{\mathbf{Q}_{tra}}^2 \right) + \| \mathbf{x}_N - \mathbf{x}_{tar} \|_{\mathbf{Q}_{x_N}}^2 \tag{4a}$$

$$s.t.\ \mathbf{u}_{min} \le \mathbf{u} \le \mathbf{u}_{max} \tag{4b}$$

$$\mathbf{x}_{k+1} = \mathbf{x}_k + d_t \hat{f}(\mathbf{x}_k, \mathbf{u}_k), \quad \mathbf{x}_0 = \mathbf{x}_{init} \tag{4c}$$

where N represents the prediction horizon; $\mathbf{u}_{min}$ and $\mathbf{u}_{max}$ denote the lower and upper bounds of the control input vector, respectively. d_t is the sampling time. $\mathbf{x}_{init}$ signifies the initial states. $\mathbf{x}_k$ and $\mathbf{u}_k$ represent the state and control input variables at time k, respectively. $\mathbf{r}_{tra}$ denotes the reference traversal state. $\mathbf{Q}_x \in R^{13\times 13}, \mathbf{Q}_u \in R^{4\times 4}, \mathbf{Q}_{\Delta u} \in R^{4\times 4}$, and $\mathbf{Q}_{tra} \in R^{4\times 4}$ are the weight matrices. Among them, $\mathbf{Q}_x$, $\mathbf{Q}_u$, and $\mathbf{Q}_{\Delta u}$ are time-invariant. The traversal cost matrix $\mathbf{Q}_{tra}$ is defined as follows:

$$\mathbf{Q}_{tra}(t_{tra}, k) = \mathbf{Q}_{max} \exp(-\beta(k d_t - t_{tra})^2) \tag{5}$$

where $\mathbf{Q}_{max} \in R^{4\times 4}$ represents the maximum traversal cost matrix, playing a pivotal role in the formulation by surpassing all other time-invariant cost matrices. $\beta \in \mathbb{R}_+$ denotes the temporal spread of the traversal cost. Assuming that the quadrotor passes through the gate at time step k_{tra} and $k_{tra} d_t = t_{tra}$, we have $\mathbf{Q}_{tra}(t_{tra}, k) \approx \mathbf{Q}_{max}$, indicating that a large penalty is imposed to ensure precise gate tracking. After the quadrotor passes the gate ($k > k_{tra}$), $\mathbf{Q}_{tra}$ will decrease exponentially, resulting in reduced impacts on the NMPC optimization. Therefore, the optimization process in (4) will prioritize the trajectory tracking task and guide the UAV toward the goal point. Finally, we obtain the planned trajectory $\{\boldsymbol{\tau}^*(\boldsymbol{z})\}_{k=0}^N$ that consists of the state vector $\mathbf{x}_{0:N}$ and the control inputs $\mathbf{u}_{0:N-1}$.The first control command $\mathbf{u}_0$ is used for the UAV to fly through the predicted state waypoints.

Algorithm 1: Decision-based Joint Planning and Control

Inputs: Initial position $\mathbf{p}_{b,init}^w$, yaw angle φ_{init}, goal position $\mathbf{p}_{b,goal}^w$, gate state $\mathbf{x}_g$, and initial policy parameters θ_0

Outputs: $\boldsymbol{\tau}^*(\boldsymbol{z})$

Step 1: DRL outputs actions $\mathbf{z} = [\mathbf{p}_{b,tra}^w, \mathbf{q}_{tra}, t_{tra}]$ based on the learning policy $\pi(\theta)$ using (3)

Step 2: Generate an objective function containing high decision variables $\mathbf{z}$ of NMPC to form new optimization problems using (4).

Step 3: Obtain SE(3) planned trajectory $\{\boldsymbol{\tau}^*(\boldsymbol{z})\}_{k=0}^N$ through online optimization of NMPC based on quadrotor dynamic model using (1).

C. Reachability-based Safe Certificate

This section conducts a reachable set safety analysis for the DRL training iteration based on the optimal trajectory $\{\boldsymbol{\tau}^*(\boldsymbol{z})\}_{k=0}^N$, reducing the collision interaction with the environment and improving the safety and convergence speed of the training process. The details of the outline in this section are given in Algorithm 2.

