Lightweight Safe Reinforcement Learning for End-to-End UAV Navigation

Shenghui Zhang, YuXuan Gao, Songwei Zhao, Jifeng Hu, Zijing Zhang, and Hechang Chen

***Abstract*— With the rapid development of autonomous aerial systems, Unmanned Aerial Vehicles (UAVs) are increasingly deployed in applications such as inspection, environmental monitoring, and rescue, creating growing demand for reliable autonomous navigation. However, autonomous UAV navigation in dense environments remains challenging under sparse perception and dynamic constraints. Most reinforcement learning (RL) methods lack explicit safety mechanisms, leading to unsafe exploration, unstable training, and risky behaviors, especially during high-speed flight. Even in safe RL approaches, safety is often enforced by projecting policy outputs onto a safe action set, which may introduce instability. Meanwhile, many learning-based methods rely on dense inputs or large networks, increasing computational burden and limiting lightweight onboard deployment. Facing the above challenges, we propose a safety-constrained perception-control integrated framework for UAV navigation. A lightweight network encodes sparse observations into collision-risk-aware features using asymmetric and depthwise separable convolutions. We formulate the task as a constrained Markov decision process within a hierarchical control architecture and solve it using a Lagrangian-based safe PPO algorithm. Curriculum learning further improves training stability. Experiments with varying obstacle densities and flight speeds demonstrate higher success rates, improved safety, and better efficiency than existing reinforcement learning baselines.**

I. INTRODUCTION

With the rapid development of avionics, sensing technologies, and intelligent control, Unmanned Aerial Vehicles (UAVs) have been widely deployed in applications, such as aerial inspection, logistics delivery, and emergency response [1], [2]. In recent years, the rapid growth of the low-altitude economy has further expanded UAV application scenarios, which also places higher requirements on autonomous flight and safe navigation capabilities in complex environments.

Recent studies on UAV autonomous navigation can be broadly categorized into model-based planning methods and learning-based approaches. Classical planning algorithms, such as A*, RRT, and their variants, generate collision-free trajectories using known or reconstructed environment maps and have demonstrated strong performance in structured environments [1], [3], [4]. However, in complex and unstructured scenarios with dense obstacles and incomplete perception, these methods often suffer from limited robustness and high computational cost due to frequent replanning.

Reactive control and learning-based approaches have been introduced to address these limitations.In particular, reinforcement learning (RL) enables UAV to learn navigation policies directly through environment interaction and has shown promising results using algorithms such as DQN [2], DDPG [5], TD3 [6], SAC [7], and PPO [8]. Meanwhile, LiDAR-based perception has been widely adopted due to its robustness to illumination changes and accurate geometric sensing [3], [9]. Several studies leverage raw range measurements or transformed representations (e.g., polar maps or depth images) as inputs to policy networks, enabling end-to-end obstacle avoidance [7], [8].

Despite these advances, several challenges remain unsolved, especially under sparse perception and high-speed navigation conditions. First, existing approaches rely on dense sensory inputs or high-capacity neural networks, which limits their deployment on resource-constrained onboard platforms and increases training complexity. Second, sparse LiDAR observations often fail to explicitly highlight high-risk regions in cluttered environments, making policy training unstable and reducing generalization ability. Finally, most RL frameworks focus on maximizing expected rewards without explicitly modeling safety constraints, which may will lead to risky behaviors during training and deployment. Therefore, how to design a lightweight perception representation that can effectively utilize sparse LiDAR observations while ensuring safe and robust navigation remains an open problem for UAV autonomous flight in complex environments.

To address the above issues, We proposes a safety-constrained perception-control integrated framework for UAV autonomous navigation. The main contributions of our work are summarized as follows:

1) We proposes a safety-constrained UAV navigation method to address instability and safety issues under sparse perception in environments with varying obstacle densities.
2) The approach integrates a lightweight perception network, a unified state representation, and a hierarchical control framework, combined with a Lagrangian-based safe PPO algorithm and a curriculum learning strategy to enable efficient and safe decision-making.
3) Experiments in environments with varying obstacle densities and flight speeds are conducted to demonstrate the improved success rate, enhanced safety, and higher efficiency of the proposed method compared with existing reinforcement learning baselines.

II. RELATED WORK

Deep Reinforcement Learning (DRL) has been widely applied to autonomous UAV navigation and obstacle avoidance due to its ability to learn reactive control policies in complex environments. PPO-based methods exhibit stable training in continuous control tasks and have been used for UAV maneuvering and path planning [10]. Hybrid frameworks combining model-based control and RL further improve

adaptability in dynamic obstacle scenarios [11], while end-to-end RL enables policy learning in multi-UAV or complex urban environments without explicit global mapping [12]. However, most existing approaches formulate the problem as a standard MDP and handle safety objectives implicitly via reward shaping. In dense obstacle environments, this often leads to unstable training and safety violations. Moreover, ensuring onboard real-time deployment and computational efficiency under high-dimensional perceptual inputs remains underexplored.

To explicitly incorporate safety constraints, UAV navigation is increasingly formulated as a CMDP [13]. Lagrangian-relaxed PPO introduces dual variables to balance task reward and constraint cost dynamically [14], achieving more stable constraint satisfaction than conventional penalty-based methods. Recent work improves stability via enhanced dual update schemes and safety value function estimation [15], and further integrates control-theoretic tools such as control barrier functions for stronger safety guarantees [16]. Nevertheless, in real UAV platforms, dual update sensitivity, approximation errors in safety value functions, and maintaining a stable trade-off between performance and safety in complex obstacle environments remain key challenges for Safe RL deployment.

