Safe End-to-End Autonomous Navigation for UAVs via Constrained Reinforcement Learning

1st dengyuan zhang
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—Ensuring safe and efficient navigation for Unmanned Aerial Vehicles (UAVs) in dynamic environments with moving obstacles is a fundamental challenge for real-world deployment. Traditional reinforcement learning (RL) algorithms, such as PPO and SAC, primarily maximize task rewards but lack explicit safety mechanisms, often resulting in unsafe behaviors and frequent collisions. To address this limitation, we propose a novel Constrained Soft Actor-Critic (Constrained SAC) framework that formulates UAV navigation as a Constrained Markov Decision Process (CMDP). By leveraging the Lagrangian relaxation technique, our method directly incorporates collision risk as a hard constraint into the policy optimization, guiding the agent to learn safety-aware behaviors throughout training. Moreover, we introduce a constraint-based curriculum learning strategy that adaptively tightens safety requirements as training progresses, effectively balancing exploration and safety. Experimental results in challenging simulation environments demonstrate that our approach significantly reduces collision rates and improves task success rates compared to standard RL baselines. Furthermore, preliminary real-world flight tests validate the sim-to-real transferability and practical effectiveness of our method for safety-critical UAV applications.**

***Index Terms*—Safe Reinforcement Learning, Constrained Soft Actor-Critic, Lagrangian Relaxation, Curriculum Learning, Dynamic Obstacle Avoidance.**

I. Introduction

Autonomous navigation for Unmanned Aerial Vehicles (UAVs) in dynamic environments is a critical challenge in robotics, essential for real-world applications such as search and rescue, and inspection. Traditional methods typically rely on handcrafted planners, which have limited adaptability in complex and changing environments.

Deep Reinforcement Learning (DRL) offers a powerful solution by enabling agents to learn navigation policies autonomously through interaction with the environment. However, mainstream algorithms like PPO and SAC primarily focus on maximizing task rewards and have inherent limitations in providing strict safety guarantees, which restricts their deployment in safety-critical applications.

To address this core challenge, we propose a constrained reinforcement learning framework for safe UAV navigation. Our key contributions are threefold:

- We formulate the safe navigation problem as a Constrained Markov Decision Process (CMDP) with a sparse, binary cost function that unambiguously quantifies collision events, providing a clear foundation for hard-constraint learning.
- We develop **Constrained Soft Actor-Critic (CSAC)**, which integrates safety constraints directly into the SAC framework via Lagrangian relaxation. Critically, we introduce an **asymmetric double-critic design**: using minimum for reward estimation (to prevent overestimation) and average for cost estimation (to prevent dangerous underestimation).
- We propose a **constraint-based curriculum learning** strategy that dynamically tightens the cost threshold during training, effectively resolving the exploration-safety dilemma without manual hyperparameter tuning.

Extensive experiments demonstrate that CSAC achieves superior performance compared to PPO and SAC baselines, significantly reducing collision rates (to 11.1%) while maintaining high task success rates (84.7%). Real-world flight tests further validate the practical effectiveness of our approach

II. Related Work

A. Reinforcement Learning for Dynamic Obstacle Avoidance

Algorithms like PPO [1] and SAC [2] are widely used for navigation but typically treat collisions as mere negative rewards. This 'reward-driven' approach lacks formal safety guarantees, making it unsuitable for safety-critical applications where predictable, safe behavior is paramount.

B. Constrained Reinforcement Learning

To address the safety concerns of standard RL, Safe Reinforcement Learning has emerged as a critical research area. Its core idea is to ensure that an agent's behavior adheres to a set of well-defined safety constraints while maximizing task rewards. Constrained Reinforcement Learning (CRL), which typically formulates the problem as a Constrained Markov Decision Process (CMDP) [3], is a primary framework for achieving this goal.