Stochastic Reachability Analysis: In safety control research, safety entails the continuous fulfillment of specific state constraints [19]. This persistent security is achievable only within a subset of the state space known as the reachable set [20], where an optimal maximum feasible set exists for a given environment. Therefore, this study will conduct a stochastic reachability analysis on the optimal states obtained by NMPC based on the uncertainty of motion and perception in the real environment.

We assume that the robot's motion and sensing models are defined as follows [20]:

$$\begin{aligned} x_{k+1} &= f(x_k, u_k) + w_{k+1} \\ &= Ax_k + Bu_k + w_{k+1} \end{aligned} \tag{6}$$

$$z_k = h(x_k) + v_k \tag{7}$$

where x_k is the state vector, u_k is the control input vector, and z_k is the measurement vector at time step k. After linearizing the motion function $f(\cdot)$, A is the state transition model and B is the control-input model. z_k can originate from various sensors for localization measurements. The functions $h(\cdot)$ pertains to motion aspect. The motion model error vector w_{k+1} and the sensing model error vector v_k follows a zero-mean Gaussian distribution.

To represent these uncertainties in the motion and sensing models, we utilize the probabilistic zonotope set, consisting of a bounded component represented by a zonotope and a stochastic component represented by a Gaussian distribution. A zonotope P is defined as follows [21]:

$$P = \left\{ x_k \in R^n | x_k = c_{x_k} + \sum_{i=1}^{p} \beta_i g_{x_k}^{\{i\}}, -1 \le \beta_i \le 1 \right\} \tag{8}$$

where x_k is the state vector; n is the zonotope dimension; c_{x_k} is the zonotope center; β_i is the coefficient of each generator; $g_{x_k}^{\{i\}}$ is the i th zonotope generator matrix of bounded uncertainty. In the three-dimensional space of UAV, $p = 3$ and β_i must be between -1 and 1. By forming β_i into the coefficient column vector β, P can be simplified as:

$$P = \{ x_k \in R^n | x_k = c_{x_k} + G\beta \} \tag{9}$$

where G is the generator matrix of $n \times p$.

We use Minkowski sum ($\oplus$) [22] and linear transformations to propagate these sets, akin to how states are advanced, through the closed-loop dynamics model derived from equation (6):

$$\begin{aligned} \begin{bmatrix} x_{k+1} \\ \hat{x}_{k+1} \end{bmatrix} = \begin{bmatrix} \check{x}_{k+1} \\ \check{x}_{k+1} \end{bmatrix} \oplus \emptyset_k \begin{bmatrix} x_k - \check{x}_k \\ \hat{x}_k - \check{x}_k \end{bmatrix} \oplus \begin{bmatrix} I \\ L_{k+1}C_{k+1} \end{bmatrix} w_{k+1} \\ \oplus \begin{bmatrix} O \\ L_{k+1} \end{bmatrix} v_{k+1} \oplus \begin{bmatrix} I \\ L_{k+1}C_{k+1} \end{bmatrix} I_k^{\bar{h}} \end{aligned} \tag{10}$$

where $\check{x}_k$ is the reference trajectory; $\hat{x}_{k+1}$ is the estimated trajectory; L_{k+1} denotes the sets of linearization errors corresponding to each step; I and O represent identity and zero matrices, respectivel; C_{k+1} is the system measurement matrix; and $I_k^{\bar{h}}$ is linearization error. Based on (10), we can calculate the forward reachable set (FRS) of a given motion trajectory and express it in the form of a probabilistic zonotope. The state vector of $\{\boldsymbol{\tau}^*(\mathbf{z})\}_{k=0}^N$ can be written as the confidence reachable set including uncertainty:

$$\mathbf{x}_k^\sigma \leftarrow \mathbf{x}_{1:N} = [\mathbf{x}(0|k)^T, \mathbf{x}(k+1|k)^T, \cdots, \mathbf{x}(k+N|k)^T]^T \tag{11}$$

where $\mathbf{x}_k^\sigma$ represents confidence sets along the planned trajectory using stochastic reachability analysis and σ is scalar value based on specified collision-safety probability.