LiDAR provides direct geometric information and is crucial for UAV navigation and obstacle avoidance. Traditional methods often rely on real-time lidar odometry and mapping frameworks, e.g., LOAM [17] and Voxblox [18]. With the rise of learning-based approaches, LiDAR observations are also used for end-to-end policy learning, though most studies assume high-line 3D LiDAR or high-resolution depth input. In contrast, lightweight UAV typically use sparse multi-line LiDAR, whose limited vertical resolution poses challenges for modeling obstacle continuity and identifying navigable gaps. Moreover, raw distance measurements do not explicitly encode collision risk monotonicity, potentially affecting RL training stability. Efficient perception modeling for sparse LiDAR thus remains an open research problem.

III. METHOD

We formulate the UAV obstacle avoidance navigation problem as a CMDP. and propose a unified framework that integrates lightweight sparse LiDAR perception, safe reinforcement learning, and curriculum learning. The proposed method is built upon PPO, enabling end-to-end collision avoidance policy learning under explicit safety constraints, while leveraging efficient perception representations and staged training to improve stability and generalization in complex environments.

A. Convolutional Feature Extraction from LiDAR Data

We propose a lightweight convolutional encoder to efficiently extract geometric features related to obstacle avoidance from LiDAR observations.The encoder learns representations from the input observation X and produces compact feature embeddings for the policy network.The overall network is illustrated in Fig. 2.

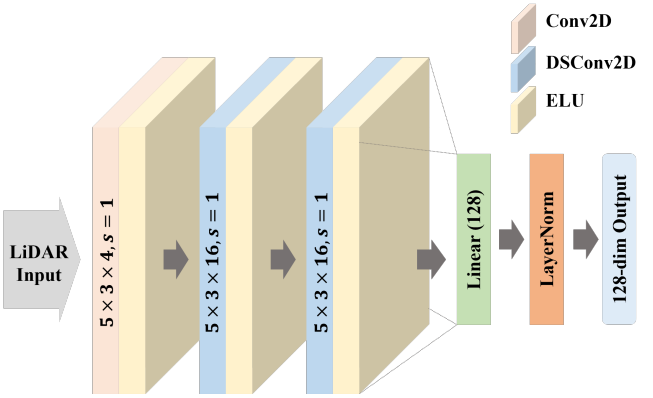

Fig. 2: Architecture of the LiDAR Feature Extraction Network,where Conv and DSConv denote standard and depthwise-separable convolutions, respectively.

The last two layers of the encoder adopt depthwise separable convolutions, decomposing standard convolutions into channel-wise spatial filtering and pointwise feature fusion operations [19]. This design substantially reduces parameter count and computational complexity while preserving the capacity to model local geometric structures, making it well suited for resource-constrained onboard platforms. For sparse LiDAR tensors with limited spatial resolution, the architecture effectively captures horizontally continuous obstacle boundaries as well as height variations reflected by vertically distributed beams. Such a structure supports both large-scale simulation training and real-time deployment in practical UAV systems.

The convolutional layers employ asymmetric kernels to enhance the modeling of horizontal angular resolution while explicitly accounting for the coupling between a limited number of vertical beams. Progressive downsampling is applied across layers to enlarge the receptive field and suppress redundant information. Finally, the convolutional features are mapped into a 128-dimensional embedding vector, followed by Layer Normalization to mitigate distribution shifts caused by variations in LiDAR return statistics across different environments.

B. State Space Definition

In our study, the observation space is designed to provide the agent with comprehensive information about its own motion state, the relative target pose, and the surrounding obstacle distribution, thereby supporting robust autonomous navigation in complex environments. The complete observation at time t can be expressed as

$$o_t = (s_t, X_t) \tag{1}$$

where s_t denotes the proprioceptive state vector, and X_t represents the environment perception information obtained and processed from external LiDAR sensors.

1) Proprioceptive State: To improve the policy's robustness to global heading variations and avoid excessive reliance on the world coordinate system during learning, key motion