Within the CMDP framework, various solutions have been proposed. One class of methods relies on Safety Shields [4], which add a safety layer external to a standard RL agent to intervene or correct potentially unsafe actions. Another class of methods, more relevant to our work, internalizes

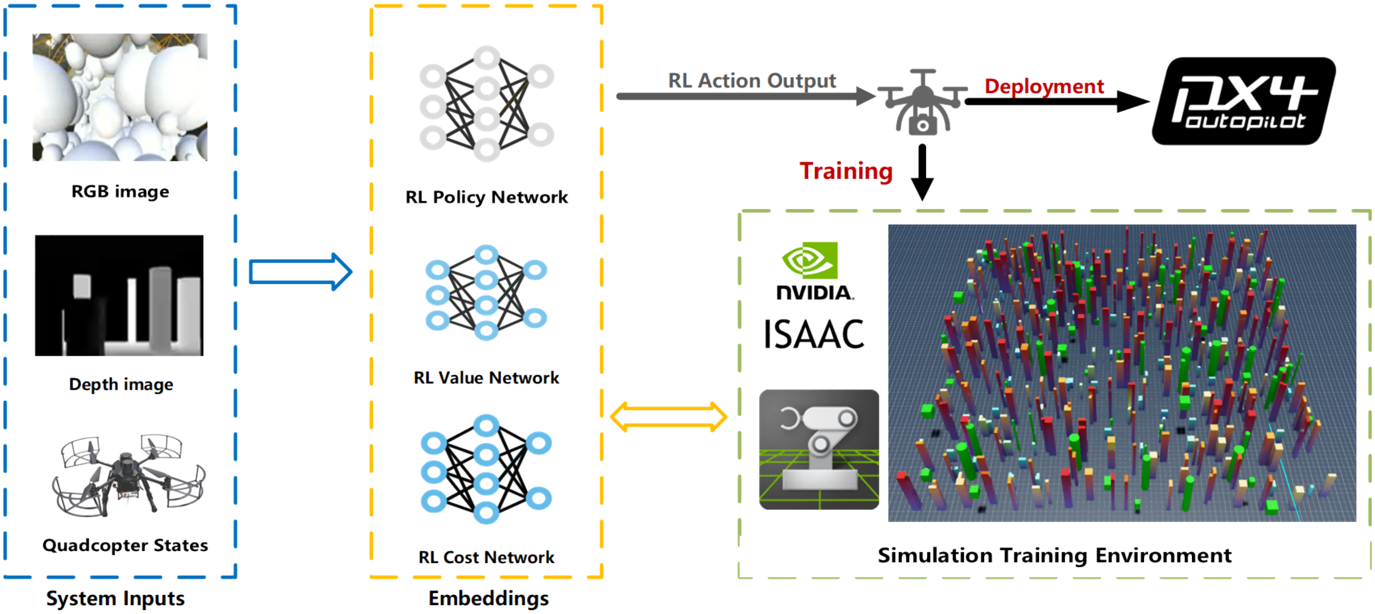

Fig. 1. Overview of the proposed algorithmic framework. The system takes as input RGB images, depth images, and quadcopter state information. These observations are processed by a reinforcement learning policy network and value network to generate high-level action commands. During training, the agent interacts with a high-fidelity simulation environment based on NVIDIA Isaac Sim to optimize its policy. For real-world deployment, the trained policy network outputs actions that are sent to the PX4 autopilot, enabling safe and autonomous UAV navigation.

safety constraints directly into the policy optimization process. For instance, approaches based on Lagrangian Relaxation [5] introduce Lagrangian multipliers to convert the constrained optimization problem into an unconstrained dual problem, thereby considering both task performance and safety costs at each policy update step.

Our work builds upon this Lagrangian-based CRL paradigm but introduces a key innovation: a *constraint curriculum learning* strategy. Instead of employing a fixed safety constraint throughout training, we dynamically adjust the stringency of the constraint. This approach effectively balances the need for exploration in the early stages of training with the need for safety in the later stages, thereby resolving the common trade-off between exploration and convergence in learning safe policies for complex dynamic environments.

