Learning Safe, Optimal, and Real-Time Flight Interaction With Deep Confidence-Enhanced Reachability Guarantee

Yuanyuan Zhang, *Graduate Student Member, IEEE*, Yingying Wang, Penggao Yan, and Weisong Wen, *Member, IEEE*

***Abstract*—In the low-altitude economy, ensuring the safe and agile flight of unmanned aerial vehicles (UAVs) in dynamic obstacle environments is essential for expanding interactive applications like parcel delivery. While deep reinforcement learning (DRL) shows promise for UAV motion planning and control, its trial-and-error exploration often struggles to ensure both agility and safety, especially under uncertain observational noise. Therefore, this paper proposes a deep confidence-enhanced reachability policy optimization (DCRPO) framework. By integrating safe DRL with nonlinear model predictive control (NMPC), DCRPO achieves high-level safety decisions, complex real-time joint planning and control for UAVs. Furthermore, we develop a deep confidence-enhanced reachability guarantee that constructs a set of stochastically forward-reachable planned trajectories under uncertainty, enabling robust safety collision probability certifications. This safe reachability mechanism adaptively selects belief space actions from planned actions to interact with the environment, further enhancing safety and reducing training time. In extensive experiments of UAVs traversing a fast-moving rectangular gate, the proposed method outperforms other state-of-the-art baseline methods under varying environments in terms of operational robustness. Furthermore, the proposed method significantly reduces overall collision violations and training time, greatly improving both training safety and efficiency. The demonstration video (https://youtu.be/7xkp9U7FSJg) and the source code (https://github.com/ZyyFLY/DCRPO) are also provided.**

***Index Terms*—Deep reinforcement learning, deep confidence-enhanced reachability guarantees, joint planning and control, unmanned aerial vehicles.**

I. Introduction

UNMANNED aerial vehicle (UAVs) are increasingly applied in critical low-altitude missions, including surveillance [1] and delivery [2]. As illustrated in Fig. 1, these tasks require UAVs to perform agile, safe, and real-time operations within environments containing dynamic obstacles, such as rotating gates [3] or unpredictable external disturbances [4]. Conventional UAV trajectory planning and control methods, primarily model-based [5], [6] or rule-based [7], often struggle to address the complexity and uncertainty of dynamic real-world settings [8]. Model predictive control (MPC) [9] and its nonlinear extension (NMPC) [10] enable real-time optimization based on predicted future states. However, NMPC lacks adaptive learning mechanisms for changing environments, which may lead to conservative behavior and degraded system performance [11]. Moreover, error accumulation in modular model-based methods can further reduce accuracy, underestimate safety risks, and decrease overall efficiency [12].

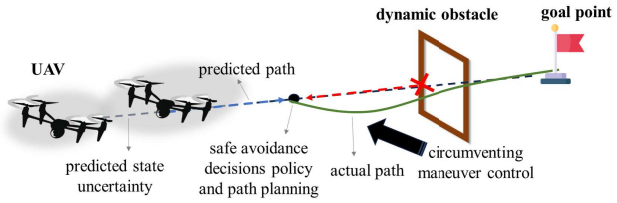


Fig. 1. Schematic diagram of the challenges of joint planning and control of UAVs in safety-critical scenarios involving dynamic obstacles or unpredictable external conditions.

Moreover, deep reinforcement learning (DRL) has enhanced UAVs decision-making by enabling optimal policy learning through environmental interaction [13]. DRL integrates planning and control with minimal prior knowledge, demonstrating superior performance in tasks such as drone racing through multiple static gates [14]. However, DRL's end-to-end training via trial-and-error can result in frequent failures during agile flight, raising safety concerns for both users and systems [15]. Observation noise in dynamic environments further increases this risk, challenging the safety of DRL in safety-critical applications and motivating research into learning methods with integrated safety guarantees into the learning pipeline [16].

Recent studies have increasingly focused on integrating DRL with model-based methods such as control barrier functions (CBF) and NMPC, exemplified by DRL-CBF [17] and deep reinforcement-imitation learning (DRIL) [18]. This integration seeks to combine the interpretability, stability, and constraint satisfaction of model-based control with the flexibility and adaptability of DRL in complex, uncertain environments. By introducing explicit safety constraints, these approaches enhance both training efficiency and operational

Received 8 October 2024; revised 22 August 2025; accepted 27 September 2025. This work was supported in part by Hong Kong Innovation and Technology Fund-Innovation and Technology Support Program (ITF-ITSP) under the Project "Safety-Certified Multi-Source Fusion Positioning for Autonomous Vehicles in Complex Scenarios (ZPE8)." The Associate Editor for this article was R. Meneguette. *(Corresponding author: Yingying Wang.)*

The authors are with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong (e-mail: ying5wang@polyu.edu.hk).

Digital Object Identifier 10.1109/TITS.2025.3616580

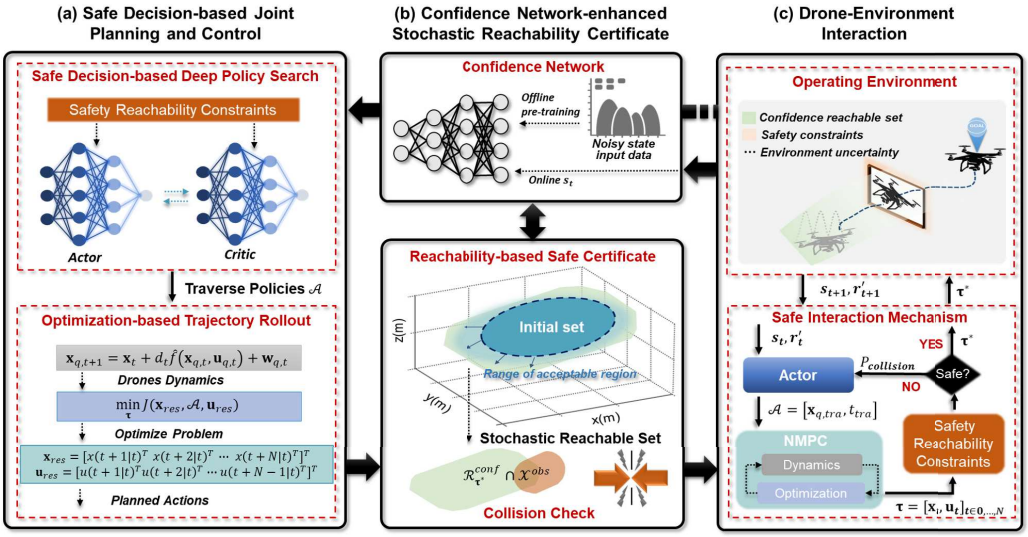


Fig. 2. The overview of the proposed method, including three parts: (a) safe decision-based joint planning and control; (b) confidence network-enhanced stochastic reachability guarantees; and (c) drone-environment interaction loop.

safety. However, DRL-CBF relies on handcrafted constraints, and DRIL, while improving NMPC adaptability, remains dependent on prior knowledge or lacks adaptive stochastic safety certification under uncertainty. Furthermore, the safety mechanisms of these methods are generally limited to local policy adjustments and do not provide system-level guarantees against cumulative risk during long-term operation.

To address these limitations, this paper proposes a deep confidence-enhanced reachability policy optimization (DCRPO) framework, integrating safe DRL with model-based NMPC, as shown in Fig. 2. The main innovation is a confidence-adaptive stochastic reachability certification that evaluates and filters planned actions prior to execution, enabling certified safe exploration without reliance on accurate models or handcrafted constraints. This framework bridges high-level decision-making with low-level optimization, supporting safe, optimal, real-time UAV navigation in uncertain, dynamic environments. The framework utilizes high-level safety decisions from DRL and low-level optimization from NMPC, and leverages a self-supervised deep confidence network (SCNet) to enhance stochastic reachability guarantees, proactively detecting collision risks and reducing training violations. Collaborative optimization between SCNet and the DRL agent ensures alignment with safety objectives and enhances robustness to environmental uncertainty. Experimental results demonstrate that DCRPO outperforms existing learning-based and model-based safe navigation methods in both safety and efficiency.

The contributions of this paper are listed as follows:

(1) This paper proposes a safe decision-based joint planning and control (SD-PC) framework by combining safe DRL for high-level safe policy search with the model-based NMPC for low-level online rolling optimization. This enables safe, optimal, and real-time UAV flight interactions in complex dynamic environments.

(2) This paper develops a deep confidence-enhanced stochastic reachability certificate, utilizing SCNet to dynamically estimate state confidence and improve the reliability of stochastic reachability analysis under uncertainty. This certificate enhances the safety of state predictions and evaluates potential risks at each decision step during training.

(3) The effectiveness of the proposed framework is validated through experiments involving UAV traversing through a fast-moving gate to reach the goal destination. The comparative results demonstrate a significant reduction in safe DRL training time violations, indicating that the proposed framework facilitates efficient, high-performance UAV flight interactions while ensuring safety.

The remainder of this paper is organized as follows. In Section II, the related work is introduced, while Section III provides a detailed description of the DCRPO framework. Following this, Section IV outlines the experimental setup, including dynamic obstacle scenarios and evaluation metrics. The results and analysis are presented in Section V. Finally, the main conclusions and some future work directions are discussed in Section VI.

II. Related Work

A. Learning-Based Safe Navigation Control

Safe DRL seeks to optimize task performance while ensuring compliance with safety constraints during both training and deployment [19]. Existing methods are generally classified into safe model-free DRL (SMFRL) and safe model-based DRL (SMBRL). SMFRL methods handle constrained Markov decision processes (CMDP) by maintaining policy costs within specified limits, often using Lagrange multipliers to convert constrained problems into unconstrained forms [20], [21]. Safety can also be reinforced through reachability analysis [22], expert demonstrations [23], or safety certification ensuring trajectories remain within safe sets. However, SMFRL may still cause unsafe interactions during training, rely heavily on human priors, and suffer from scalability limitations due to overly conservative policies. In contrast, SMBRL improves

TABLE I
COMPARISONS WITH THE EXISTING LITERATURE

Category	Reference	Prior knowledge	Adaptive safety guarantee	Environment model	Online optimization	Sample efficiency	Disturbance robustness	Main differences with ours
SMFRL	[17], [19], [20], [21], [22], [23]	✓	✗	✗	✗	Low	Low	Rely on hand-crafted constraints with limited efficiency and generalization.
SMBRL	[24], [25], [26]	✗	✗	✓	✗	High	Medium	Sensitive to model errors; lack real-time policy adaptation and epistemic safety modeling.
Combined DRL and model-based methods	[18], [27], [28], [29], [30]	✓	✗	✗	✓	Medium	Medium	Lack adaptive safety certification under uncertainty or enable certified safe exploration.
Ours (DCRPO)	——	✗	✓	✗	✓	High	High	——

✓: applicable; ✗: not applicable.

sample efficiency and operational safety by jointly learning policies and environment models [24]. Leveraging probabilistic dynamics models, these methods synthesize virtual data and impose strong penalties on unsafe trajectories [25], [26].