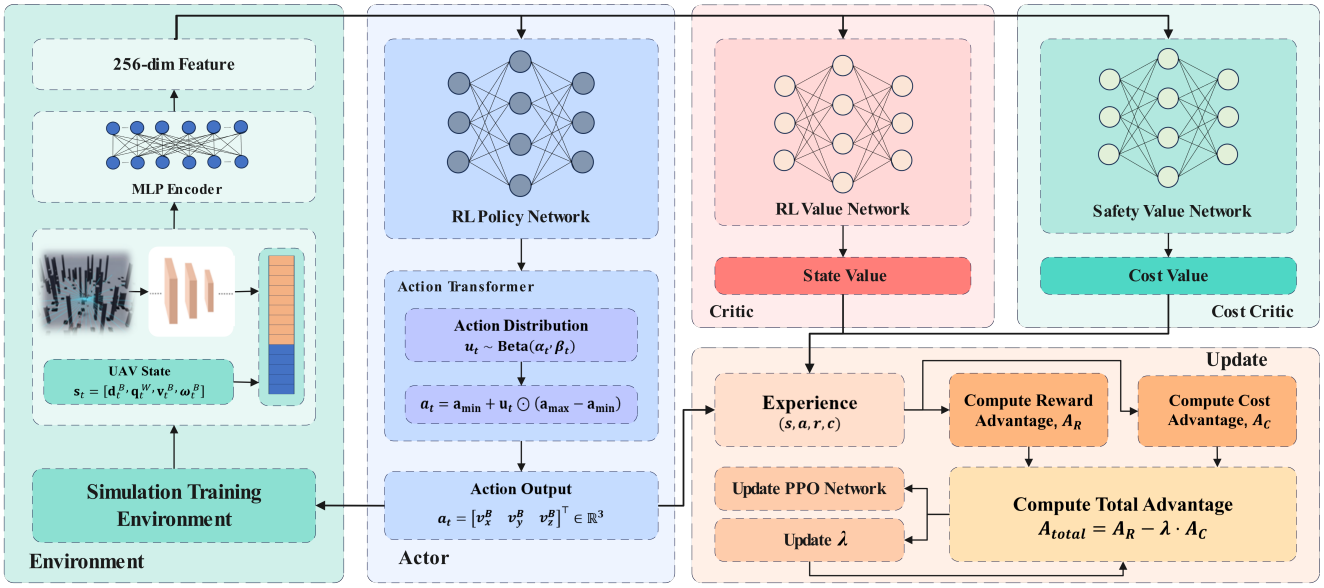

Fig. 1: Overview of the proposed safety-aware PPO framework. Sparse LiDAR observations and UAV states are encoded into a shared latent representation that feeds the Actor, Critic, and Cost Critic. The Actor outputs bounded velocity commands via a Beta policy, while the Critic and Cost Critic estimate reward and safety cost, respectively. Policy updates are performed using a Lagrangian-based objective that balances task performance and safety.

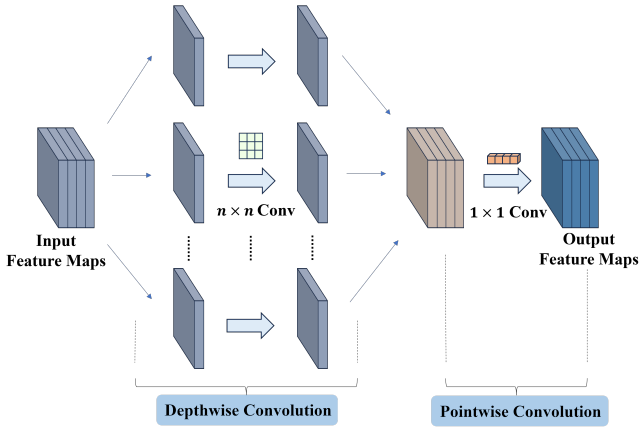

Fig. 3: Schematic of Depthwise-Separable Convolution

state variables are expressed in the drone's body frame (B). The proprioceptive state vector is defined as

$$s_t = \left[d_t^B, q_t^W, v_t^B, \omega_t^B\right] \tag{2}$$

where d_t^B denotes the normalized direction vector pointing toward the goal, q_t^W represents the attitude quaternion in the world coordinate frame, and v_t^B and ω_t^B denote the linear velocity and angular velocity in the body frame, respectively. This design preserves essential dynamic information while effectively reducing the dependence of the state representation on absolute pose, thereby facilitating the learning of motion patterns with rotational invariance.

2) Exteroceptive Perception: External environmental perception is provided by an onboard LiDAR sensor. At each time step t, the observation is represented as a 2D tensor

$$X_t \in \mathbb{R}^{1\times 36\times 4} \tag{3}$$

where the horizontal direction is discretized into 36 angular bins, and the vertical dimension corresponds to four laser beams with different elevation angles.

A proximity-based representation is used to convert raw LiDAR distances into features that better reflect collision risk by emphasizing nearby obstacles and reducing the influence of distant ones. This improves feature contrast and helps the convolutional network capture local geometric structures more effectively. The resulting proximity values are normalized to $[0, 1]$ and arranged as an angle-elevation grid, which is then fused with the ego-state for policy learning.

3) Shared Feature Encoder: To effectively fuse geometric perception information with system dynamics, we employ a shared feature encoder $f_{\text{enc}}(\cdot)$ to learn a unified representation of the observations. The LiDAR observation X_t is first processed by the convolutional encoder described earlier to extract geometric features. The ego-state s_t is linearly embedded, concatenated with the perception features, and then mapped by a multilayer perceptron into a latent representation

$$h_t = f_{\text{enc}}(s_t, X_t) \tag{4}$$

This latent representation is used as a shared input for the Actor, Critic, and Safety Value Network. Such a design reduces parameter redundancy, stabilizes the training process, and improves the generalization capability of the learned policy.

C. Reward Function Design

Efficient, smooth, and safe autonomous navigation in complex forest environments is encouraged by formulating the learning objective as the maximization of the discounted cumulative return.The overall reward function is defined as

$$r_t = r_{\text{vel}} + r_{\text{safety}} + \lambda_h r_{\text{height}} + \lambda_s r_{\text{smooth}} + r_{\text{event}} \tag{5}$$

where each term corresponds to task progress, safety constraints, flight height regulation, action smoothness, and sparse event rewards, respectively. The coefficients λ_h and λ_s control the relative importance of the corresponding terms.