III. Methodology

Our proposed method addresses the safe navigation problem by first formulating it as a Constrained Markov Decision Process (CMDP). We then introduce our core algorithm, Constrained Soft Actor-Critic (Constrained SAC), which solves this CMDP by integrating safety constraints directly into the policy optimization loop. Finally, we describe our novel constraint curriculum learning strategy designed to enhance exploration and training stability.

A. Problem Formulation as a CMDP

To formally address the safety requirements, we formulate the autonomous UAV navigation task in dynamic environments as a **Constrained Markov Decision Process (CMDP)**. A CMDP is defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, C, \gamma, d)$, where our objective is to learn an optimal policy π^* that maximizes the expected cumulative reward $J_R(\pi)$ while ensuring the expected cumulative cost $J_C(\pi)$ remains below a predefined safety threshold d. This optimization problem is formally stated as:

$$\max_{\pi} \quad J_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t, a_t)\right] \tag{1}$$

$$\text{s.t.} \quad J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t C(s_t, a_t)\right] \leq d \tag{2}$$

where τ is a trajectory sampled under policy π, with $s_t \in \mathcal{S}$ and $a_t \in \mathcal{A}$ being the state and action at timestep t.

The core elements of this CMDP framework are defined as follows:

- **States($\mathcal{S}$), Actions($\mathcal{A}$), and Rewards(R) :** Following NavRL [4], our state $\mathcal{S}$ includes UAV proprioception and environmental perception; action $\mathcal{A}$ is continuous velocity; reward R encourages goal-reaching. The transition P and discount γ follow standard MDP definitions.
- **Cost Function (C):** The cost function $C(s_t, a_t)$ is the cornerstone of our safety framework, designed to **unambiguously quantify unsafe behavior**. Unlike approaches that conflate safety penalties with task rewards, we define cost as a sparse, binary signal:

$$C(s_t, a_t) = \begin{cases} 1 & \text{if a collision occurs at time } t, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

This design offers two key advantages: first, it provides a clear, non-negotiable measure of safety violation, preventing the agent from ”haggling” task success against safety.

Second, it serves as a direct, quantifiable feedback signal for constrained optimization algorithms, forming the theoretical prerequisite for our ”hard constraint” learning approach.

- **Cost Threshold** (d)**‘:** This parameter defines the maximum allowed risk budget. However, a fixed d creates a fundamental **Exploration-Safety Dilemma**: a strict threshold hinders exploration, while a lenient one compromises final safety. To resolve this, we propose to dynamically adjust d during training. This forms the core of our **constraint-based curriculum learning** strategy, which is detailed in Section **??**.

B. Constrained Soft Actor-Critic (CSAC)

To solve the aforementioned Constrained Markov Decision Process (CMDP), we propose the **Constrained Soft Actor-Critic (CSAC)** algorithm. CSAC directly integrates safety constraints into the Soft Actor-Critic (SAC) framework via **Lagrangian Relaxation**, transforming the constrained optimization problem into an unconstrained saddle-point problem. We introduce a learnable Lagrange multiplier $\lambda \geq 0$ and formulate the following Lagrangian:

$$\mathcal{L}(\pi, \lambda) = J_R(\pi) - \lambda(J_C(\pi) - d) \tag{4}$$

Finding the saddle point of this constrained problem is equivalent to alternately optimizing a set of parameterized functions within a training loop via gradient descent: a policy network (actor) π_ϕ, two reward critic networks $Q^R_{\theta_1}, Q^R_{\theta_2}$, two cost critic networks $Q^C_{\psi_1}, Q^C_{\psi_2}$, and two adaptive parameters—the entropy temperature α and the Lagrange multiplier λ.