B. Combined Learning-Based and Model-Based Safe Navigation Control

Model-based methods depend on predefined dynamic models, but as tasks and environments become more complex, they require significant manual design, including crafting cost functions and developing specialized planning strategies [12]. Even advanced methods like MPC must rely on online re-planning and manual system tuning. To address these limitations, recent research has focused on integrating model-based controllers with learning-based DRL [27], [28]. This integration enables processing high-dimensional data, reduces manual intervention, and achieves adaptive, optimal control. For instance, Song and Scaramuzza [29] propose a hybrid method using deep neural networks to predict NMPC decision variables, though with limited UAV agility due to fixed attitudes. Wang et al. [18] improve NMPC adaptability through a DRIL framework, but did not address safe exploration during training. Romero et al. [30] combine differentiable MPC with actor-critic RL, achieving real-time, complex UAV control.

In summary, as shown in Table I, the proposed DCRPO framework advances safe learning-based control by introducing an adaptive confidence-aware safety certification mechanism. Unlike SMFRL, DCRPO does not require prior knowledge, employing predictive safety filtering. Compared to SMBRL, it avoids dependence on accurate environment models by leveraging real-time NMPC planning. Furthermore, DCRPO provides system-level safety guarantees beyond traditional hybrid methods, enabling robust and certifiable decision-making in complex, dynamic environments.

III. METHODOLOGY

As shown in Fig. 2, the proposed DCRPO framework consists of three core modules for safe UAV flight interaction in dynamic environments: (a) a safe decision-based joint planning and control module integrating safe DRL for high-level policy selection with NMPC for low-level trajectory optimization; (b) a confidence network-enhanced stochastic reachability certification module, where SCNet dynamically estimates state confidence and certifies candidate actions under uncertainty; and (c) a drone–environment interaction module forming a closed loop between environmental feedback and control, enabling real-time risk assessment and adaptive response to environmental changes. This modular architecture facilitates proactive safety verification, efficient policy optimization, and robust adaptation to rapidly evolving scenarios. Key components and their notations of the DCRPO framework are summarized in Table II.

TABLE II
NOTATIONS USED IN THIS PAPER

Notations	Specifications
$\mathbf{X}$	Traditional bounded zonotope
$\mathcal{X}$	Inflating zonotope
$\mathbf{c}_{\mathbf{X}}$, $\mathbf{c}_{\mathcal{X}}$, $\mathbf{c}^{\text{conf}}$	Traditional / inflating / confidence zonotope center
$\mathbf{G}_{\mathbf{X}}$,$\mathbf{G}_{\mathcal{X}}$, $\mathbf{G}^{\text{conf}}$	Traditional / inflating / confidence generator matrix for bounded uncertainty
$\mathbf{\Sigma}_{\mathcal{X}}$, $\mathbf{\Sigma}^{\text{conf}}$	Inflating / confidence covariance matrix for stochastic uncertainty
$\mathcal{X}^{\text{conf}}$	Confidence zonotope
$\mathcal{X}_{\text{obs}}$	Union of obstacle zones
α	Scaling factor determined by collision confidence threshold
λ_i, $\mathbf{v}_i$	Eigenvalues and eigenvectors of $\mathbf{\Sigma}^{\text{conf}}$
$\mathcal{S}$	State space
$\mathcal{A}$	Action space
$\mathcal{P}$	State transition probability
r	Reward function
c	Cost function
γ	Discount factor
π, π^*	Deterministic / optimal policy function
$\mathbf{s}_t$	State at time step t
$\mathbf{a}_t$	Action at time step t
$Q^\pi(\cdot), Q_c^\pi(\cdot)$	Action / cost value function under policy π
$\mathcal{R}_t$	Reachable set at time t
$\mathcal{R}_{\tau^*}^{\text{conf}}$	Confidence reachable set under planned trajectory τ^*
$\mathbf{w}_t$	System noise at time t
$\mathbf{x}_{q,t}, \mathbf{x}_{g,t}$	UAV / gate state at time t
$\mathbf{x}_{\text{goal}}$	Goal point state
t_{tra}	Desired traversal time
$\mathbf{x}_{q,\text{tra}}$	Desired traversal state
$\mathcal{S}_{\Delta}^{\text{env}}$	Relative environment observation
$\mathcal{S}^{\text{goal}}$	Absolute goal observation
$\mathcal{W}$	Noise zonotope $\mathbf{X}(\mathbf{c}_{\mathcal{W}}, \mathbf{G}_{\mathcal{W}}, \mathbf{\Sigma}_{\mathcal{W}})$
N	NMPC prediction horizon
d_t	Sampling time interval
$\mathbf{Q}_x$, $\mathbf{Q}_u$	State and input cost weights
$\mathbf{Q}_{\Delta u}$	Control variation cost weight
$\mathbf{Q}_{\text{tra}}$	Traversal reference tracking cost matrix
$\mathbf{Q}_{x_N}$	Terminal state cost weight
β	Temporal spread of traversal penalty
ε	Safety margin in collision penalty
$r_{\max}$	Maximum task reward
r_{goal}	Goal deviation penalty
$r_{\text{collision}}$	Collision penalty
$P_{\text{collision}}$	Predictive safety violation penalty
ω_1, ω_2, ω_3	Reward weighting coefficients
$\hat{\sigma}_t$	Empirical confidence value (for SCNet)
$\sigma_{t,i}$	SCNet predicted confidence value
$\mathcal{L}_{\text{offline}}$, $\mathcal{L}_{\text{online}}$	Offline / online training loss for SCNet
θ_π, θ_{conf}	Policy / SCNet parameter set

A. Definitions

This subsection introduces the set representations and the definition of the forward reachable set (FRS) used in this work.

1) Set Representations: We employ probabilistic zonotopes to model both bounded and stochastic uncertainties in system states. Additionally, this representation facilitates efficient closed under Minkowski summation ($\oplus$) and linear transformations [31]. A standard zonotope is defined as [32]:

$$\mathbf{X} = \left\{ \mathbf{x} \in \mathbb{R}^n \middle| \mathbf{x} = \mathbf{c_X} + \sum_{i=1}^{j} \beta_i \mathbf{g}_i, -1 \le \beta_i \le 1 \right\}, \tag{1}$$

where $\mathbf{c_X}$ represents the center, $\mathbf{g}_i$ are the generator defining the zonotope's shape associated with bounded uncertainty, and β_i are coefficient constrained to $[-1, 1]$. In matrix form, the zonotope $\mathbf{X}(\mathbf{c}_X, \mathbf{G}_X)$ can be simplified as

$$\mathbf{X}(\mathbf{c_X}, \mathbf{G_X}) = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{x} = \mathbf{c_X} + \boldsymbol{\beta}\mathbf{G_X}\}, \tag{2}$$

where $\mathbf{G_X} = [\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_j]$ represents the corresponding $n \times j$ generator matrix; n is the system's dimensionality and j is the number of generators.

While traditional zonotopes $\mathbf{X}$ capture bounded uncertainties, they do not account for stochastic uncertainties from environmental factors. To address this, [33] extends an inflating zonotope to include stochastic uncertainties by modelling a probabilistic region as a Gaussian distribution with an uncertain but bounded mean. It is defined as

$$\mathcal{X} = \mathbf{X}(\mathbf{c}_\mathcal{X}, \mathbf{G}_\mathcal{X}, \mathbf{\Sigma}_\mathcal{X}), \tag{3}$$

where $\mathbf{c}_\mathcal{X}$ and $\mathbf{G}_\mathcal{X}$ represent the bounded uncertainty region, and $\mathbf{\Sigma}_\mathcal{X}$ denotes the Gaussian covariance matrix for stochastic uncertainty.

Building upon the probabilistic zonotope, we define a confidence zonotope, which quantifies state uncertainty. The confidence zonotope $\mathcal{X}^{\text{conf}}(\mathbf{c}^{\text{conf}}, \mathbf{G}^{\text{conf}})$ is defined as

$$\mathcal{X}^{\text{conf}}(\mathbf{c}^{\text{conf}}, \mathbf{G}^{\text{conf}}) = \mathbf{X}(\mathbf{c}^{\text{conf}}, \mathbf{G}^{\text{conf}}, \mathbf{\Sigma}^{\text{conf}}), \tag{4}$$

where $\mathbf{c}^{\text{conf}}$ is set to the mean state $\mathbf{c}_\mathcal{X}$ of the system, and the generator matrix $\mathbf{G}^{\text{conf}}$ is constructed as

$$\mathbf{G}^{\text{conf}} = [\mathbf{G}_\mathcal{X}, \alpha\sqrt{\lambda_1}\mathbf{v}_1, \ldots, \alpha\sqrt{\lambda_n}\mathbf{v}_n], \tag{5}$$

where λ_i and $\mathbf{v}_i$ represent the eigenvalues and eigenvectors of the covariance matrix $\mathbf{\Sigma}^{\text{conf}}$. The covariation metric, defined as $\mathbf{\Sigma}^{\text{conf}} = \text{trace}(\mathbf{G}^{\text{conf}})^\top \mathbf{G}^{\text{conf}}$, is used to measure the uncertainty size, reflecting both bounded and stochastic components. The confidence zonotope is used to quantify the probability that the true state lies within a certain region, given both bounded and stochastic (Gaussian) uncertainties in the system. The confidence threshold α (e.g., corresponding to a 95% probability) determines the size of the reachable set used for safety certification.