1) Velocity Tracking Reward: We introduce a velocity projection-based reward to encourage the UAV to move toward the goal direction:

$$r_t^{\text{track}} = \text{clip}\left(\mathbf{v}_t \cdot \mathbf{d}_t, -v_{\max}, v_{\max}\right) \tag{6}$$

where $\mathbf{v}_t$ denotes the current linear velocity, $\mathbf{d}_t$ is the unit direction vector toward the goal, and $v_{\max}$ is a truncation threshold to prevent the agent from obtaining excessive rewards through overly high speeds, which may increase collision risk.

2) LiDAR-based Safety Reward: We construct a logarithmic obstacle potential reward based on LiDAR proximity to encourage the agent to actively maintain a safe distance from obstacles:

$$r_{\text{safety}} = \mathbb{E}_{i\in\mathscr{S}}\left[\log\left(\text{clip}\left(d_{\text{scan},i}, \varepsilon, R\right)\right)\right] \tag{7}$$

where $\mathscr{S}$ denotes the set of LiDAR beams, R is the maximum sensing range, $d_{\text{scan},i}$ is the measured distance of the i-th beam, and ε is a small constant for numerical stability. Intuitively, this term encourages the agent to maximize the average residual safe distance from surrounding obstacles.

3) Height Constraint Reward: To prevent the UAV from flying too high or touching the ground, a quadratic penalty centered on a safe altitude interval is introduced:

$$r_t^{\text{height}} = \begin{cases} -(z_t - z_{\max})^2, & z_t > z_{\max}, \\ -(z_t - z_{\min})^2, & z_t < z_{\min}, \\ 0, & \text{otherwise}. \end{cases} \tag{8}$$

where z_t denotes the current flight altitude.

4) Action Smoothness Reward: We introduce a penalty on velocity variation to reduce aggressive maneuvers and improve energy efficiency:

$$r_t^{\text{smooth}} = -\left\|\mathbf{v}_t - \mathbf{v}_{t-1}\right\|^2 \tag{9}$$

5) Sparse Event Reward: At the end of each episode, discrete rewards or penalties are applied based on the final state:

- **Goal reached:** A reward of $+100$ is given when the UAV reaches the goal within a distance of 0.5 m.
- **Collision:** A penalty of -50 is applied when a collision with obstacles is detected.
- **Abnormal state:** A penalty of -20 is assigned when the UAV exits the safety boundary (e.g., excessively high or low altitude).

D. Safe Reinforcement Learning Algorithm Design

In our work, we adopt a Lagrangian relaxation-based safe Proximal Policy Optimization (Safe PPO) algorithm as the decision-making framework. Built upon the standard PPO algorithm, explicit safety constraints are introduced, and a dual optimization mechanism is employed to dynamically balance task performance and safety. This framework is designed to enable safe and autonomous UAV navigation in complex obstacle-rich environments.

1) Actor Network and Action Modeling: We adopt a hierarchical control architecture, where the RL policy generates high-level velocity commands and a geometric controller tracks the desired motion and stabilizes the UAV dynamics.

The action space is defined as the desired body-frame velocity

$$\mathbf{a}_t = \begin{bmatrix} v_x^B & v_y^B & v_z^B \end{bmatrix}^\top \in \mathbb{R}^3 \tag{10}$$

which provides rotational invariance across different global headings.

Since the action space is continuous and bounded, the actor models the policy using a Beta distribution to avoid the distortion caused by Gaussian action clipping. Given the latent feature h_t, the actor predicts the distribution parameters

$$\alpha_t = \text{Softplus}(z_\alpha) + \varepsilon, \quad \beta_t = \text{Softplus}(z_\beta) + \varepsilon. \tag{11}$$

The action is sampled from

$$\mathbf{u}_t \sim \text{Beta}(\alpha_t, \beta_t) \tag{12}$$

and mapped to the valid velocity range

$$\mathbf{a}_t = \mathbf{a}_{\min} + \mathbf{u}_t \odot (\mathbf{a}_{\max} - \mathbf{a}_{\min}) \tag{13}$$

where $\odot$ denotes element-wise multiplication.

The resulting velocity command is transformed to the world frame and tracked using the geometric controller proposed by Lee et al [20].

2) Critic Network Structure: The Critic network is used to estimate the state value function $V(s)$ and provides a baseline for Generalized Advantage Estimation (GAE). Based on the shared latent representation h_t, the Critic outputs a scalar state value through a single linear layer. To improve numerical stability during the early stage of training, orthogonal initialization is applied to the output layer of the Critic.

3) Safety Value Network (Cost Network): To strictly limit risky behaviors of the UAV while maximizing task rewards, we introduce a safety cost function within the CMDP framework

$$C_{\text{total}} = I_{\text{collision}} + w \cdot C_{\text{margin}} \tag{14}$$

where $I_{\text{collision}}$ represents a hard constraint signal indicating collision or severe boundary violation.

The safety margin term is defined as

$$C_{\text{margin}} = \frac{\text{clamp}\left(d_{\text{safe}} - d_{\min}, 0\right)}{d_{\text{safe}}} \tag{15}$$

where $d_{\min}$ denotes the minimum obstacle distance detected by LiDAR and d_{safe} is a predefined safety threshold. When

$d_{\min} < d_{\text{safe}}$, a linear penalty is applied, providing a differentiable gradient signal before collision occurs.