*Safe-assured DRL Reward Function: S*ince rolling out the policy naively may lead to collisions, we would check whether the confidence reachable set $\mathbf{x}_k^\sigma$ within the predictive time horizon includes fault-safe operations. The criterion for checking is to ensure $\mathbf{x}_k^\sigma \cap \mathbf{x}_k^{unsafe} = \emptyset$, where $\mathbf{x}_k^{unsafe}$ represents the union of zones constrained by obstacles perceived by the UAV as well as the drone's hovering state. Therefore, the proactive collision penalty α^* of reachsafe $\{\boldsymbol{\tau}^*(\mathbf{z})\}_{k=0}^N$ is defined as:

$$\alpha^* = \begin{cases} 0, \left(\mathbf{x}_k^\sigma \cap \mathbf{x}_k^{unsafe}\right) = \emptyset \\ \alpha, otherwise \end{cases} \tag{12}$$

where α is the specific proactive collision penalty.

If the above conditions are not met, the interaction between the agent and the environment will be directly terminated, and output a proactive collision penalty thereby improving the safety of the training process. This shielding strategy is realized by designing the reward function of DRL, which is crafted to assess the quality of $\boldsymbol{\tau}^*(\mathbf{z}) = \{x_k^*(\mathbf{z})\}_{k=0}^N$, and can be formalized as follows:

$$r\left(\boldsymbol{\tau}^*(\mathbf{z})\right) = r_{max} - Q_1 J_{goal}\left(\boldsymbol{\tau}^*(\mathbf{z})\right) - Q_2 J_{\text{coll}}\left(\boldsymbol{\tau}^*(\mathbf{z})\right) - \alpha^* \tag{13a}$$

$$J_{goal} = \sum_{k=N-n}^{N} \| \mathbf{p}_b^w(\mathbf{z}) - \mathbf{p}_{b,goal}^w \|_2^2 \tag{13b}$$

$$J_{\text{coll}} = \sum_{i=1}^{4} 2\varepsilon * d_i + \varepsilon^2 \tag{13c}$$

where J_{goal} and J_{coll} are the goal penalty term and collision penalty term, respectively; ε is a safety margin and d_i is the shortest distance between the UAV and the gate border; Q_1 and Q_2 are the weighting coefficients. The relationship between Q_2 and α^* is as follows:

$$Q_2 = \begin{cases} \xi, & \text{if } \alpha^* \neq 0 \\ \zeta, & otherwise \end{cases} \tag{14}$$

where ζ is a very large value (e.g., 100), while ξ is a very small value (e.g., 0.00001).

Algorithm 2: Reachability-based Safe Certificate

Inputs: Planned trajectory $\{\boldsymbol{\tau}^*(\boldsymbol{z})\}_{k=0}^N$

Outputs: Reward $r\left(\boldsymbol{\tau}^*(\mathbf{z})\right)$

Step 1: Calculate the $\mathbf{x}_k^\sigma$ of planned trajectory $\{\boldsymbol{\tau}^*(\boldsymbol{z})\}_{k=0}^N$ by using stochastic reachability analysis (10) and (11).

Step 2: Determine whether the $\mathbf{x}_k^\sigma$ intersects with the $\mathbf{x}_k^{unsafe}$ using (12).

Step 3: Calculate the reward function $r\left(\boldsymbol{\tau}^*(\mathbf{z})\right)$ of the DRL agent using (13).

Step 4: If $\alpha^* \neq 0$, terminates the current episode, otherwise outputs the belief space action directly to interact with the environment.

D. DRL Policy Search Update

After obtaining the observation state and reward function $r\left(\boldsymbol{\tau}^*(\mathbf{z})\right)$ from the operating environment, the optimal learning policy π^* can be determined based on the expectation of maximizing the cumulative reward in the Markov environment. This policy can search for the action a_t that maximizes the cumulative reward in a specific state s_t, and it can be represented as:

$$\pi^*(\theta) = arg\max_a E\left[\sum_{t=0}^{N} \gamma^t r(s_t, a_t)\right] \qquad (15)$$

The processes in Sections II.B to II.D are repeated until (15) is satisfied. At this stage, DRL obtained the highest expectation of cumulative rewards with the policy $\pi^*(\theta)$.