2) Reachable Set Based on System Dynamics: Given the above representation, we can compute the FRS to measure dynamic uncertainties and enable safe trajectory planning. The FRS at a given time step is the set of all possible states the system can reach, starting from an initial state, given a sequence of control inputs and process noise. For a discrete-time nonlinear control system, the FRS $\mathcal{R}_t = \mathbf{X}(\mathbf{c}_{\mathcal{R}_t}, \mathbf{G}_{\mathcal{R}_t}, \mathbf{\Sigma}_{\mathcal{R}_t})$, at time step t is defined as

$$\mathcal{R}_t = \left\{ \mathbf{x}_t \in \mathbb{R}^n \mid \begin{matrix} \mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t) + \mathbf{w}_t, \\ \forall t = 0, \ldots, n-1 \end{matrix} \right\}, \tag{6}$$

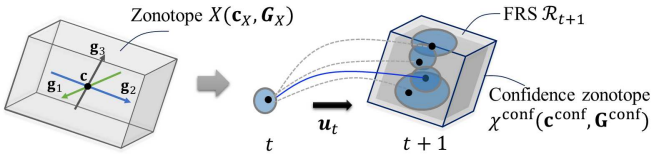


Fig. 3. Visualization examples of a 3D zonotope and a 3D reachable set based on confidence zonotope. **Left:** A zonotope as bounding volume in the three-dimensional space of UAVs. **Right:** Measurement uncertainty along the planned trajectory (blue line) leads to a distribution of possible next states (blue ellipsoid), enclosed by the FRS (gray hypercube) with its boundary defined by the confidence zonotope (dark blue) at a specified confidence level.

where $\mathbf{x}_t \in \mathcal{X} \subset \mathbb{R}^n$ represents the state, starting from the initial state $\mathcal{X}_0$; $\mathbf{u}_t \in \mathcal{U}_t \subset \mathbb{R}^m$ is the control input and $\mathbf{w}_t \in \mathcal{W} \subset \mathbb{R}^n$ is modeled as a probabilistic zonotope $\mathcal{W} = \mathbf{X}(\mathbf{c}_\mathcal{W}, \mathbf{G}_\mathcal{W}, \mathbf{\Sigma}_\mathcal{W})$, representing stochastic uncertainties in the system. Fig. 3 presents visualization examples of a zonotope as bounding volume along with its generator matrix and a one-step 3D FRS based on a confidence zonotope.

Importantly, all stochastic uncertainties in the system (including process and observation noise) are modeled as zero-mean Gaussian random variables with known or estimated covariance. This Gaussianity assumption is critical for the confidence-based reachability analysis and for interpreting α as a collision probability bound.

B. Safe Decision-Based Joint Planning and Control

We propose an SD-PC framework, where a safe DRL agent extracts low-dimensional state information from high-dimensional observations for NMPC compatibility. This allows the UAV to operate safely and efficiently in dynamic environments and enhances policy robustness. At a higher level, safe DRL guides decision-making in complex tasks, such as timing and positioning for traversing through moving gates, ensuring the UAV can adapt its strategy to changing conditions.

The interaction loop between safe DRL and the environment can be described by a Constrained Markov decision process (CMDP) [20], denoted as 6-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, r, c, \gamma \rangle$. Both state space $\mathcal{S}$ and action space $\mathcal{A}$ are bounded and continuous; the initial state set is denoted as $\mathcal{S}_0$; $\mathcal{P}$ represents the state transition probability; $r : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward; c is the cost function defined as $c = 1$ if the state constraint $h(\mathbf{s}_t) \le 0$ is violated and 0 otherwise; $\gamma \in (0, 1)$ serves as the discount factor; a deterministic policy $\pi : \mathcal{S} \mapsto \mathcal{A}$ chooses action variable $\mathbf{a}_t$ at state variable $\mathbf{s}_t$ at time step t. The CMDP computes the cumulative reward by constructing an action-value function $Q^\pi(\mathbf{s}_t, \mathbf{a}_t)$ which can be expressed using the Bellman equation [34]:

$$\begin{aligned} Q^\pi(\mathbf{s}_t, \mathbf{a}_t) &= r(\mathbf{s}_t, \mathbf{a}_t) + \mathbb{E}\left[\gamma \max_{\mathbf{a}_t} \mathbb{E}[Q^\pi(\mathbf{s}_{t+1}, \mathbf{a}_{t+1})]\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(\mathbf{s}_t, \mathbf{a}_t)\right]. \end{aligned} \tag{7}$$

The optimal policy of safe DRL reflects the dual optimization objectives of precision and safety. The learning policy π can be parameterized using θ_π, which is randomly initialized.

Therefore, in the Markov environment, the optimal policy π^* can be determined based on the expectation of maximizing the cumulative reward.

$$\pi^*(\theta_\pi) = \arg\max_{\mathbf{a}_t \in \mathcal{A}} \mathbb{E}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t)] \quad \text{s.t.} \quad \mathbb{E}[Q_c^\pi(\mathbf{s}_t, \mathbf{a}_t)] \le \epsilon\,, \tag{8}$$

where t is the time step, $Q_c^\pi(\mathbf{s}_t, \mathbf{a}_t) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t c(\mathbf{s}_t)\right]$ represents the expected discounted violation probability, and ϵ is a prescribed cost threshold. When $\epsilon = 0$, this formulation also leads to a robust optimal control problem [35].

To achieve a critical property of safety-critical systems in UAV interactions, known as forward invariance, i.e., the continuous satisfaction of safety constraints. During the policy training phase, we utilize DRL based on safety reachability constraints. This method guarantees that the system avoids violations at each time step, resulting in the final optimal policy representing a maximum number of persistently safe states. Consequently, (8) can be reformulated to characterize persistently safe states instead of merely final instantaneous safe states:

$$\pi^*(\theta_\pi) = \arg\max_{\mathbf{a} \in \mathcal{A}} \mathbb{E}[Q^\pi(\mathbf{s}_t, \mathbf{a}_t) \cdot 1_{\mathbf{s}_t \in \mathcal{S}_f^\pi} - Q_c^\pi(\mathbf{s}_t, \mathbf{a}_t) \cdot 1_{\mathbf{s}_t \notin \mathcal{S}_f^\pi}] \quad \text{s.t.} \quad Q_c^\pi(\mathbf{s}_t, \mathbf{a}_t) \le 0, \quad \forall \mathbf{s}_t \in \mathcal{S}_0 \cap \mathcal{S}_f^\pi\,, \tag{9}$$

where $1_A = 1$ indicates that the value is 1 when the event A occurs and $1_A = 0$ otherwise; $\mathcal{S}_f^\pi = \{\mathbf{s}_t \in \mathcal{S} : \min_{\mathbf{a}_t \in \mathcal{A}} Q_c^\pi(\mathbf{s}_t, \mathbf{a}_t) \le 0\}$ represents the set of persistently safe states, meaning all states from which constraints are never violated under a given policy π, and these states are contained within the feasible set of the policy π. This type of constraint is also referred to as a statewise constraint [17]. The detailed safety criteria are provided in Section III-B.4.

We utilize a sparse reward method to balance the constraints and solve the CMDP (see Algorithm 1, line 5, 16). In the remainder of this section, the CMDP is formulized for the UAV traverse problem, and we start with the definition of the state space, action space, and reward function.

1) State Space: The state space $\mathcal{S}$ encompasses relative observational information from the surrounding environment $\mathcal{S}_\Delta^{\text{env}}$ and absolute information about the goal point $\mathcal{S}^{\text{goal}}$ (see Algorithm 1, line 4). This can be represented as

$$\mathcal{S} = [\mathcal{S}_\Delta^{\text{env}}, \mathcal{S}^{\text{goal}}]\,. \tag{10}$$

In specific, $\mathcal{S}_\Delta^{\text{env}}$ is defined as the difference between the UAV's current state and the state of the gate at time step t:

$$\mathcal{S}_\Delta^{\text{env}} = \mathbf{x}_{q,t} - \mathbf{x}_{g,t}\,, \tag{11}$$

where $\mathbf{x}_{q,t}$ and $\mathbf{x}_{g,t}$ represent the complete state variables of the UAV and the gate, respectively. To learn policies useful for online parameter adaptation or compatible with high-dimensional sensory observations, we randomly initialize the environment and the UAV's state during the safe DRL training process. Therefore, the initial state of UAV $\mathbf{x}_{q,0}$ and gate $\mathbf{x}_{g,0}$ at time $t = 0$ will be randomly reset at each training episode.

The goal point-related features $\mathcal{S}^{\text{goal}}$ is represented as

$$\mathcal{S}^{\text{goal}} = \mathbf{x}_{\text{goal}}\,, \tag{12}$$

where $\mathbf{x}_{\text{goal}}$ is the goal state of the UAV.

Algorithm 1 DCRPO

Input: Initial UAV state $\mathbf{x}_0$, safe DRL policy π with parameters θ_π, NMPC controller, SCNet with parameters θ_{conf}, offline training dataset $\mathcal{D}_{\text{offline}} = \{(\mathbf{x}_{q,t}, \mathbf{u}_{q,t}, \mathbf{w}_{q,t}, \hat{\sigma}_t)\}$, confidence threshold α, prediction horizon N, safety threshold γ, constants C_1, C_2, and weighting coefficients ω_1, ω_2, ω_3.

Output: Optimized policy $\pi^*(\theta_\pi)$ and updated SCNet.

1: Initialize safe DRL policy parameters θ_π (randomly initialized)
2: Initialize SCNet parameters θ_{conf} (pre-trained with offline dataset)
3: **for** each time step t **do**
4: Obtain current observation $\mathcal{S}$ from the environment ▷ Eq. (10)
5: Calculate high-level safe decision $\pi^*(\theta_\pi)$ ensuring safety constraints are satisfied ▷ Eq. (9)
6: Calculate the planned trajectory $\{\boldsymbol{\tau}^*\}_{t=0}^N$ ▷ Eq. (18)
7: Calculate confidence reachable set $\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}}$ ▷ Eq. (22)
8: **if** $\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}} \cap \mathcal{X}_{\text{obs}} \neq \emptyset$ **then**
9: Apply penalty $P_{\text{collision}}$ ▷ Eq. (24)
10: Terminate interaction and log safety violation
11: **else**
12: Calculate SCNet online loss $\mathcal{L}_{\text{online}}$ ▷ Eq. (21)
13: Update SCNet parameters θ_{conf}
14: **end if**
15: Calculate reshaped reward r' ▷ Eq. (25)
16: Update safe DRL policy π parameters θ_π based on r'
17: **end for**

2) Action Space: The key to integrating safe DRL with NMPC lies in utilizing the action set $\mathcal{A}$ from the upper-level safe DRL as the safety decision variables for the lower-level NMPC. Specifically, in learning the trajectory to traverse a fast-moving rotating gate and safely executing this challenging task, it is essential to identify the traversal characteristics required by the UAV, i.e., the safe interaction information. In this work, the desired traversal state $\mathbf{x}_{q,\text{tra}}$ and the desired traversal time t_{tra} are used to characterize the safe interaction information. Therefore, the action output of the safe DRL can be represented by $\mathcal{A} \in \mathbb{R}^7$:

$$\mathcal{A} = [\mathbf{x}_{q,\text{tra}}, t_{\text{tra}}]\,. \tag{13}$$

3) Reward Function: The reward function plays a key role in guiding the UAV agent's learning, ensuring it prioritizes both goal achievement and safety constraints. The objective is to balance the competing objectives of maximizing task completion and minimizing safety interaction violations.