1) Critics Update: Both the reward and cost critics are updated by minimizing the Mean Squared Bellman Error (MSBE). For each batch of transitions (s, a, r, c, s') sampled from the replay buffer $\mathcal{D}$, we first compute the target Q-values for reward and cost using the **target networks** (denoted by parameters $\bar{\theta}_j$ and $\bar{\psi}_j$, which are slowly updated copies of the main networks). Crucially, we employ an **Asymmetric Double-Critic Update Rule**:

$$y_R(s') = r + \gamma \left(\min_{j=1,2} Q^R_{\bar{\theta}_j}(s', a') - \alpha \log \pi_\phi(a'|s') \right) \tag{5}$$

$$y_C(s') = c + \gamma \left(\frac{1}{2} \sum_{j=1,2} Q^C_{\bar{\psi}_j}(s', a') \right) \tag{6}$$

where $a' \sim \pi_\phi(\cdot|s')$. For the reward target y_R, we use the standard Clipped Double-Q-learning technique (taking the minimum) to mitigate Q-value overestimation. For the cost target y_C, however, we take the **average**. This design choice is motivated by safety: in safety-critical tasks, the **underestimation** of cost is far more dangerous than its overestimation. Taking the average provides a more stable and less biased estimate of cost, guiding the policy towards more conservative and safer actions.

Algorithm 1 The CSAC Training Loop with Asymmetric Critics and Warm-up

```
Initialize: Actor $\pi_\phi$, reward critics $Q^R_{\theta_j}$, cost critics $Q^C_{\psi_j}$ ($j = 1, 2$)
Initialize: Target network parameters $\bar{\theta}_j \leftarrow \theta_j$, $\bar{\psi}_j \leftarrow \psi_j$ (hard copy)
Initialize: $\alpha$, $\lambda$, replay buffer $\mathcal{D}$, warm-up steps $N_{warmup}$
Initialize: Target update coefficient $\tau$
— Warm-up Phase —
for $t = 1$ to $N_{warmup}$ do
    Sample random action $a_t$, execute in environment, and observe $(r_t, c_t, s_{t+1})$
    Store the transition $(s_t, a_t, r_t, c_t, s_{t+1})$ in $\mathcal{D}$
end for
— Training Phase —
for each training iteration do
    Execute action $a_t \sim \pi_\phi(\cdot|s_t)$, observe $(r_t, c_t, s_{t+1})$, and store in $\mathcal{D}$
    Sample a minibatch $\mathcal{B} = \{(s, a, r, c, s')\}$ from $\mathcal{D}$
    — Update Critics (Asymmetric) —
    Compute targets $y_R$, $y_C$ (Eq. 5-6); update critics $\theta_j$, $\psi_j$ (Eq. 7-8)
    — Update Actor & Adaptive Params —
    Update actor and adaptive params: $\phi$, $\alpha$, $\lambda$ (Eq. 9-11)
    — Update Target Networks —
    Soft update target networks using Polyak averaging:
        $\bar{\theta}_j \leftarrow \tau\theta_j + (1 - \tau)\bar{\theta}_j$, $\bar{\psi}_j \leftarrow \tau\psi_j + (1 - \tau)\bar{\psi}_j$
end for
```

The loss functions for the reward and cost critics are then defined as:

$$L(\theta_j) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\left(Q^R_{\theta_j}(s, a) - y_R(s') \right)^2 \right] \tag{7}$$

$$L(\psi_j) = \mathbb{E}_{(s,a,s')\sim\mathcal{D}} \left[\left(Q^C_{\psi_j}(s, a) - y_C(s') \right)^2 \right] \tag{8}$$

2) Actor and Adaptive Parameters Update: The actor and the two adaptive parameters are updated in the same optimization step, each minimizing its own independent loss.

a) Actor Loss: The actor's loss function also follows the asymmetric design:

$$\begin{aligned} L(\phi) = \mathbb{E}_{s\sim\mathcal{D}, a\sim\pi_\phi} \Big[& \alpha \log \pi_\phi(a|s) - \min_{j=1,2} Q^R_{\theta_j}(s, a) \\ & + \lambda \cdot \left(\frac{1}{2} \sum_{j=1,2} Q^C_{\psi_j}(s, a) \right) \Big] \end{aligned} \tag{9}$$