III. Simulation Experiments

A. Environment Configuration

To validate the effectiveness of the proposed method, the carefully designed simulated environment is employed with a UAV flying through a fast-moving gate in an unknown dynamic environment, as demonstrated in Fig. 2. The reasons that only the simulated experiment is conducted in this paper are listed below:

(1) *Flexibility in collecting the training dataset*: Real-world experiments involving UAV traversal of moving gates entail inherent risks and safety concerns. The continuous collision of UAVs with the moving gates would significantly increase the cost of training. Additionally, DRL is a data-intensive approach that requires extensive trial-and-error and parameter adjustments. Conducting experiments in a simulation environment accelerates the process of algorithm validation and optimization. The effectiveness of the proposed method will be validated using real experiments in the future to further validate the generality of the proposed method.

(2) *Flexibility in controlling the noise model of the measurement and environments*: By conducting experiments in a simulation environment, it is possible to better control other extraneous variables such as weather conditions, thereby maintaining the consistency and reliability of the experiment. Simulation experiments reduce resource requirements and increase the diversity of gate motion variations.

In the simulation experiments, we aim to bolster the generalization capability of the proposed method in diverse environments. To achieve this, we introduce variations in the initialization of the gate's motion speed, the UAV's initial states, and the goal positions. Specifically, the parameters of the experiment and the physical parameters of the UAV, such as arm length l, are provided in Table I. The dynamic gate's velocity $\mathbf{v}_g^w$ is sampled from a Gaussian distribution. The initial angular rate of gate $\theta_{g,init}$, the UAV's yaw angle are sampled from the uniform distribution. The UAV's initial positions $\mathbf{p}_{b,init}^w$ and goal positions $\mathbf{p}_{b,goal}^w$ are initialized and subjected to uniform random perturbations in each training iteration, while the remaining UAV initial states were fixed at zeros. Furthermore, to simulate the perception uncertainty in a real environment, we introduce additional Gaussian noise with a mean of zero and a specific standard deviation to the DRL observation states, thereby capturing random perturbations in the state.

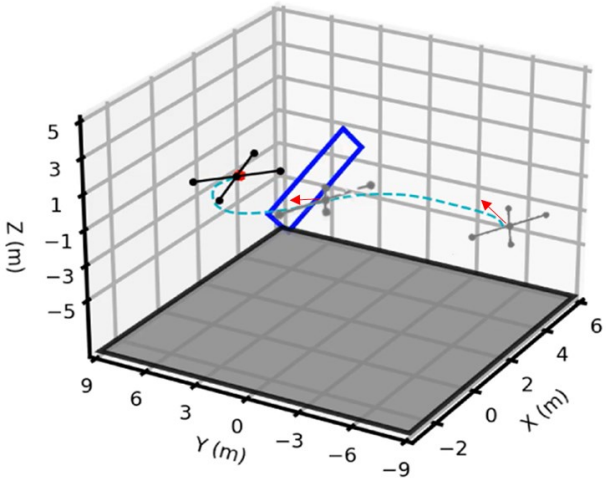


Fig. 2. Illustration of the scenario. The UAV traverses a moving gate (solid blue rectangle) from a random starting point and reaches the goal point (solid red dot), and the dotted line indicates one planned route. The physical parameters of the UAV, such as arm length l, are provided in Table I.

TABLE I. Experiment Parameters

Parameters	Value
$\mathbf{v}_g^w$	$N(\boldsymbol{\mu}, \boldsymbol{\sigma}^2)\ m/s$
$\theta_{g,init}$	$U\left(-\frac{\pi}{2}, \frac{\pi}{2}\right)\ rad/s$
$\mathbf{p}_{b,init}^w$	$[1, -8, 1]^T\ m^*$
$\mathbf{p}_{b,goal}^w$	$[1, 8, 0]^T\ m^*$
$\psi_{b,init}^w$	$U(-0.1, 0.1)\ rad$
Gaussian observation noise	$N(\mathbf{0}, \boldsymbol{\Sigma}_v)^{**}$
σ	0.3
g_w	$9.85\ m/s^2$
m	$0.5\ kg$
l	$0.4\ m$
$\mathbf{J}$	$[0.003, 0.003, 0.004]^T\ kg \cdot m^2$

* $\mathbf{p}_{b,init}^w$ and $\mathbf{p}_{b,goal}^w$ are perturbed by $U(-2, 2)$

**The $\boldsymbol{\Sigma}_v = 0.01^2 \times \mathbf{I}$ is the covariance matrix of $\boldsymbol{v}_k$, $\mathbf{I}$ is identity matrix.