First, we define the baseline or maximum possible reward $r_{\max}$ as

$$r_{\max} = C_1\,, \tag{14}$$

where $r_{\max}$ serves as the maximum possible reward that the agent can achieve if it perfectly accomplishes the task without any deviations or safety violations, with a constant value C_1 (e.g., 100).

Second, the UAV agent receives a negative goal reward r_{goal} associated with the deviation from the goal point. This reward

is expressed as the Euclidean distance between the UAV's position and the goal point during the final n time steps of the trajectory:

$$r_{\text{goal}} = -\sum_{t=N-n}^{N} \| \mathbf{x}_{q,t} - \mathbf{x}_{\text{goal}} \|_2^2 \ , \tag{15}$$

where N represents the total prediction horizon of the trajectory.

Third, when the UAV methods the rectangular gate, a negative collision reward $r_{\text{collision}}$ is given based on the distance between the UAV and the corners of the gate. The closer the UAV is to a collision, the greater the penalty.

$$r_{\text{collision}} = -\left(\sum_{i=1}^{4} 2\varepsilon \cdot d_i + \varepsilon^2\right) , \tag{16}$$

where ε is a safety margin and d_i is the shortest distance between the UAV and the gate border.

In summary, the reward function r in safe DRL can be formulated as

$$r = r_{\max} + \omega_1 \cdot r_{\text{goal}} + \omega_2 \cdot r_{\text{collision}} , \tag{17}$$

where ω_1 and ω_2 are the weighting coefficients.

Once a high-level decision is made, the NMPC is employed to generate precise control inputs that guide the UAV along the optimal trajectory while ensuring safety. The NMPC method solves a constrained optimal problem at each time step, which involves predicting the future states of the UAV over a finite horizon N, based on a sequence of state $\mathbf{x}_{q,t=0,\dots,N}$ and the current control inputs $\mathbf{u}_{q,t}$. Let $\boldsymbol{\tau} = [\mathbf{x}_{q,t=0,\dots,N}, \mathbf{u}_{q,t}]$ denotes the trajectory generated by the NMPC controller, which can be expressed as $\boldsymbol{\tau} = \text{NMPC}(\mathcal{A})$.

4) Objective Function: NMPC is capable of generating an optimal state sequence $\{\mathbf{x}^*_{q,t} \mid \forall t \in [0, N]\}$ that directs the UAV towards $\mathbf{x}_{\text{goal}}$. Additionally, NMPC computes an optimal control sequence $\{\mathbf{u}^*_{q,t} \mid \forall t \in [0, N-1]\}$ over a horizon N that encompasses the planned trajectory $\boldsymbol{\tau}^*$. In the NMPC framework, the location of the gate is treated as an intermediate waypoint, and the primary objective is to minimize the sum of five quadratic components:

$$\begin{aligned}
\min_{\boldsymbol{\tau}} J = \min_{\mathbf{x}_{0:N}, \mathbf{u}_{0:N-1}} & \sum_{t=1}^{N} \Big(\|\mathbf{x}_{q,t} - \mathbf{x}_{\text{goal}}\|^2_{\mathbf{Q}_x} + \|\mathbf{u}_{q,t}\|^2_{\mathbf{Q}_u} \\
& + \|\mathbf{u}_{q,t} - \mathbf{u}_{q,t-1}\|^2_{\mathbf{Q}_{\Delta u}} + \|\mathbf{x}_{q,t} - \mathbf{x}_{q,\text{tra}}\|^2_{\mathbf{Q}_{\text{tra}}} \Big) \\
& + \|\mathbf{x}_N - \mathbf{x}_{\text{goal}}\|^2_{\mathbf{Q}_{x_N}} \\
\text{s.t.} \quad & \mathbf{x}_{q,t+1} = \mathbf{x}_{q,t} + d_t \hat{f}(\mathbf{x}_{q,t}, \mathbf{u}_{q,t}) + \mathbf{w}_{q,t} \\
& \mathbf{u}_{\min} \le \mathbf{u}_{q,t} \le \mathbf{u}_{\max}, \quad \mathbf{x}_0 = \mathbf{x}_0 ,
\end{aligned} \tag{18}$$

where $\mathbf{u}_{\min}$ and $\mathbf{u}_{\max}$ denote the lower and upper bounds of the control input vector, respectively; d_t is the sampling time; $\mathbf{x}_0$ signifies the initial states; $\mathbf{u}_{q,t}$ represents the control input variables at time t and $\mathbf{r}_{\text{tra}}$ denotes the reference traversal state. The matrices $\mathbf{Q}_x \in \mathbb{R}^{13\times13}$, $\mathbf{Q}_u \in \mathbb{R}^{4\times4}$, $\mathbf{Q}_{\Delta u} \in \mathbb{R}^{4\times4}$, and $\mathbf{Q}_{tra} \in \mathbb{R}^{4\times4}$ are the weight matrices. Among these, $\mathbf{Q}_x$, $\mathbf{Q}_u$, and $\mathbf{Q}_{\Delta u}$ are time-invariant. The traversal cost matrix $\mathbf{Q}_{\text{tra}}$ is defined as

$$\mathbf{Q}_{\text{tra}}(t_{\text{tra}}, t) = \mathbf{Q}_{\max} \exp\left(-\beta(td_t - t_{\text{tra}})^2\right) , \tag{19}$$

where $\mathbf{Q}_{\max} \in \mathbb{R}^{4\times4}$ represents the maximum traversal cost matrix, surpassing all other time-invariant cost matrices in significance. $\beta \in \mathbb{R}^+$ denotes the temporal spread of the traversal cost. Assuming the UAV passes through the gate at time step t_{tra} and $td_t = t_{\text{tra}}$, we have $\mathbf{Q}_{\text{tra}}(t_{\text{tra}}, t) \approx \mathbf{Q}_{\max}$, imposing a large penalty to ensure precise gate tracking. After passing the gate ($t > t_{\text{tra}}$), $\mathbf{Q}_{\text{tra}}$ decreases exponentially, reducing its influence on the NMPC optimization. Consequently, the optimization in (18) prioritizes trajectory tracking and directs the UAV toward the goal point.

Finally, we obtain the planned trajectory $\{\boldsymbol{\tau}^*\}_{t=0}^{N}$, which 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 (see Algorithm 1, line 6).

C. Deep Confidence-Enhanced Stochastic Reachability Certificate

1) Confidence Network Based on Self-Supervised Deep Learning (SCNet): To enhance the accuracy of the reachability safety certificate, we propose using SCNet, a Multilayer Perceptron (MLP), to predict the confidence of the reachable set. This method leverages both offline pre-training and online self-supervised deep learning to improve the accuracy and robustness of the predicted confidence values, thereby strengthening the reliability of the reachability safety certificates.

We collect a large dataset of system states, control inputs, noise, and corresponding reachability outcomes during offline training to train the SCNet. The dataset, denoted as $\mathcal{D}_{\text{offline}} = \{(\mathbf{x}_{q,t}, \mathbf{u}_{q,t}, \mathbf{w}_{q,t}, \hat{\sigma}_t)\}$, where $\hat{\sigma}_t$ represents the empirical confidence value for the state $\mathbf{x}_{q,t}$, is used to train the SCNet. The network is parameterized by θ_{conf} and the loss function $\mathcal{L}_{\text{offline}}(\theta_{\text{conf}})$ can be written as

$$\mathcal{L}_{\text{offline}}(\theta_{\text{conf}}) = \frac{1}{N}\sum_{i=1}^{N} (\hat{\sigma}_{t,i} - \sigma_{t,i})^2 , \tag{20}$$

where $\sigma_{t,i}$ represents the predicted confidence values.

In online scenarios, we ensure that SCNet predictions align with the system's safety objectives by conducting online training in coordination with safe DRL. This collaboration enhances training efficiency and overall system safety. To enable SCNet to effectively differentiate between high-risk and low-risk situations while minimizing unnecessary intervention in low-risk cases, we define the loss function for online training based on (16):

$$\mathcal{L}_{\text{online}}(\theta_{\text{conf}}) = \begin{cases} -1, & \text{if } r_{\text{collision}} > \gamma \\ 0, & \text{if } r_{\text{collision}} \le \gamma \end{cases} , \tag{21}$$

where γ is a predefined safety threshold. When the collision risk exceeds this threshold, the loss function penalizes the network by assigning a loss of -1. This drives the network to focus on reducing collision risk. If the risk is below the threshold, the loss is set to 0, indicating no immediate corrective action is required (see Algorithm 1, lines 12-13).

2) Reachability-Based Safe Certificate: We enhance the reachability safety certificate process by integrating the SCNet into the probabilistic zonotope framework. The planned trajectory $\{\boldsymbol{\tau}^*\}_{t=0}^{N}$ can be written as the confidence reachable set $\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}}$ including uncertainty based on (4) and (6):

$$\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}} = \mathbf{X}(\mathbf{c}_{\boldsymbol{\tau}^*}^{\text{conf}}, \mathbf{G}_{\boldsymbol{\tau}^*}^{\text{conf}}, \boldsymbol{\Sigma}_{\boldsymbol{\tau}^*}^{\text{conf}}), \tag{22}$$

where $\mathbf{c}_{\boldsymbol{\tau}^*}^{\text{conf}}$ and $\mathbf{G}_{\boldsymbol{\tau}^*}^{\text{conf}}$ represent the trajectory center and generator matrix generated by NMPC; $\boldsymbol{\Sigma}_{\boldsymbol{\tau}^*}^{\text{conf}}$ represents the covariance matrix representing the stochastic uncertainty updated by the SCNet (see Algorithm 1, line 7).