By performing gradient ascent on this loss (or minimizing its negative), the policy network is guided to produce actions that yield higher rewards, lower (and more robustly estimated) costs, and higher entropy.

b) Entropy Temperature and Lagrange Multiplier Loss: The loss functions for the entropy temperature α and the Lagrange multiplier λ are defined as follows, which serve to drive the policy's entropy towards a target value $\mathcal{H}_{target}$ and to enforce the cost constraint, respectively:

$$L(\alpha) = \mathbb{E}_{a\sim\pi_\phi}\left[-\alpha(\log \pi_\phi(a|s) + \mathcal{H}_{target})\right] \tag{10}$$

$$L(\lambda) = \mathbb{E}_{\substack{s\sim\mathcal{D}\\ a\sim\pi_\phi}}\left[\lambda\left(d - \frac{1}{2}\sum_{j=1,2} Q^C_{\psi_j}(s,a)\right)\right] \tag{11}$$

The complete training procedure, including a warm-up phase, is summarized in Algorithm 1.

C. Constraint-based Curriculum Learning

A key challenge in CSAC is setting the cost threshold d, which creates an **Exploration-Safety Dilemma**: if d is too stringent, the agent may under-explore and converge to a suboptimal safe policy; if too lenient, the agent may learn reckless behaviors that are difficult to correct later.

To resolve this, we propose a **Constraint-based Automated Curriculum Learning** strategy that **dynamically tightens** d **during training** without manual tuning. The training proceeds through three auto-switching stages:

1) **Exploration-Priority:** Start with a lenient (large) d to provide a "budget for mistakes," encouraging bold exploration to learn task fundamentals.
2) **Balanced Transition:** When the agent's average cumulative cost consistently stays below the current d for a predefined number of epochs, automatically tighten d to a more demanding level.
3) **Safety-Priority:** This cycle repeats until d approaches a very strict value (e.g., near zero) in late training, compelling the agent to refine its policy to satisfy stringent safety requirements.

This adaptive strategy effectively resolves the exploration-safety conflict, significantly improving convergence speed and stability while yielding a final policy that balances high task success with superior safety.

D. Network Architecture

Given the heterogeneous nature of our state representation, a specialized feature extraction pipeline is designed to process the multi-modal inputs. Specifically:

- **Static Obstacle Perception:** For the 2D range-image generated from Occupancy Grid Map, we employ a Convolutional Neural Network (CNN). The architecture of the CNN is adept at capturing local spatial dependencies within the environmental layout.
- **Dynamic and Internal State Processing:** For low-dimensional vector data, such as the states of dynamic obstacles and the UAV's internal states, we utilize a Multi-Layer Perceptron (MLP) for encoding.

The feature embeddings extracted by these distinct networks are subsequently concatenated into a unified, high-dimensional feature vector. This vector, which comprehensively represents the agent's complete state at a given moment, serves as the common input for the policy network (Actor) and the value/cost networks (Critics).

IV. RESULT AND DISCUSSION

To evaluate the proposed framework, we present our training results under different configurations and conduct simulation and physical flight tests in various environments. The policy was trained in NVIDIA Isaac Sim on a NVIDIA GeForce RTX 4090 GPU for around 18 hours. The maximum velocity of the robot is set to 2.0 m/s. The simulation experiments are conducted on the RTX 4090 desktop, while computations for the physical flights are performed on our quadcopter's onboard computer (NVIDIA Jetson Orin NX). An Intel RealSense D435i camera is utilized for static and dynamic obstacle perception. The static and dynamic feature extractors use 3-layer convolutional neural networks, outputting embeddings of sizes 128 and 64, respectively. The policy network consists of a two-layer multi-layer . All networks and learnable parameters are trained using the ADAM optimizer. To optimize training stability and efficiency, we set distinct learning rates for different components: the actor network's learning rate is set to 5×10^{-5}, the reward critic's to 5×10^{-4}, and the cost critic's to 1×10^{-3}. Furthermore, the learning rates for the automatically tuned entropy temperature (α) and the Lagrange multiplier (λ) are both set to 1×10^{-5}.The reward discounting factor is set to 0.99.