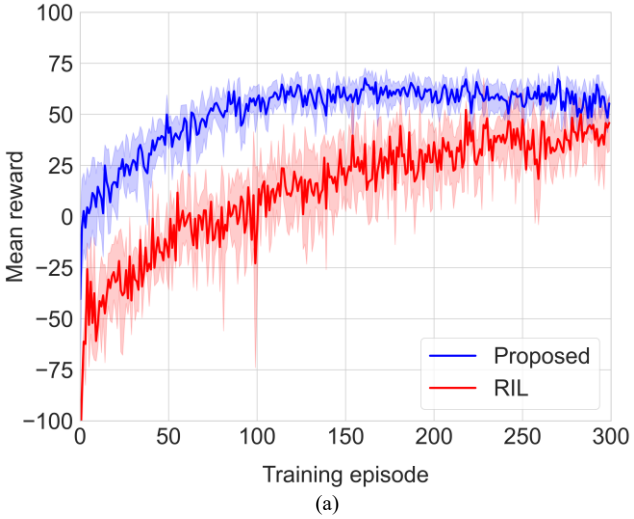
In our DRL setup, we utilize the actor-critic architecture [23] as our DRL agent. Each network comprises 4 layers, featuring hidden layers of size 256 and leaky ReLU [24] activations. The learning rate for each network is set as 10^{-4},

while the discount factor γ is set as 0.99. Training spans 300 episodes, employing the Adam optimizer. For NMPC, we select $d_t = 0.1s$ with a prediction horizon of $N = 50$. We employ CasADi [25] with IPOP [26] as the solver for the numerical optimization problem. The algorithm and related control procedures are implemented in Python, where PyTorch [27] is utilized for constructing the neural network. To ensure the reliability of the simulation experiments, each method undergoes seven training attempts with distinct random seeds.

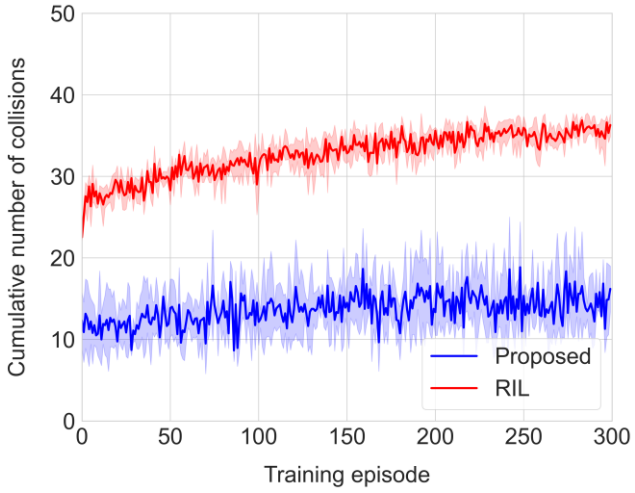
B. Results and Analysis

We evaluate the proposed method from two perspectives: training performance and task completion ability. The RIL framework [14] is employed as the baseline method, which breaks down the training process into two manageable subproblems. Initially, RIL uses RL to train a DNN for navigating static gates, followed by supervised learning to train a new DNN capable of imitating and adapting to non-static scenarios.

We first compare the training performance of the proposed method and the RIL-based method. As illustrated in Fig. 3-(a), the proposed method demonstrates a faster learning speed and achieves significantly higher asymptotic rewards compared to the baseline. This finding reveals the outstanding optimization capability of the proposed method during the training process. In Fig. 3-(b), the cumulative number of collisions of the proposed method remains consistently lower than the RIL. The proposed method results in 18 cumulative collision incidents, whereas the baseline method experiences up to 35 conflicts, as listed in Table II. This further validates the advantages of the proposed strategy in improving safety. Overall, the proposed method provides a higher level of safety during training, attributed to its safety stochastic reachability certificate.