To prevent collisions from naive policy rollout, we would check whether the confidence reachable set $\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}}$ within the predictive time horizon includes fault-safe operations. The criterion for checking is to ensure the reachable set remains free from collisions with obstacles:

$$\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}} \cap \mathcal{X}_{\text{obs}} = \emptyset, \tag{23}$$

where $\mathcal{X}_{\text{obs}}$ represents the union of zones constrained by obstacles perceived by the UAV as well as the UAV's hovering state (see Algorithm 1, lines 8-10).

3) Reward Shaping: If the above safe conditions are not met, the agent-environment interaction will be directly terminated, and output a predictive safety violation penalty thereby improving the safety of the training process. The predictive safety violation penalty is defined as

$$P_{\text{collision}} = \begin{cases} C_2 & \text{if } \mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}} \cap \mathcal{X}_{\text{obs}} \neq \emptyset \\ 0 & \text{otherwise} \end{cases}, \tag{24}$$

where C_2 is a negative constant penalty value applied when an intersection occurs.

This kind of shielding strategy is realized by re-designing the reward function of safe DRL, which is crafted to assess the quality of $\mathcal{R}_{\boldsymbol{\tau}^*}^{\text{conf}}$ and can be formalized as

$$r' = r + \omega_3 \cdot P_{\text{collision}}, \tag{25}$$

where ω_3 is the coefficient that scales the impact of the $P_{\text{collision}}$ (see Algorithm 1, line 15).

To avoid excessive penalties and achieve a balance between positive safety certification and actual interaction collision penalties, the relationship between $P_{\text{collision}}$ and ω_2 of the $r_{\text{collision}}$ is as

$$\omega_2 = \begin{cases} \epsilon_{\text{collision}} & \text{if } P_{\text{collision}} \neq 0 \\ \omega_{\text{collision}} & \text{otherwise} \end{cases}, \tag{26}$$

where $\omega_{\text{collision}}$ is the standard collision penalty coefficient, and $\epsilon_{\text{collision}}$ is a very small value (e.g., 0.00001).

IV. Experimental Validation

We apply the proposed DCRPO framework to tackle the complex challenge of agile UAV flight in dynamic environments, as illustrated in Fig. 4. Specifically, we focus on enabling the UAV to fly through a fast-moving rectangular gate and reach the subsequent goal point. Successfully flying through this fast-moving gate allows UAVs to operate in noisy environments where free space is constantly changing. This task is challenging, as the UAV must simultaneously plan an accurate trajectory through the center of a moving gate, reach the goal point under varying environmental noise, and maintain precise control to follow the trajectory.

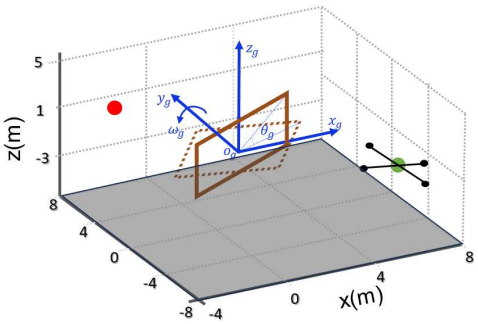


Fig. 4. Illustration of the scenario and the rectangular gate model. The UAV traverses a moving gate (solid brown rectangle) from a random starting point (solid green dot) and reaches the goal point (solid red dot).

A. Dynamics Model

1) Quadrotor Dynamics Model: In this study, the UAV's continuous-time kinematics model can be written as [36]:

$$\dot{\mathbf{x}}_q = \begin{bmatrix} \dot{\mathbf{p}} \\ \dot{\mathbf{q}} \\ \dot{\mathbf{v}} \\ \dot{\boldsymbol{\omega}}_b \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \mathbf{q} \cdot \left[\mathbf{0}, \frac{\boldsymbol{\omega}_b}{2}\right]^T \\ \frac{1}{m}\left(\mathbf{q} \odot (f_r \mathbf{e}_z) - g\mathbf{e}_z\right) \\ \mathbf{J}^{-1}\left(\boldsymbol{\tau}_r - \boldsymbol{\omega}_b \times \mathbf{J}\boldsymbol{\omega}_b\right) \end{bmatrix}, \tag{27}$$

where $\mathbf{p} = [x, y, z]^T$, $\mathbf{q} = \left[q_0, q_x, q_y, q_z\right]$ and $\mathbf{v} = \left[v_x, v_y, v_z\right]^T$ are the position vector, the unit attitude quaternions, and the 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, $\left[\mathbf{0}, \frac{\boldsymbol{\omega}_b}{2}\right]$ is the rotation represented by quaternion. g is the gravity constant and $\mathbf{e}_z = [0, 0, 1]^T$. 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 quadrotor are defined as $\mathbf{x}_q = [\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}_b]^T$ and $\mathbf{u}_q = [f_1, f_2, f_3, f_4]^T$, respectively.

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

$$\dot{\mathbf{p}}_g^w = \mathbf{v}_g^w, \quad \dot{\omega}_g = \ddot{\theta}_g = 0, \tag{28}$$

where $\mathbf{p}_g^w = \left[x_g, y_g, z_g\right]^T$ and $\mathbf{v}_g^w = \left[v_{xg}, v_{yg}, v_{zg}\right]^T$ are the position vector and linear velocity vector of the gate's COM in the world frame, respectively. The rectangular gate frame is depicted in Fig. 4. Assuming the gate is perpendicular to the ground, it rotates by an angle θ_g in the $x_g o_g z_g$ plane, with its initial angle $\theta_{g,0}$ inversely proportional to the gate's width l_g. ω_g is angular rate and its rate of change is zero. Thus, gate states are defined as $\mathbf{x}_g = \left[\mathbf{p}_g^w, \mathbf{v}_g^w, \theta_g\right]^T$.

B. Experimental Settings

1) Environment Configuration: To validate the effectiveness of the proposed methodology for safe interaction in complex dynamic environments, we select a high-risk task where a

TABLE III
EXPERIMENT PARAMETERS

Notations	Specifications	Value
$\mathbf{p}_0$	UAV initial position	$[0,-8,0]^{\mathrm{T}}$ m*
ψ_0	UAV initial yaw angle	$U(-0.1,0.1)$ rad
$\mathbf{p}_{\text{goal}}$	UAV goal position	$[0,8,0]^{\mathrm{T}}$ m*
$\mathbf{v}_g$	Gate time-varying velocity	$\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\sigma}^2)$ m/s
$\boldsymbol{\sigma}$	Gate velocity standard deviation	$\mathrm{diag}(0.1,0.1,0.1)$ m/s
$\theta_{g,0}$	Gate initial angle	$U\left(-\frac{\pi}{2},\frac{\pi}{2}\right)$ rad/s
g	Gravity constant	9.85 m/s^2
m	UAV weight	0.5 kg
l_q	UAV arm length	0.85 m
l_g	Gate width	$U(1.8,2.5)$ m
$\mathbf{J}$	UAV moment of inertia	$[0.003,0.003,0.004]^{\mathrm{T}}$ kg·m^2
$\mathcal{W}$	Gaussian observation noise	$\mathcal{N}(0,\boldsymbol{\Sigma}_v)$**

* $\mathbf{p}_0$ and $\mathbf{p}_{\text{goal}}$ are perturbed by $U(-5,5)$.
** $\boldsymbol{\Sigma}_v = k_{\text{level}} \times 0.01^2 \times \mathbf{I}$, where $k_{\text{level}} \in \{0, 0.01, 0.05\}$ is the noise level coefficient and $\mathbf{I}$ is the identity matrix.

UAV flies from an initial point through a fast-moving rotating gate and reaches a goal point, as shown in Fig. 4. To train and test the DRL-based policies, we create a dynamic and highly randomized scenario in the simulator, resetting the system in random states. Specifically, the UAV's initial state (position, orientation, and heading), target point location, and the gate's initial state (position and rotational angular velocity) are randomly chosen according to a uniform distribution. To simulate the uncertain noise dynamics in the environment, we set the observation noise as Gaussian with a mean of 0 and a covariance matrix that follows a uniform random distribution, which is reset for each task. The environmental physical parameters, such as arm length l_b, are detailed in Table III. Specifically, the UAV's initial positions $\mathbf{p}_0$ and goal positions $\mathbf{p}_{\text{goal}}$ are initialized with uniform random perturbations for each training iteration, whereas the remaining UAV initial states are fixed at zero. The dynamic gate's velocity $\mathbf{v}_g^w$ is drawn from a Gaussian distribution, while the initial angular rate of the gate $\theta_{g,0}$ and the UAV's yaw angle are sampled from a uniform distribution.

2) Implementation Details: We utilize the actor-critic architecture [37] as our DRL agent in solving the UAV interaction problem. The policy and value functions are implemented using neural networks, with the network architecture and key parameters detailed in Table IV.

We employ CasADi [38] with IPOPT [39] as the solver for the numerical optimization problem. The algorithm and associated control procedures are implemented in Python, where PyTorch [13] is utilized for constructing the neural network. For NMPC, we set the time step $dt = 0.1$ s and a prediction horizon of $N = 50$. All code is executed on a desktop computer equipped with an Intel i7-13700KF CPU and an Nvidia RTX 4070 Ti GPU.

C. Baseline Methods

We conduct a comparative evaluation using several baseline methods, ensuring that the inputs and outputs for all baselines are consistent with those of the proposed method.

TABLE IV
HYPERPARAMETERS OF THE SAFE DRL ALGORITHM

Parameters	Value
Safe DRL	
Discount factor γ	0.99
Actor Learning rate	3e-4
Critic Learning rate	1e-3
Total episodes	300
Minibatch size	100
Approximation function	MLP
Hidden layers	4
Neurons per layer	256
Activation	RELU
Optimizer	Adam
Reply buffer size	4e5
SCNet	
Discount factor γ	0.99
Learning rate	1e-4
Minibatch size	100
Approximation function	MLP
Hidden layers	2
Neurons per layer	256
Activation	RELU
Optimizer	Adam

DRL-RA: This method employs reachability analysis (RA) based on forward invariance theory to evaluate policy safety [40]. The forward accuracy set captures all states reachable from bounded initial conditions under various inputs within a finite or infinite time horizon. The system is deemed safe if this set does not intersect with unsafe regions. DRL-RA integrates these principles as a state-of-the-art DRL method for safety assessment.