A. RL Training Results

The learning dynamics of our Constrained SAC (CSAC) and the baseline algorithms are analyzed by their training curves, as shown in Figure 3.

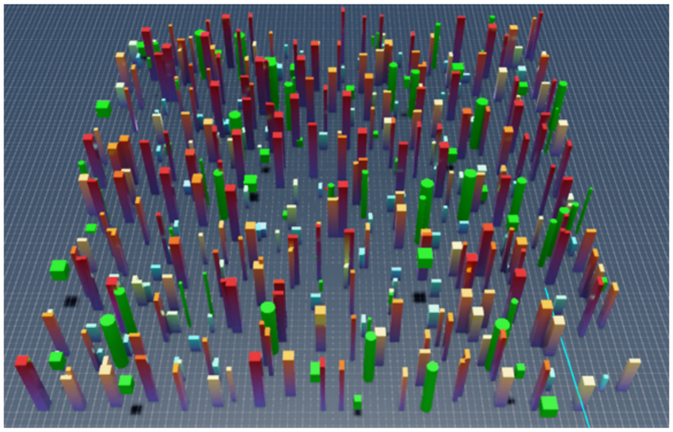

Fig. 2. Visualization of the robot training environment. The 50 m × 50 m area contains up to 100 dynamic and 350 static obstacles, with obstacle density gradually increased during training to enhance task difficulty and generalization.

The results highlight three key advantages of our proposed method:

- **Superior Performance and Safety:** CSAC achieves the highest final success rate of approximately 85%, significantly outperforming SAC (approx. 80%) and PPO (approx. 65%). Concurrently, it maintains the lowest collision rate of around 12%, which is substantially better than SAC's 15% and PPO's 30%. This confirms that our framework excels in both task completion and safety.

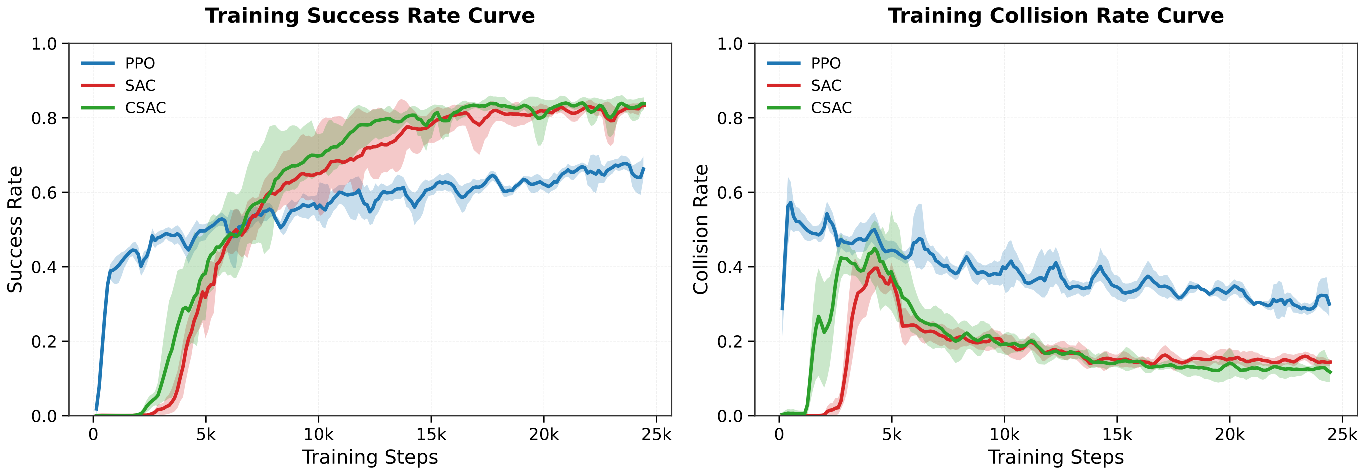

Fig. 3. Learning curves for different algorithms during training. **Left**: Success rate (reach) comparison; **Right**: Collision rate comparison. The solid lines denote the mean over multiple random seeds, and the shaded regions represent standard deviations. CSAC achieves faster convergence, higher final success rate, and lower collision rate compared to PPO and SAC.