Formally, our objective is to find an optimal policy π that maximizes the expected cumulative reward while satisfying an expected cost constraint:

$$\begin{aligned} \max_{\pi} \quad & J_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t r_t\right], \\ \text{s.t.} \quad & J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\frac{1}{T}\sum_{t=0}^{T} C_t\right] \le d_{\text{limit}} \end{aligned} \tag{16}$$

where

- $J_R(\pi)$ denotes the expected cumulative reward,
- $J_C(\pi)$ represents the expected average cost per step,
- C_t is the safety cost defined above,
- d_{limit} is the predefined safety tolerance threshold.

Using Lagrangian relaxation, we introduce a multiplier λ and convert the constrained problem into the following dual optimization problem:

$$\min_{\lambda}\max_{\pi}\mathscr{L}(\pi,\lambda) = J_R(\pi) - \lambda\left(J_C(\pi) - d_{\text{limit}}\right) \tag{17}$$

To compute gradients of the Lagrangian objective under the PPO framework [21], we introduce a safety value network (Cost Critic), denoted as $V_\phi^C(s_t)$, which operates in parallel with the reward value network $V_\theta^\pi(s_t)$. The Cost Critic estimates the expected cumulative cost from the current state. Its training objective is

$$\mathscr{L}_{\text{Cost}}(\phi) = \mathbb{E}_t\left[\left(V_\phi^C(s_t) - \hat{R}_t^C\right)^2\right] \tag{18}$$

where $\hat{R}_t^C$ denotes the cost return computed from sampled trajectories.

Based on these two value networks, we compute the reward advantage $A_R^\pi(s,a)$ and the cost advantage $A_C^\pi(s,a)$. To reduce variance, both advantages are estimated using Generalized Advantage Estimation (GAE).

4) Safe PPO Objective: Under the PPO framework, we define a joint advantage function:

$$A_{\text{total}}(s,a) = \hat{A}_R^{\text{GAE}}(s,a) - \lambda\hat{A}_C^{\text{GAE}}(s,a) \tag{19}$$

The clipped policy objective is then formulated as

$$\begin{aligned} \mathscr{L}_{\text{policy}}(\theta) = \mathbb{E}_t\Big[\min\Big(&\rho_t(\theta)A_{\text{total}}, \\ &\text{clip}\left(\rho_t(\theta), 1-\varepsilon, 1+\varepsilon\right)A_{\text{total}}\Big)\Big] \end{aligned} \tag{20}$$

where

$$\rho_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \tag{21}$$

This formulation suppresses high-risk actions even if they yield high reward advantages, due to the cost penalty.

5) Adaptive Update of the Lagrange Multiplier: The Lagrange multiplier λ is updated using dual gradient ascent:

$$\lambda_{k+1} = \text{ReLU}\left(\lambda_k + \eta_\lambda\left(\bar{C}_k - d_{\text{limit}}\right)\right) \tag{22}$$

where $\bar{C}_k$ denotes the average cost of the current iteration. This mechanism increases the constraint strength when safety violations occur frequently and relaxes it when the policy becomes safer.

Additionally, a linear learning rate decay is applied to both the actor and critic during training to improve convergence stability.

E. Curriculum Learning via Feature Reuse and Policy Reset

We adopt a feature-transfer-based curriculum learning strategy to mitigate training instability caused by increasing obstacle density [22]. The model is first pretrained in a low-density environment to learn stable geometric representations and fundamental dynamics control capabilities. It is then transferred to a high-density environment with the pretrained weights loaded.

During the transfer stage, the parameters of the shared feature encoder are preserved to reuse the learned environmental representation capability, while the output layer of the Actor network is orthogonally reinitialized. This design prevents the previous policy from overfitting or converging to suboptimal solutions in the new environment and restores the policy to a high-entropy exploration regime. Supported by the pretrained feature space, the agent can converge more sample-efficiently to a safe policy adapted to high-density scenarios.

IV. Experiment

A. Simulation Environment and Task Setup

We evaluate the effectiveness of the proposed method in complex unstructured environments by constructing an autonomous UAV navigation task in a high-fidelity 3D forest-like environment using the NVIDIA Isaac Sim simulation platform together with the OmniDrones framework [23].

The simulation environment is a flat $40\,\text{m} \times 40\,\text{m}$ area, as shown in Fig. 4. Cylindrical obstacles with numbers of 100, 200, and 300 are randomly placed on the ground to represent environments from sparse to highly cluttered. Obstacle geometries are randomized to increase scene diversity: widths are uniformly sampled from 0.4-1.0 m, and heights follow a piecewise distribution over $[1.0, 2.0)$, $[2.0, 4.0)$, and $[4.0, 5.0]$ m, producing multi-scale occlusions and altitude-dependent collision risks.

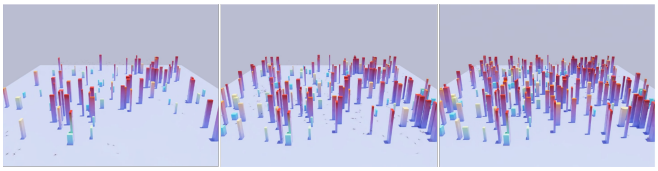

Fig. 4: Simulation environments with varying obstacle densities (100, 200, and 300 obstacles from left to right).