DRL-RS: This reward shaping (RS) baseline augments the DRL reward function with a fixed-penalty term reflecting the distance to a moving gate [41]. By incorporating prior knowledge as an additional penalty, the agent is guided to maintain safe distances from obstacles. This baseline is used to evaluate the effectiveness of our method compared to traditional reward-based safety integration.

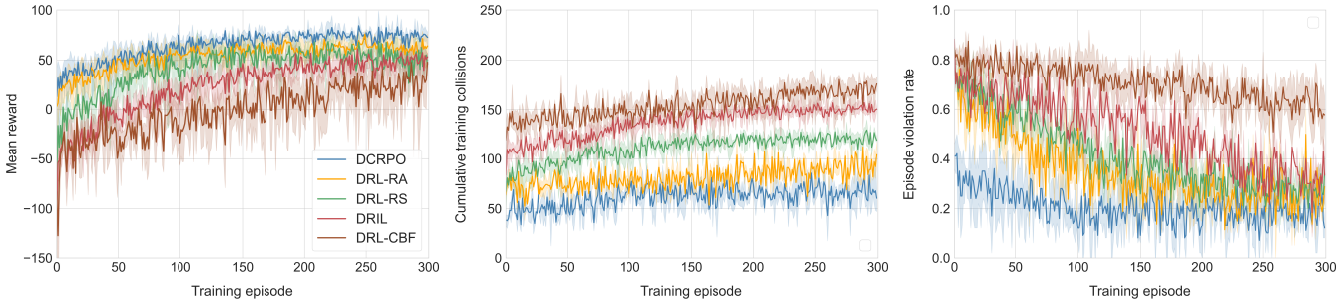
DRIL: The DRIL framework decomposes training into two stages [18]. It first uses DRL to train a neural network for static gate navigation, then applies supervised learning with a binary search algorithm to adapt the network for dynamic environments. DRIL serves as a benchmark to compare the advantages of our DCRPO over imitation learning methods.

DRL-CBF: This method integrates CBF into DRL to define and maintain safety states through energy-based constraints [17]. The CBF constraint,$B(s) \triangleq \dot{h}(s) + \mu h(s) \leq 0$ with $\mu \in (0,1)$, regulates system energy as it methods safety boundaries. This baseline assesses whether DCRPO offers improvements over conventional energy-function-based safety methods.

D. Metrics

To evaluate the UAV flight interaction policy, we use specific metrics during both training and testing phases.

Training Metrics: The training performance is assessed using three metrics: (1) **mean reward** reflects the average performance of agents per episode; (2) **cumulative training collisions** measure exploration safety by counting total collisions since training began; (3) **episode violation rate** reflects



(a) (b) (c)

Fig. 5. Comprehensive training performance of different agents on the interaction task between UAV and fast-moving rectangular gate. (a) Mean reward. (b) Cumulative training collisions. (c) Episode violation rate.

TABLE V

TRAINING ASSESSMENT RESULTS IN THE LAST 100 EPISODES, INCLUDING THE MEAN AND STANDARD DEVIATION (IN BRACKETS)

Methods	Mean reward	Cumulative training collisions	Episode violation rate (%)
DCRPO	**81.03 (5.86)**	**68.67 (4.92)**	**0.14 (0.11)**
DRL-RA	68.10 (6.07)	92.16 (4.66)	0.22 (0.11)
DRL-RS	60.98 (8.46)	121.90 (4.95)	0.29 (0.16)
DRIL	51.79 (10.91)	157.83 (5.32)	0.35 (0.17)
DRL-CBF	46.84 (12.53)	178.10 (5.77)	0.60 (0.16)

the frequency of constraint violations per episode during policy learning. Notably, we do not terminate the episode upon encountering a violation. Among these, cumulative training collisions are the primary safety indicator in critical DRL tasks.

Test Metrics: Post-training, seven metrics are used to evaluate the DRL agent: (1) **safe rate** is the proportion of episodes without collisions; (2) **goal rate** is the percentage of successful goal completions; (3) **mean/max speed** reflect the ratio of average to maximum speed, indicating agility; (4) **reaching time** indicates the time taken for the UAV to reach the goal; (5) **computing time** reflects the algorithm's computational load; (6) **traversal error** represents the mean and standard deviation of the distance error between the UAV and gate centers at traversal; and (7) **goal error** indicatesthe mean and standard deviation of distance errors to the goal.

V. RESULT AND ANALYSIS

We evaluate the proposed method from two perspectives: training performance evaluation and test task evaluation.

A. Training Evaluation

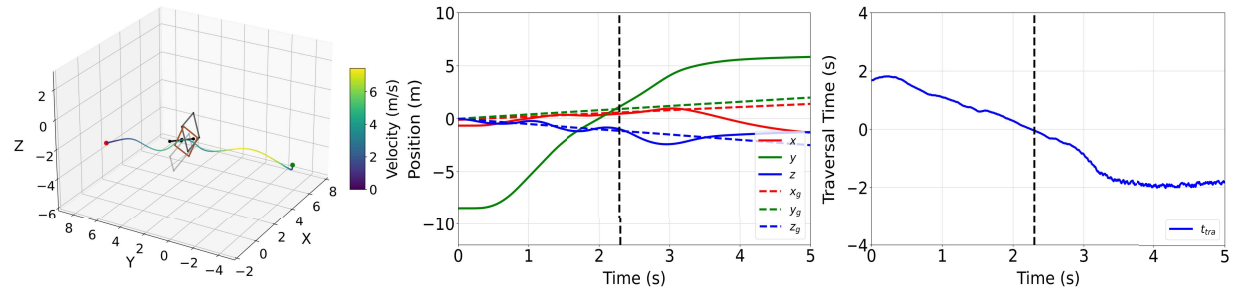
We first evaluate the proposed algorithm during policy search, focusing on the agent's optimal training outcomes and the guarantee of safe exploration. Fig. 5 presents the overall training results and Table V shows the quantitative results of the training evaluation over the last 100 episodes, including mean and standard deviation. To ensure a fair comparison, each task for each algorithm is repeated with seven identical random seed sequences.

From the training rewards shown in Fig. 5(a), DCRPO achieves a significantly higher mean reward of 81.03 (with a standard deviation of 5.86), surpassing the closest competitor, DRL-RA, which only reaches 68.10 (6.07). While DRL-RA offers competitive performance in some aspects, it falls short in terms of safety, as reflected by its high collision count and violation rate. The other methods, DRL-RS, DRIL, and DRL-CBF, show lower mean rewards of 60.98 (8.46), 51.79 (10.91), and 46.84 (12.53), respectively. This indicates that DCRPO can rapidly improve its policy, achieving the highest asymptotic performance with minimal performance variation. As depicted by the total collision count in 5(b), the reachability-based DCRPO, due to its ability to assess potential risks in the forward-predicted trajectory, experiences significantly fewer collisions during interaction with the environment compared to other algorithms. The proposed DCRPO, enhanced by deep confidence network-based reachability guarantees, further improves the accuracy of reachability verification, resulting in the lowest total number of collisions. Specifically, DCRPO reduces total collisions by 25.49%, 43.67%, 56.49%, and 61.44% compared to the DRL-RA, DRL-RS, DRIL, and DRL-CBF methods, respectively. The superiority of DCRPO in ensuring training safety is further illustrated by the episode violation rate shown in Fig. 5(c), which progressively methods zero as the intermediate policy search progresses. The lower violation rate indicates that DCRPO is highly effective in adhering to safety constraints during training, as the DCRPO is designed to minimize the likelihood of the UAV breaching safety boundaries. This performance advantage is critical in safety-sensitive applications, as it ensures that the UAV operates within predefined safety limits at all times.

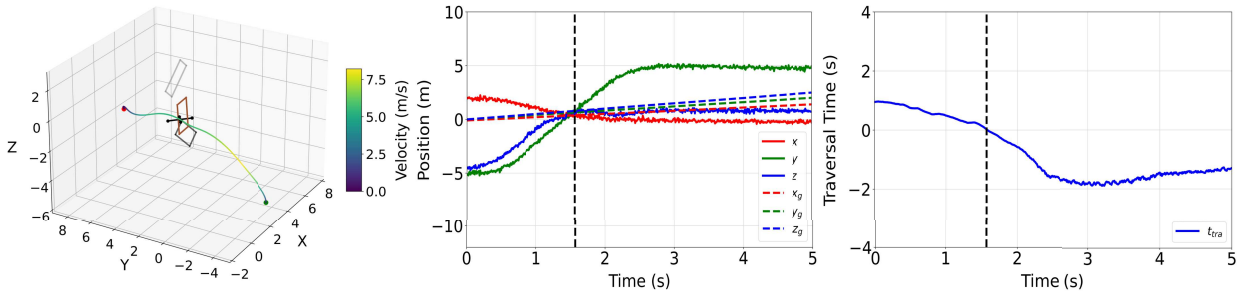
In a word, the DCRPO's ability to efficiently search for an optimal policy with high average rewards while minimizing collision behavior highlights its effectiveness in achieving both safety and performance objectives.