- **Faster Convergence:** The learning curves for both success and collision rates reveal that CSAC converges significantly faster than the baselines. The success rate curve shows a steeper ascent, while the collision rate drops more rapidly after an initial exploratory phase.
- **Effective Curriculum Learning:** The transient peak in CSAC's collision rate during early training is a direct manifestation of our curriculum strategy. By starting with a lenient safety constraint, the agent is encouraged to explore, which lays the foundation for faster and safer policy convergence in later stages. This is further corroborated by the initial dip in the return curve, indicating a deliberate trade-off of short-term rewards for long-term gains.

In summary, the training results validate that CSAC achieves a superior balance of performance, safety, and convergence efficiency, effectively guided by our constraint-based curriculum learning mechanism.

B. Simulation Experiments

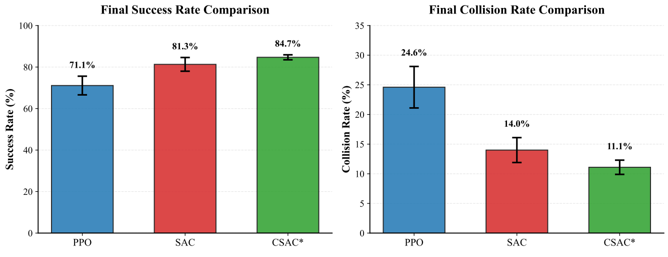

Fig. 4. Performance comparison of different algorithms on Success Rate and Collision Rate. Policies were trained with multiple random seeds and evaluated in a fixed test environment. The error bars represent the standard deviation across training runs, indicating training robustness.

To quantitatively evaluate the final performance of our proposed CSAC algorithm, we conducted a series of rigorous comparative tests. Specifically, for each algorithm (PPO, SAC, and our CSAC), we trained multiple independent policies using different random seeds. Subsequently, **all resulting policies were evaluated in a unified, pre-defined dynamic test environment (corresponding to an environment seed of 1)** to ensure a fair comparison. We focus on two key metrics: Success Rate and Collision Rate.

The final evaluation results are presented as a bar chart in Figure 4. The reported values represent the **mean performance** of the different trained policies in the test environment, with the error bars indicating the **standard deviation**.

Clear and compelling conclusions can be drawn from these experimental results:

- **Superior Overall Performance:** Our CSAC algorithm demonstrates the strongest performance across all key metrics. In terms of **Success Rate**, CSAC achieves the highest score of **84.7%**, significantly outperforming SAC (81.3%) and PPO (71.1%). This indicates that our constraint framework not only enhances safety but also more effectively facilitates task completion.
- **Significant Safety Improvement:** The advantage of CSAC is particularly pronounced in the critical safety metric of **Collision Rate**. Its average collision rate is merely **11.1%**, which is substantially lower than SAC's 14.0% and PPO's 24.6%. This provides strong evidence for the effectiveness of our constraint-based curriculum learning strategy in acquiring reliable and safe obstacle avoidance behaviors.
- **Enhanced Training Robustness:** By observing the error bars (standard deviation) in the figure, it is evident that CSAC exhibits the **smallest variance** in both metrics. This indicates that our algorithm consistently converges to high-quality policies regardless of the random seed used during training. This enhanced **training robustness** is crucial for reliable real-world deployment.

In summary, these quantitative data clearly demonstrate that, compared to state-of-the-art reinforcement learning algorithms, our proposed CSAC framework successfully achieves a superior balance between task performance, safety, and training stability.