The UAV is equipped with a lightweight sparse multi-beam LiDAR sensor with a maximum sensing range of 4.0 m. The observations are organized as a distance tensor of size $1 \times 36 \times 4$, covering the full horizontal field of view and a limited vertical field of view. The agent starts from a random initial position and autonomously navigates toward a target location while maintaining a specified flight altitude and avoiding collisions with obstacles, the ground, and the environment boundaries.

This task evaluates both obstacle-avoidance responsiveness and goal-directed navigation stability under extremely sparse perception conditions. To ensure reproducibility, all environment generation processes are conducted with a fixed random seed, and the environment scale, obstacle distribution rules, and sensor configuration remain consistent across all experiments.

B. Evaluation Metrics

Autonomous navigation performance in complex unstructured environments is quantitatively evaluated from two complementary perspectives: task completion capability and flight safety. The evaluation metrics are defined as follows:

1) **Success Rate (SR)** The success rate is defined as the proportion of test episodes in which the UAV successfully completes the navigation task without any collision. This metric reflects the overall task completion capability of the policy.
2) **Collision Rate (CR)** The collision rate represents the proportion of test episodes that terminate prematurely due to collisions with obstacles, the ground, or the environment boundaries. It is used to evaluate the safety performance of the policy in complex environments.
3) **Average Return (AR)** The average return is defined as the mean cumulative reward obtained per episode during the evaluation phase. Provides a comprehensive measure of the performance of the policy in terms of task efficiency and compliance with safety constraints.
4) **Average Episode Length (AEL)** This metric denotes the average number of steps (or the equivalent flight distance) per episode during evaluation, reflecting the stability and efficiency of the navigation policy.

C. Module Contribution under Increasing Complexity

We conduct module-wise evaluations in three forest scenarios containing 100, 200, and 300 obstacles to systematically assess their contributions under varying environmental complexity.All methods share the same training configuration and evaluation metrics and are progressively compared, including Vanilla PPO, PPO+DSC, PPO+DSC+Safe RL, and the full model with curriculum learning (Ours). The curriculum learning strategy follows a staged transfer scheme, where training in higher-density environments is initialized from models that have converged in lower-density environments, thereby gradually increasing task difficulty.

The quantitative results are presented in Table I. Vanilla PPO achieves a success rate of 0.83594 in the 100 obstacles scenario, but its performance decreases to 0.69531 and 0.60938 in the 200 and 300 obstacles environments, indicating limited capability in modeling complex obstacle structures under sparse LiDAR observations. As obstacle density increases, navigation becomes more challenging due to fragmented observations, occlusions, and reduced free-space continuity, which further exposes the limitations of the baseline policy in cluttered environments. After introducing depthwise separable convolutions, performance improves across all scenarios, with more noticeable gains in the 300 obstacles setting. This result suggests that the proposed feature extraction module can better capture local geometric continuity and structural information from sparse LiDAR data while maintaining computational efficiency. Incorporating Safe RL further increases the success rates to 0.96875, 0.90625, and 0.86719, respectively. The improvements are particularly evident in medium- and high-density environments, highlighting the stabilizing effect of explicit safety constraints and their ability to guide the policy toward safer navigation behaviors during training.

Finally, integrating curriculum learning leads to additional gains in the 200 and 300 obstacles scenarios, achieving success rates of 0.95313 and 0.94531, effectively mitigating performance degradation as environmental complexity increases. By gradually increasing the difficulty of training environments, the agent is able to progressively adapt to more complex obstacle distributions and learn more robust navigation strategies. Notably, the improvements become more significant as the obstacle density increases, particularly in the 300 obstacles scenario, indicating that the proposed framework scales well to highly cluttered environments. Overall, these results demonstrate that the proposed components play complementary roles: the perception module enhances representation capability under sparse observations, the safety mechanism stabilizes policy optimization, and curriculum learning improves adaptability and generalization. Together, they jointly improve navigation robustness and reliability in complex unstructured environments.

D. Robust Navigation under Increasing Obstacle Density

To evaluate performance in complex unstructured environments, we compare the proposed full model (Ours) with representative reinforcement learning baselines, including PPO, off-policy methods SAC [24] and TD3 [25], and recurrent variants PPO+GRU [26] and PPO+LSTM [27]. We further investigate recurrent extensions of our framework (Ours+GRU and Ours+LSTM). All methods are evaluated in environments containing 100, 200, and 300 obstacles. As shown in Table II, most baseline methods exhibit noticeable performance degradation as obstacle density increases, whereas our method consistently achieves the highest success rates, demonstrating stronger robustness and stability in cluttered scenarios. Although incorporating GRU or LSTM introduces only marginal performance changes, it significantly increases computational cost. Ours+GRU and Ours+LSTM require approximately 2.4× and 2.5× longer training time than the proposed model, while SAC and TD3 require about 2.7× more time to reach stable performance.

TABLE I: Module impact on navigation performance under different obstacle densities.

Method	DSC	Safe RL	Curriculum	SR (100)	SR (200)	SR (300)
Vanilla PPO	×	×	×	0.83594	0.69531	0.60938
PPO + DSC	✓	×	×	0.86719	0.70313	0.66406
PPO + DSC + Safe RL	✓	✓	×	0.96875	0.90625	0.86719
Ours (Full)	✓	✓	✓	**0.96875**	**0.95313**	**0.94531**

TABLE II: Comparison with baseline RL methods under different obstacle densities.