(a)

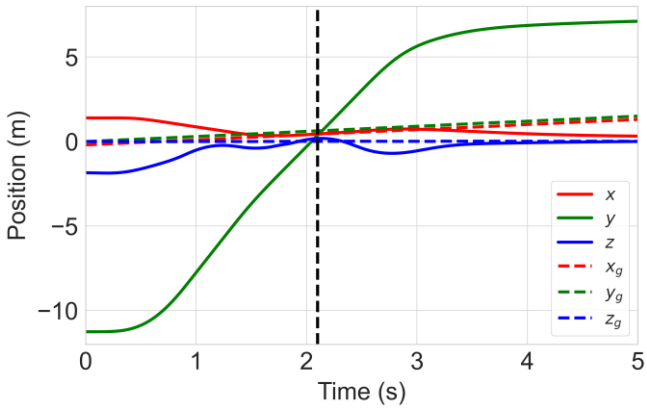


(b)

Fig. 3. Training results comparison between the proposed method and the RIL-based method (solid lines for average values and error bars for standard deviations): (a) Episode reward during the training process; (b) Cumulative collision count of the UAV throughout the training process.

To assess the task completion capabilities of the well-trained proposed method, we conduct tests. Fig. 4-(a) and Fig. 4-(b) depict the trajectories of the UAV and the moving gate when $\boldsymbol{\mu} = [0.3, 0.3, -0.3]^T m/s$ and $\boldsymbol{\sigma} = [0.1, 0.1, 0.1]^T m/s$. As the UAV approaches the gate, it follows the gate's movement along the x-axis (in red) and the z-axis (in blue). At this point, the UAV's position and orientation in the $x_g o_g z_g$ plane closely match those of the gate's center of mass, ultimately ensuring the traversal safely.

Fig. 4-(c) displays the expected traversal time learned by DRL. It is evident that as the traversal time decreases to zero, the UAV and gate intersect in the y-direction. This observation validates that the optimized UAV's position and orientation gradually align with the gate's center only when the UAV approaches the gate, i.e., when the traversal time decreases to 0. Such a feature has significant implications for real-world deployment, as quadcopters must follow the gate's center closely to resist environmental disturbances and ensure system safety when approaching the gate. Additionally, Fig. 4-(a) also illustrates the UAV's position at the final moment ($[1.04, 8.16, 0.10]^T$) which can be compared with the goal position to validate the overall task completion effectiveness.



(a)

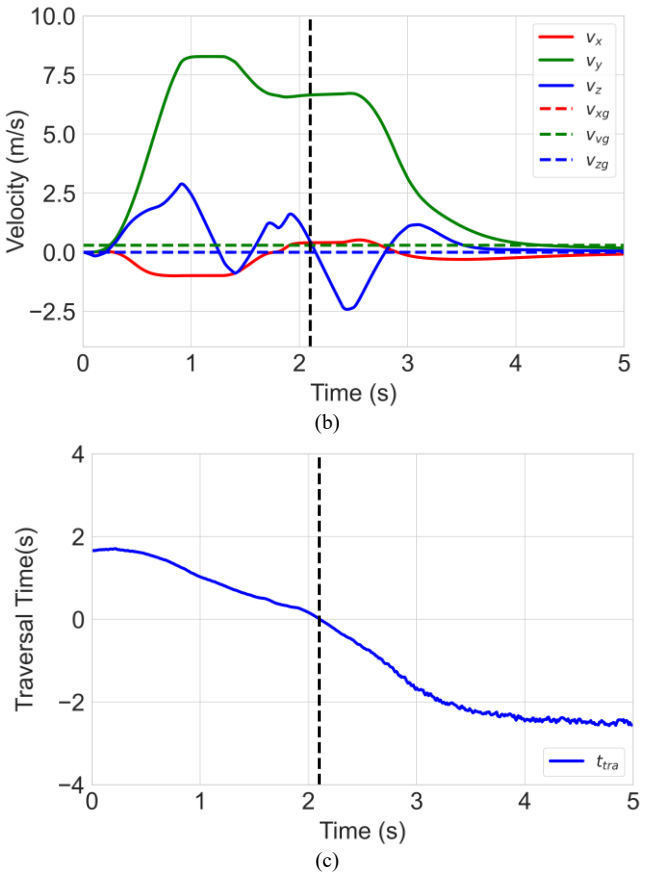


Fig. 4. Evaluation of test results of UAV passing through moving gate in one case: (a) Trajectories of the UAV (solid line) and the moving gate (dashed line); (b) The velocity of the UAV (solid line) and the moving gate (dashed line); (c) The predicted traversal time variable. The vertical dashed black lines indicate the time when the UAV is traversing the gate.