C. Physical Flight Tests

To validate the sim-to-real transferability of our algorithm, we deployed the policy trained in the ”Mixed” scenario directly onto a physical quadrotor. As shown in Figure 5, the platform is equipped with an NVIDIA Jetson Nano and localized by a Vicon system. A human-held obstacle was used to replicate the dynamic environment.

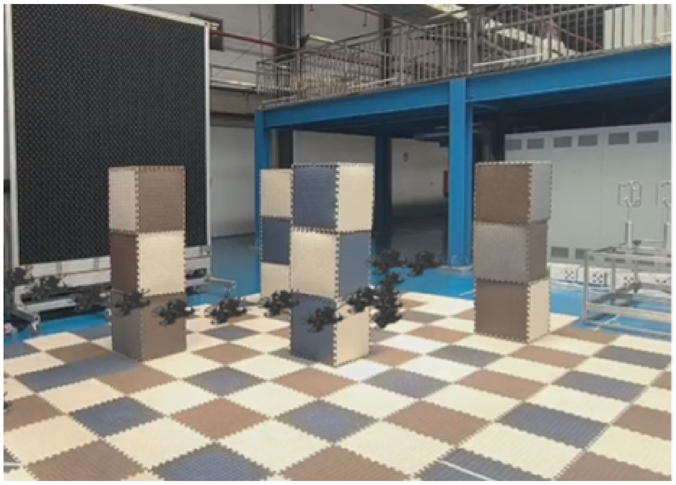

Fig. 5. Physical flight test. Left: The drone platform and experimental setup. Right: A sequence showing the drone executing a smooth, proactive avoidance maneuver against a human-held obstacle.

The drone exhibited intelligent avoidance behaviors highly consistent with those observed in simulation:

- **Proactive Avoidance:** The drone executed smooth, anticipatory detours, rather than making abrupt, last-minute evasive maneuvers.
- **Robustness:** It successfully handled random, human-induced motions not strictly present in the training data, maintaining stable flight while avoiding the obstacle.

These successful tests demonstrate our algorithm's real-world viability and its potential for deployment in complex physical environments.

V. CONCLUSION AND FUTURE WORK

Conclusion. In this paper, we introduced a novel reinforcement learning framework for safe UAV navigation in dynamic environments. Our approach, named Constrained Soft Actor-Critic (CSAC), integrates a Lagrangian-based constraint into the policy optimization process. This is synergistically combined with a constraint-based curriculum learning strategy, which dynamically tightens the safety cost threshold to guide the agent from exploration to exploitation. Extensive simulation experiments validated the superiority of our method. Compared to standard RL baselines, CSAC not only achieved a higher success rate (84.7%) but also significantly reduced the collision rate to just 11.1%, demonstrating a superior balance between task performance and safety. Furthermore, its lower performance variance across multiple seeds highlights its enhanced stability and robustness.

Future Work. While our results are promising, we identify several avenues for future research. The primary limitation is the lack of direct comparison against other paradigms of Safe RL, such as those using Safety Shields or Control Barrier Functions (CBFs). Therefore, our future work will focus on:

1) Benchmarking CSAC against a wider spectrum of state-of-the-art Safe RL algorithms.
2) Extending the framework to more complex 3D navigation and multi-agent collision avoidance scenarios.
3) Conducting extensive, long-term physical flight tests to rigorously analyze and bridge the sim-to-real gap.

REFERENCES

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

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

[3] E. Altman, *Constrained Markov decision processes*. Chapman and Hall/CRC, 1999, vol. 7.

[4] Z. Xu, X. Han, H. Shen, H. Jin, and K. Shimada, “Navrl: Learning safe flight in dynamic environments,” *IEEE Robotics and Automation Letters*, vol. 10, no. 4, pp. 3668–3675, 2025.

[5] C. Tessler, D. J. Mankowitz, and S. Mannor, “Reward constrained policy optimization,” in *International Conference on Learning Representations (ICLR)*, 2019.