Method	**SR (100)**	**SR (200)**	**SR (300)**
PPO	0.83594	0.69531	0.60938
SAC	0.86719	0.73438	0.66406
TD3	0.92969	0.89063	0.79688
PPO+GRU	0.85938	0.71875	0.61719
PPO+LSTM	0.87500	0.75781	0.73438
Ours	0.96875	0.95313	0.94531
Ours+GRU	0.97656	0.96875	0.91406
Ours+LSTM	0.95313	0.92969	0.88281

In addition to training efficiency, model complexity is further analyzed. As reported in Table III, the proposed model contains only 143k parameters in the main network, substantially fewer than SAC, TD3, and recurrent PPO variants, whose parameter counts typically exceed one million due to additional critic and target networks. This lightweight design reduces computational overhead and accelerates convergence, making the framework more suitable for resource-constrained onboard deployment. Considering performance, efficiency, and model complexity together, we adopt the non-recurrent version as the final model, indicating that the proposed sparse-perception-aware representation and safety-constrained learning mechanism are sufficient for robust navigation under sparse LiDAR observations.

TABLE III: Network Parameter Comparison of Different Methods

Method	**Main Network Params**	**Including Target Networks**
Ours	143,208	–
SAC	694,314	1,245,934
TD3	693,543	1,245,163
PPO+GRU	1,073,567	–
PPO+LSTM	1,337,759	–
Ours+GRU	1,609,708	–
Ours+LSTM	2,005,996	–

Fig. 5 further reports the quantitative comparison in terms of average return for environments containing 100, 200, and 300 obstacles. Overall, our method achieves higher returns across all scenarios, indicating stronger robustness in cluttered environments. In contrast, TD3 and recurrent PPO variants exhibit noticeable performance degradation as the obstacle density increases, particularly in the 300 obstacles setting. These results demonstrate that the proposed framework maintains stable navigation performance even under significantly increased environmental complexity.

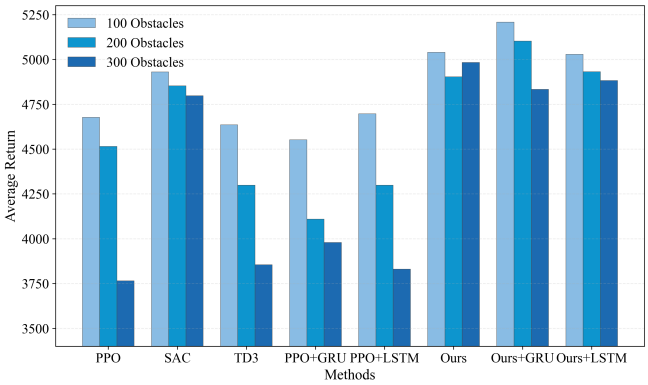

Fig. 5: Comparison with baseline RL methods under different obstacle densities.

E. High-Speed Flight under Constraints

We evaluate robustness under varying dynamic and environmental conditions by analyzing the navigation success rate (SR) across maximum flight velocities of 2–11 m/s and obstacle densities of 100, 200, and 300, as illustrated in Fig. 6. Overall, the policy maintains high performance across a wide velocity range, with SR remaining around or above 0.94 in most cases. In particular, the best performance is observed near 6 m/s in the 100 obstacles environment, where the SR approaches 0.99, indicating a favorable balance between maneuverability and control stability. This also suggests that the learned policy can effectively exploit dynamic feasibility while maintaining reliable obstacle avoidance. As the velocity increases beyond this range, the SR gradually decreases, and the degradation becomes more pronounced in denser environments. For example, in the 300 obstacles scenario, the SR shows a noticeable drop when the velocity exceeds 8 m/s, reflecting the increased difficulty of collision avoidance under limited reaction time and perception uncertainty. This trend highlights the growing challenge of safe navigation when both environmental complexity and dynamic constraints increase simultaneously. Nevertheless, the proposed method still maintains competitive performance even at higher velocities, demonstrating strong robustness and stable generalization across different dynamic constraints.

V. CONCLUSIONS

We present a safe reinforcement learning framework for UAV navigation under sparse LiDAR perception. The method integrates a lightweight perception encoder, safety-constrained policy optimization, and curriculum learning to

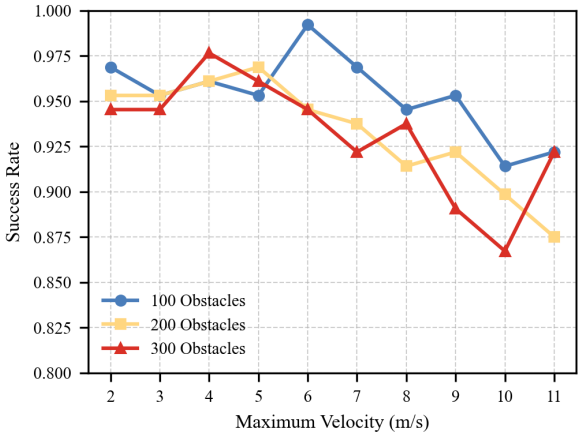

Fig. 6: Navigation success rate under different maximum flight velocities.

enhance training stability and navigation robustness. Experiments show consistent improvements over state-of-the-art reinforcement learning baselines across different obstacle densities and flight speeds. Future work will explore more diverse environments and improve robustness to sensing noise.