For a detailed numerical comparison between the proposed method and the baseline method, one can refer to Table II. Further demonstration tests can be found in the video:https://www.youtube.com/watch?v=IvikfaYtCz4 showcasing a series of safe and successful traversal actions. Overall, in the studied scenarios, the proposed strategy exhibited more favorable safety and performance during both training and testing.

TABLE II. Performance evaluations of listed methods

Method	Proposed	RIL
Mean reward ↑	**62.17**	49.09
Convergence episode ↓	**112**	266
Cumulative collision number ↓	**18**	35
Traversal distance error (m)	(0.11,0.12,0.06)	–
Traversal velocity error (m/s)	(0.22,6.52,0.35)	–
Target distance error (m)	(0.04,0.16,0.10)	–

IV. Conclusion

This paper proposes a safe-assured learning-based deep SE(3) motion joint planning and control approach for UAVs. Specifically, we integrate DRL based on stochastic reachability certification with model-based NMPC to achieve high-level safety decision-making, online complex motion planning and control for UAVs in cluttered environments. Actions satisfying confidence intervals are then selected for real interaction with the environment to ensure safety. The proposed method addresses the issues of conservatism or potential danger caused by approximation errors and observation noise in optimization problems and strategy execution. In scenarios involving agile and safety flight through a moving gate, our proposed method is compared with RIL, demonstrating its ability to balance performance and safety.

In the future, we will explore dynamic confidence interval settings, reduce conservatism in reachability analysis, and undertake further validation through real-world experiments.

References

[1] C. Maget, S. Gutmann and K. Bogenberger, "Model-based Evaluations Combining Autonomous Cars and a Large-scale Passenger Drone Service: The Bavarian Case Study," *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1-6.

[2] P. Yang and W. Wen, "Tightly Joining Positioning and Control for Trustworthy Unmanned Aerial Vehicles Based on Factor Graph Optimization in Urban Transportation," *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC),* 2023, pp. 3589-359.

[3] A. Ranjbar, S. Hornauer, J. Fredriksson, S. X. Yu and C. -Y. Chan, "Safety Monitoring of Neural Networks Using Unsupervised Feature Learning and Novelty Estimation," *in IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 3, pp. 711-721, 2022.

[4] Z. Han, Z. Wang, N. Pan, Y. Lin, C. Xu and F. Gao, "Fast-Racing: An Open-Source Strong Baseline for SE(3) Planning in Autonomous Drone Racing," *in IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 8631-8638, 2021.

[5] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning", *Nature*, 620(7976): 982-987, 2023.

[6] Z. Yan, A. R. Kreidieh, E. Vinitsky, A. M. Bayen and C. Wu, "Unified Automatic Control of Vehicular Systems With Reinforcement Learning," *in IEEE Transactions on Automation Science and Engineering*, vol. 20, no. 2, pp. 789-804, 2023.

[7] M. Kaymaz, R. Ayzit, O. Akgün, K. Atik, M. Erdem, B. Yalcin, G. Cetin and N. Ure, "Trading-Off Safety with Agility Using Deep Pose Error Estimation and Reinforcement Learning for Perception-Driven UAV Motion Planning", *Journal of Intelligent & Robotic Systems*, 110(2): 1-17, 2024.

[8] A. Shetty, G. Gao, "Predicting State Uncertainty Bounds Using Non-Linear Stochastic Reachability Analysis for Urban GNSS-Based UAS Navigation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 9, pp. 5952-5961, 2021.

[9] A. Shetty, A. Dai, A. Tzikas and G. Gao, "Safeguarding Learning-Based Planners Under Motion and Sensing Uncertainties Using Reachability Analysis," *2023 IEEE International Conference on Robotics and Automation (ICRA),* pp. 7872-7878, 2023.