B. Test Evaluation

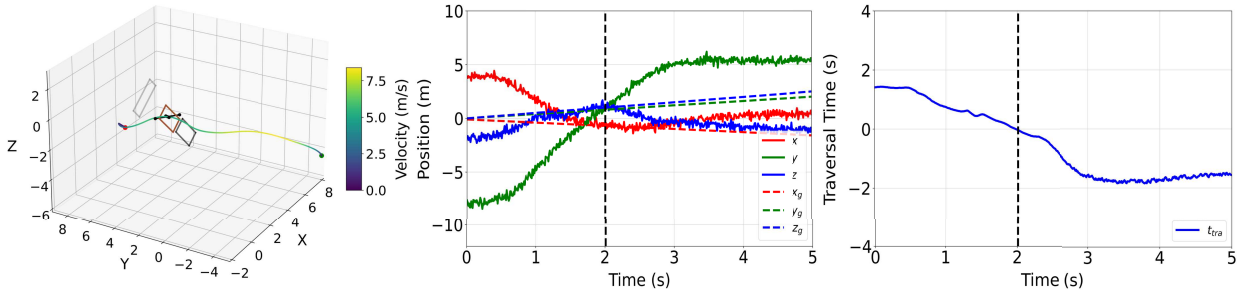
In this section, the post-trained safe DRL agent of each algorithm is tested using a series of simulated scenarios with randomly generated features. To evaluate the task completion capability of the proposed method under well-trained conditions, we conduct batch tests across three different levels



(a) noise level 0; $\boldsymbol{\mu}(0.3, 0.4, -0.5)$



(b) noise level 1; $\boldsymbol{\mu}(0.3, 0.4, 0.5)$



(c) noise level 2; $\boldsymbol{\mu}(-0.3, 0.4, 0.5)$

Fig. 6. Three test scenarios with different noise levels and gate velocities. **Left:** The 3D planned trajectory for flying through a dynamic rectangular gate, with a color bar indicating the quadrotor's velocity (m/s). The initial state of the fast-moving gate is represented in black, its state during traversal in orange, and its final state in gray. **Middle:** Trajectories of the quadrotor (solid line) and the fast-moving gate (dashed line). **Right:** The predicted traversal time variable output by the DRL policy. The vertical dashed black lines indicate the moments when the quadrotor is traversing the gate.

of environmental noise. Noise level (0) indicates the absence of noise, and other noise levels (1) and (2) are modeled as Gaussian with $\mathcal{N}(0, 0.01\mathbf{I})$ and $\mathcal{N}(0, 0.05\mathbf{I})$ distributions, where $\mathbf{I}$ is the identity matrix. Each environment is tested with 1000 random trials.

Three random test samples with different levels of environmental noise and varying average gate speeds are shown in Fig. 6. Our method allows the UAV to navigate through the dynamic rectangular gate as closely as possible to the goal while adapting to varying noise levels, gate speeds, and sizes. As illustrated in the 3D trajectory on the left side of Fig. 6, the UAV successfully enters the predefined safe traversal area of the gate while maintaining an optimal posture. Furthermore, considering the receding horizon method of NMPC, the time from the initial point to the optimal position corresponds to the traversal time output by the DRL policy. The execution trajectories comparison between the UAV and the moving gate, shown in the middle of Fig. 6, demonstrates that as the DRL-predicted traversal time methods zero (right side of Fig. 6, indicated by the vertical black dashed line), the UAV gradually decelerates and aligns itself with the centroid of the gate to ensure a safe passage.

Additionally, each DRL-based algorithm is comprehensively evaluated from several perspectives: safe rate, goal achievement rate, average-to-maximum speed ratio, reaching time, and computing time. The results of 500 random test samples for each noise level are summarized in Table VI. The findings indicate that DCRPO, DRL-RA, and DRL-RS maintained safety across all trials in the no-noise scenario (noise level (0)), with other methods also demonstrating high safety. As the level of environmental noise increases, the DCRPO shows a significant advantage, particularly in terms of safety and goal achievement rate. The DCRPO maintains an enhanced safety rate across all noise levels, averaging 99.87%, outperforming other methods, especially DRL-CBF, which only achieves a safety rate of 94.07%. However, there were considerable differences in their ability to safely achieve the target. The DCRPO improves the goal rate by 11.03%, 12.40%, 14.00%, and 33.43% compared to DRL-RA, DRL-RS, DRIL, and DRL-CBF, respectively. This reflects DCRPO's ability to

TABLE VI

COMPARISON OF TEST STATISTICAL RESULTS IN THREE SCENARIOS. REACHING TIME AND COMPUTING TIME ARE PRESENTED WITH THE MEAN VALUE AND STANDARD DEVIATION (IN BRACKETS)

Methods	Metric	Noise Level (0)	Noise Level (1)	Noise Level (2)	Summary
DCRPO	Safe rate (%)	**100.00**	**100.00**	**99.60**	**99.87**
	Goal rate (%)	**98.80**	**96.20**	**94.80**	**96.60**
	Mean/max speed (m/s)	3.32/7.80	2.62/5.98	3.18/7.66	3.04/7.15
	Reaching time (ms)	5.21 (2.52)	5.39 (2.02)	5.44 (3.23)	**5.35 (2.59)**
	Computing time (ms)	18.97 (1.63)	19.48 (2.31)	20.16 (1.91)	19.54 (1.95)
DRL-RA	Safe rate (%)	**100.00**	99.20	96.40	98.53
	Goal rate (%)	92.80	84.60	83.60	87.00
	Mean/max speed (m/s)	3.43/8.19	3.32/8.14	3.28/9.92	**3.34/8.75**
	Reaching time (ms)	5.27 (2.24)	5.35 (2.92)	5.52 (3.57)	5.38 (2.91)
	Computing time (ms)	18.94 (4.58)	18.99 (2.59)	20.95 (4.38)	19.63 (3.85)
DRL-RS	Safe rate (%)	**100**	99.80	97.00	98.93
	Goal rate (%)	92.60	83.80	81.40	85.94
	Mean/max speed (m/s)	2.76/7.47	2.71/6.69	3.02/7.44	2.83/7.20
	Reaching time (ms)	5.30 (3.25)	5.34 (2.01)	5.61 (2.78)	5.42 (2.68)
	Computing time (ms)	16.46 (3.74)	14.28 (1.85)	15.01 (4.14)	**15.25 (3.24)**
DRIL	Safe rate (%)	99.60	95.80	95.40	96.93
	Goal rate (%)	86.60	85.40	82.20	84.73
	Mean/max speed (m/s)	3.55/7.98	3.61/7.07	2.82/7.26	3.32/7.44
	Reaching time (ms)	5.44 (2.67)	5.73 (2.25)	5.89 (3.91)	5.69 (2.94)
	Computing time (ms)	16.96 (4.09)	17.30 (3.76)	17.98 (3.95)	17.41 (3.93)
DRL-CBF	Safe rate (%)	96.00	93.40	92.80	94.07
	Goal rate (%)	77.80	74.60	64.80	72.40
	Mean/max speed (m/s)	2.83/7.33	3.16/7.50	3.38/7.81	3.12/7.55
	Reaching time (ms)	5.71 (3.68)	5.94 (3.09)	6.32 (3.95)	5.99 (3.57)
	Computing time (ms)	18.28 (2.19)	17.30 (1.72)	20.77 (3.32)	18.78 (2.41)

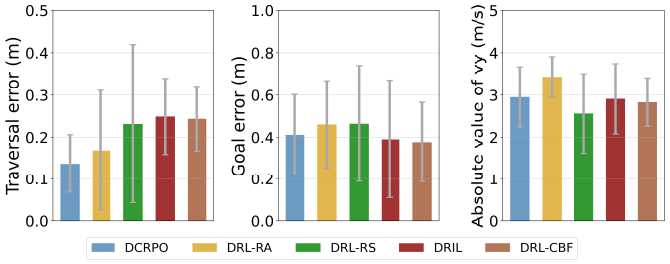


Fig. 7. Statistical results of different strategies in the test scene.

TABLE VII

MODULE-LEVEL AVERAGE COMPUTING TIME (MS)

Method	DRL	NMPC	SCNet
DCRPO	7.14 (0.98)	9.83 (1.23)	2.51 (0.47)
DRL-RA	8.35 (1.07)	10.59 (1.46)	–
DRL-RS	7.92 (1.13)	7.33 (1.05)	–
DRIL	8.12 (1.12)	8.94 (1.28)	–
DRL-CBF	8.77 (1.22)	9.51 (1.30)	–

effectively adjust its policy and achieve mission goals even under varying noise conditions. The high goal rate of DCRPO underscores its effective balance between safety and mission completion.

In terms of computing time, DCRPO records an average of 19.54 ms, slightly higher than that of DRL-RS (15.25 ms). While DCRPO's reaching time and computing time are competitive, it manages a balanced trade-off between computational efficiency and safety, outperforming other baseline methods in overall performance. Notably, the conservative nature of DRL-CBF, as indicated by its lower goal rate and longer reaching time, highlights the limitations of energy-based methods, which lead to a smaller feasible set and reduced overall performance.

The statistical results for the different strategies evaluated in all test sampling scenarios are summarized in Fig. 7. DCRPO maintains high flight efficiency (average speed v_y) while achieving the lowest traversal distance error and goal distance error compared to other baseline methods, indicating the highest task completion rate for interactive missions. This is due to the deep confidence-enhanced reachability analysis in DCRPO, which allows for more accurate risk prediction and mitigation during trajectory planning. Overall, DCRPO performs on par with DRL-RA in simple environments and surpasses it in more challenging scenarios, making DCRPO a safe and effective solution for real-time trajectory planning of UAVs in complex and dynamic environments.

To evaluate the real-time feasibility of DCRPO for onboard UAV deployment, Table VII presents the average inference time for each major module, including safe DRL, NMPC, and SCNet, under noise level (1). The total average inference time

of DCRPO is 19.48 ms per step. Among the modules, NMPC requires the most computation time at 9.83 ms (standard deviation 1.23), followed by the DRL network at 7.14 ms (0.98), and SCNet at 2.51 ms (0.47). Compared with baseline methods, DCRPO has a slightly higher computational cost due to the SCNet module. However, each submodule remains under 10 ms on average, and the total time is capable of satisfying typical UAV real-time control [42]. Importantly, the SCNet adds only about 13% to the total computation time, while significantly improving system safety and robustness. These results confirm that DCRPO achieves a practical balance between computational efficiency and enhanced safety.

VI. Conclusion and Future Work

In this paper, we present a DCRPO framework to address the challenge of optimizing UAV trajectory planning and control in complex dynamic environments. This framework integrates joint planning and control to enhance UAV safety and performance in real-time applications. By combining safe DRL with NMPC, DCRPO provides a secure and efficient solution for UAV joint planning and control, enabling seamless interaction with complex dynamic environments. Furthermore, by incorporating deep confidence-enhanced reachability guarantees, DCRPO minimizes potential collision risks in advance, achieving a balance between safety and task completion performance. This method offers a solution to ensure safe operation within predefined constraints while adapting to dynamic conditions, particularly in safety-critical scenarios involving rapidly moving rotating gates. Experimental results show that DCRPO significantly outperforms other state-of-the-art baseline methods in terms of rewards, safety metrics, and goal achievement rates. In test scenarios with varying environmental noise levels, DCRPO consistently maintains high safety and goal achievement rates, demonstrating its robustness and adaptability in handling complex and dynamic environments.

While DCRPO achieves promising results, it has some limitations. Its reachability analysis increases computation time, and its effectiveness has only been tested in simulations. Future research should conduct real-world experiments to evaluate DCRPO's performance under different conditions and hardware constraints.