References

[1] J. Mao, Z. Jia, H. Gu, C. Shi, H. Shi, L. He, and Q. Wu, "Robust UAV path planning with obstacle avoidance for emergency rescue," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6, IEEE, 2025.

[2] Y. Guo and Z. Liu, "UAV path planning based on deep reinforcement learning," *Monitoring and Control (ANMC) Cooperate: Xi'an Technological University (CHINA) West Virginia University (USA) Huddersfield University of UK (UK)*, p. 81, 2023.

[3] X. Cao, H. Chen, B. Aksun-Guvenc, and L. Guvenc, "Modified hybrid A* collision-free path-planning for automated reverse parking," *arXiv preprint arXiv:2512.12021*, 2025.

[4] C. Zammit and E.-J. Van Kampen, "Comparison between A* and RRT algorithms for 3D UAV path planning," *Unmanned Systems*, vol. 10, no. 02, pp. 129–146, 2022.

[5] O. Bouhamed, H. Ghazzai, H. Besbes, and Y. Massoud, "Autonomous UAV navigation: A DDPG-based deep reinforcement learning approach," in *2020 IEEE International Symposium on circuits and systems (ISCAS)*, pp. 1–5, IEEE, 2020.

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

[7] B. Lei, W. Hu, Z. Ren, and S. Ji, "DRL-based UAV autonomous navigation and obstacle avoidance with LiDAR and depth camera fusion," *Aerospace*, vol. 12, no. 9, p. 848, 2025.

[8] L. Tai, G. Paolo, and M. Liu, "Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation," in *2017 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 31–36, IEEE, 2017.

[9] B. Lakshmi, K. C. Kasthala, N. Yamsani, G. Vasukidevi, M. Srujana, and A. Athiraja, "Enhancing autonomous driving with reinforcement learning and lidar-based object detection," in *2024 5th International Conference on Data Intelligence and Cognitive Informatics (ICDICI)*, pp. 743–750, IEEE, 2024.

[10] Y. Wang, Y. Jiang, H. Xu, C. Xiao, and K. Zhao, "Research on unmanned aerial vehicle intelligent maneuvering method based on hierarchical proximal policy optimization," *Processes*, vol. 13, no. 2, p. 357, 2025.

[11] W. Skarka and R. Ashfaq, "Hybrid machine learning and reinforcement learning framework for adaptive UAV obstacle avoidance," *Aerospace*, vol. 11, no. 11, p. 870, 2024.

[12] H. Pan, L. Han, J. Yan, and R. Liu, "Action correction-enhanced multi-agent reinforcement learning for path planning in urban environments," *Unmanned Systems*, vol. 14, no. 02, pp. 461–479, 2026.

[13] A. Kushwaha, K. Ravish, P. Lamba, and P. Kumar, "A survey of safe reinforcement learning and constrained mdps: A technical survey on single-agent and multi-agent safety," *arXiv preprint arXiv:2505.17342*, 2025.

[14] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in *International conference on machine learning*, pp. 22–31, Pmlr, 2017.

[15] W. Xu, Z. Yao, W. Li, Z. Song, Y. Song, T. Li, and Y. Li, "Tcrl: Temporal-coupled adversarial training for robust constrained reinforcement learning in worst-case scenarios," *arXiv preprint arXiv:2602.13040*, 2026.

[16] H. Ahmad, E. Sabouni, A. Wasilkoff, P. Budhraja, Z. Guo, S. Zhang, C. Fan, C. Cassandras, and W. Li, "Hierarchical multi-agent reinforcement learning with control barrier functions for safety-critical autonomous systems," *arXiv preprint arXiv:2507.14850*, 2025.

[17] J. Zhang, S. Singh, *et al.*, "Loam: Lidar odometry and mapping in real-time.," in *Robotics: Science and systems*, vol. 2, pp. 1–9, Berkeley, CA, 2014.

[18] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1366–1373, IEEE, 2017.

[19] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[20] T. Lee, M. Leok, and N. H. McClamroch, "Control of complex maneuvers for a quadrotor UAV using geometric methods on SE (3)," *arXiv preprint arXiv:1003.2005*, 2010.

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

[22] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

[23] J. Chen, C. Yu, Y. Xie, F. Gao, Y. Chen, S. Yu, W. Tang, S. Ji, M. Mu, Y. Wu, *et al.*, "What matters in learning a zero-shot sim-to-real rl policy for quadrotor control? a comprehensive study," *IEEE Robotics and Automation Letters*, 2025.

[24] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, *et al.*, "Soft actor-critic algorithms and applications," *arXiv preprint arXiv:1812.05905*, 2018.

[25] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *International conference on machine learning*, pp. 1587–1596, PMLR, 2018.

[26] C. Zhang, C. Tao, Y. Xu, W. Feng, J. Rasol, T. Hui, and L. Dong, "Autonomous defense of unmanned aerial vehicles against missile attacks using a GRU-based PPO algorithm," *International Journal of Aeronautical and Space Sciences*, vol. 25, no. 3, pp. 1034–1049, 2024.

[27] B. Ghojogh and A. Ghodsi, "Recurrent neural networks and long short-term memory networks: Tutorial and survey," *arXiv preprint arXiv:2304.11461*, 2023.