[10] M. Selim, A. Alanwar, S. Kousik, G. Gao, M. Pavone and K. H. Johansson, "Safe Reinforcement Learning Using Black-Box Reachability Analysis," *in IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10665-10672, 2022.

[11] J. Fischer, M. Steiner, Ö. Ş. Taş and C. Stiller, "Safety Reinforced Model Predictive Control (SRMPC): Improving MPC with Reinforcement Learning for Motion Planning in Autonomous Driving," *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC),* pp. 2811-2818, 2023.

[12] J. Lubars, H. Gupta, S. Chinchali, L. Li, A. Raja and R. Srikant, "Combining Reinforcement Learning with Model Predictive Control for On-Ramp Merging," *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pp. 942-947, 2021.

[13] Y. Song and D. Scaramuzza, "Policy Search for Model Predictive Control With Application to Agile Drone Flight," *in IEEE Transactions on Robotics*, vol. 38, no. 4, pp. 2114-2130, 2022.

[14] Y. Wang, B. Wang, S. Zhang, H. W. Sia and L. Zhao, "Learning Agile Flight Maneuvers: Deep SE(3) Motion Planning and Control for Quadrotors," *2023 IEEE International Conference on Robotics and Automation (ICRA),* pp. 1680-1686, 2023.

[15] T. Zhang, G. Kahn, S. Levine and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," *2016 IEEE International Conference on Robotics and Automation (ICRA),* pp. 528-535, 2016.

[16] P. Foehn, A. Romero, D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight", *Science Robotics,* vol. 6, no. 56, 2021.

[17] D. Hu, Y. Zhang, "Deep reinforcement learning based on driver experience embedding for energy management strategies in hybrid electric vehicles", *Energy Technology*, 2022.

[18] C. Philippe, L. Adouane, B. Thuilot, A. Tsourdos and H. -S. Shin, "Safe and Online MPC for Managing Safety and Comfort of Autonomous Vehicles in Urban Environment," *2018 21st International Conference on Intelligent Transportation Systems (ITSC),* pp. 300-306, 2018.

[19] E. Asarin, T. Dang, A. Girard, "Reachability analysis of nonlinear systems using conservative approximation", *Hybrid Systems: Computation and Control*, pp. 20–35, 2003.

[20] F. Valenti, D. Giaquinto, L. Musto, A. Zinelli, M. Bertozzi and A. Broggi, "Enabling Computer Vision-Based Autonomous Navigation for Unmanned Aerial Vehicles in Cluttered GPS-Denied Environments," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3886-3891, 2018.

[21] N. Kochdumper and M. Althoff, "Sparse Polynomial Zonotopes: A Novel Set Representation for Reachability Analysis," *in IEEE Transactions on Automatic Control*, vol. 66, no. 9, pp. 4043-4058, 2021.

[22] K. Agarwal, E. Flato, D. Halperin, "Polygon decomposition for efficient construction of Minkowski sums", *Computational Geometry*, 21(1-2): 39-61, 2002.

[23] Q. Liu, L. Shi, L. Sun, J. Li, M. Ding and F. Shu, "Path Planning for UAV-Mounted Mobile Edge Computing With Deep Reinforcement Learning," *in IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5723-5728, 2020.

[24] Z. Jian, Z. Yan, X. Lei, Z. Lu, B. Lan, X. Wang and B. Liang, "Dynamic Control Barrier Function-based Model Predictive Control to Safety-Critical Obstacle-Avoidance of Mobile Robot," *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 3679-3685, 2023.

[25] M. Fevre, P. M. Wensing and J. P. Schmiedeler, "Rapid Bipedal Gait Optimization in CasADi," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3672-3678, 2020.

[26] T. Zhang, G. Kahn, S. Levine and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," *2016 IEEE International Conference on Robotics and Automation (ICRA),* pp. 528-535, 2016.

[27] D. Hu, C. Huang, J. Wu and H. Gao, "Pre-trained Transformer-Enabled Strategies with Human-Guided Fine-Tuning for End-to-end Navigation of Autonomous Vehicles," *arXiv preprint arXiv*:2402.12666, 2024.