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," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–6.

[2] B. Liu, W. Ni, R. P. Liu, Y. J. Guo, and H. Zhu, "Optimal routing of unmanned aerial vehicle for joint goods delivery and in-situ sensing," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3594–3599, Mar. 2023.

[3] K. Kalenberg et al., "Stargate: Multimodal sensor fusion for autonomous navigation on miniaturized UAVs," *IEEE Internet Things J.*, vol. 11, no. 12, pp. 21372–21390, Jun. 2024.

[4] M. O'Connell et al., "Neural-fly enables rapid learning for agile flight in strong winds," *Sci. Robot.*, vol. 7, no. 66, p. 6597, May 2022.

[5] M. Maboudi, M. Homaei, S. Song, S. Malihi, M. Saadatseresht, and M. Gerke, "A review on viewpoints and path planning for UAV-based 3-D reconstruction," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 5026–5048, 2023.

[6] H. Song, D. Luan, W. Ding, M. Y. Wang, and Q. Chen, "Learning to predict vehicle trajectories with model-based planning," in *Proc. Conf. Robot Learn.*, 2022, pp. 1035–1045.

[7] M. Ortlieb and F.-M. Adolf, "Rule-based path planning for unmanned aerial vehicles in non-segregated air space over congested areas," in *Proc. AIAA/IEEE 39th Digit. Avionics Syst. Conf. (DASC)*, Oct. 2020, pp. 1–9.

[8] S. Dixit et al., "Trajectory planning for autonomous high-speed overtaking in structured environments using robust MPC," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2310–2323, Jun. 2020.

[9] A. N. Venkat, I. A. Hiskens, J. B. Rawlings, and S. J. Wright, "Distributed MPC strategies with application to power system automatic generation control," *IEEE Trans. Control Syst. Technol.*, vol. 16, no. 6, pp. 1192–1206, Nov. 2008.

[10] V. M. Zavala and L. T. Biegler, "The advanced-step NMPC controller: Optimality, stability and robustness," *Automatica*, vol. 45, no. 1, pp. 86–93, Jan. 2009.

[11] A. Romero, S. Sun, P. Foehn, and D. Scaramuzza, "Model predictive contouring control for time-optimal quadrotor flight," *IEEE Trans. Robot.*, vol. 38, no. 6, pp. 3340–3356, Dec. 2022.

[12] P. Foehn, A. Romero, and D. Scaramuzza, "Time-optimal planning for quadrotor waypoint flight," *Sci. Robot.*, vol. 6, no. 56, p. 1221, Jul. 2021.

[13] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Sci. Robot.*, vol. 8, no. 82, p. 1462, Sep. 2023.

[14] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, Aug. 2023.

[15] D. Hu, L. Mo, J. Wu, and C. Huang, "'Feariosity'-guided reinforcement learning for safe and efficient autonomous end-to-end navigation," *IEEE Robot. Autom. Lett.*, vol. 10, no. 8, pp. 7723–7730, 2025.

[16] L. Wen, J. Duan, S. E. Li, S. Xu, and H. Peng, "Safe reinforcement learning for autonomous vehicles through parallel constrained policy optimization," in *Proc. IEEE 23rd Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2020, pp. 1–7.

[17] H. Ma, C. Liu, S. E. Li, S. Zheng, and J. Chen, "Joint synthesis of safety certificate and safe control policy using constrained reinforcement learning," in *Proc. Learn. Dyn. Control Conf.*, 2021, pp. 97–109.

[18] 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," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2023, pp. 1680–1686.

[19] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," *J. Mach. Learn. Res.*, vol. 16, no. 1, pp. 1437–1480, 2015.

[20] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 22–31.

[21] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone, "Risk-constrained reinforcement learning with percentile risk criteria," *J. Mach. Learn. Res.*, vol. 18, no. 167, pp. 1–51, 2018.

[22] Y. Zhang, W. Wen, and P. Yan, "Safe-assured learning-based deep SE(3) motion joint planning and control for UAV interactions with dynamic environments," in *Proc. IEEE 27th Int. Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2024, pp. 4222–4229.

[23] D. Hu, C. Huang, J. Wu, and X. Yuan, "Toward multi-task generalization in autonomous navigation: A human-in-the-loop adversarial reinforcement learning with diffusion policy," *IEEE Trans. Intell. Transp. Syst.*, early access, Jul. 31, 2025, doi: 10.1109/TITS.2025.3591239.

[24] Y. Jason, A. Shen, O. Bastani, and J. Dinesh, "Conservative and adaptive penalty for model-based safe reinforcement learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 5, pp. 5404–5412.

[25] D. Hu, C. Huang, J. Zhao, Y. Zhao, and J. Wu, "Autonomous driving economic car-following motion strategy based on adaptive rollout model-based policy optimization," *IEEE Trans. Transport. Electrific.*, vol. 11, no. 5, pp. 12416–12427, Oct. 2025.

[26] Z. Liu et al., "Constrained variational policy optimization for safe reinforcement learning," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 13644–13668.

[27] L. Brunke et al., "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annu. Rev. Control, Robot., Auto. Syst.*, vol. 5, no. 1, pp. 411–444, May 2022.

[28] D. Sun, A. Jamshidnejad, and B. De Schutter, "A novel framework combining MPC and deep reinforcement learning with application to freeway traffic control," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 7, pp. 6756–6769, Jul. 2024.

[29] Y. Song and D. Scaramuzza, "Policy search for model predictive control with application to agile drone flight," *IEEE Trans. Robot.*, vol. 38, no. 4, pp. 2114–2130, Aug. 2022.
[30] A. Romero, Y. Song, and D. Scaramuzza, "Actor-critic model predictive control," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2024, pp. 14777–14784.
[31] A. Halder, "Smallest ellipsoid containing *p*-sum of ellipsoids with application to reachability analysis," *IEEE Trans. Autom. Control*, vol. 66, no. 6, pp. 2512–2525, Jun. 2021.
[32] L. Guibas, A. Nguyen, and L. Zhang, "Zonotopes as bounding volumes," in *Proc. SODA*, 2003, pp. 803–812.
[33] A. Rauh, S. Wirtensohn, P. Hoher, J. Reuter, and L. Jaulin, "Reliability assessment of an unscented Kalman filter by using ellipsoidal enclosure techniques," *Mathematics*, vol. 10, no. 16, p. 3011, Aug. 2022.
[34] D. Hu and Y. Zhang, "Deep reinforcement learning based on driver experience embedding for energy management strategies in hybrid electric vehicles," *Energy Technol.*, vol. 10, no. 6, Jun. 2022, Art. no. 2200123.
[35] H. Ko, S. Pack, and V. C. M. Leung, "An optimal battery charging algorithm in electric vehicle-assisted battery swapping environments," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 5, pp. 3985–3994, May 2022.
[36] A. Saviolo, G. Li, and G. Loianno, "Physics-inspired temporal learning of quadrotor dynamics for accurate model predictive trajectory tracking," *IEEE Robot. Autom. Lett.*, vol. 7, no. 4, pp. 10256–10263, Oct. 2022.
[37] I. Grondman, L. Busoniu, G. A. D. Lopes, and R. Babuska, "A survey of actor-critic reinforcement learning: Standard and natural policy gradients," *IEEE Trans. Syst., Man, Cybern., C (Appl. Rev.)*, vol. 42, no. 6, pp. 1291–1307, Nov. 2012.
[38] M. Fevre, P. M. Wensing, and J. P. Schmiedeler, "Rapid bipedal gait optimization in CasADi," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 3672–3678.
[39] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 528–535.
[40] D. Yu et al., "Safe model-based reinforcement learning with an uncertainty-aware reachability certificate," *IEEE Trans. Autom. Sci. Eng.*, vol. 21, no. 3, pp. 4129–4142, Jul. 2024.
[41] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Proc. ICML*, May 1999, pp. 278–287.
[42] Y. Zhang, Y. Hu, Y. Song, D. Zou, and W. Lin, "Learning vision-based agile flight via differentiable physics," *Nature Mach. Intell.*, vol. 7, no. 6, pp. 954–966, Jun. 2025.

Yuanyuan Zhang (Graduate Student Member, IEEE) received the bachelor's degree in traffic and transportation engineering from Shandong University, Jinan, China, in 2020, and the master's degree in power engineering from Tianjin University, Tianjin, China, in 2023. She is currently pursuing the Ph.D. degree with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University, Hong Kong. Her research interests include trustworthy navigation, adaptive control, deep reinforcement learning, and robotics.

Yingying Wang received the B.E. degree in electronic engineering and the M.S. degree in signal processing from Northeastern University, Shenyang, Liaoning, China, in 2016 and 2019, respectively, and the Ph.D. degree from the Department of Electronic Engineering, The Chinese University of Hong Kong (CUHK), Hong Kong, SAR, China, in 2023. She was a Post-Doctoral Researcher with Hong Kong Automotive Platforms and Application Systems (APAS) Research and Development Center from August 2023 to July 2024. She is currently a Post-Doctoral Fellow with the Intelligent Positioning and Navigation Laboratory, The Hong Kong Polytechnic University, Hong Kong. Her research interests include pedestrian localization and non-intrusive intelligent sensing.

Penggao Yan received the bachelor's degree in communication engineering and the master's degree in pattern recognition and intelligent systems from Wuhan University, China, in 2018 and 2021, respectively. He is currently pursuing the Ph.D. degree with the Department of Aeronautical and Aviation Engineering, Faculty of Engineering, The Hong Kong Polytechnic University. His research interests include high-efficiency microwave power amplifiers, microwave dc/dc converters, radar systems, and wireless power transmission.

Weisong Wen (Member, IEEE) received the B.Eng. degree in mechanical engineering from Beijing Information Science and Technology University (BISTU), Beijing, China, in 2015, the M.Eng. degree in mechanical engineering from China Agricultural University in 2017, and the Ph.D. degree in mechanical engineering from The Hong Kong Polytechnic University. He was a Visiting Student Researcher at the University of California at Berkeley (UCB) in 2018. He is currently an Assistant Professor with the Department of Aeronautical and Aviation Engineering, The Hong Kong Polytechnic University. His research interests include multi-sensor integrated localization for autonomous vehicles, SLAM, and GNSS positioning in urban canyons.