PPO-Based Autonomous Drone Racing through Gate Sequences in Isaac Sim

Jinyuan Zhang, Ningze Zhong
{jinyuanz, zhong666}@seas.upenn.edu

I. Overview

We train a reinforcement learning policy using Proximal Policy Optimization (PPO) [1] to fly a simulated Crazyflie quadrotor through a seven-gate racing circuit in NVIDIA Isaac Sim. The track contains a challenging powerloop segment in which two consecutive gates share the same yaw orientation ($\psi_{g_2} = \psi_{g_3} = +\frac{\pi}{2}$), high-altitude gates ($z = 2.0\,\text{m}$), and a chicane. The layout was intended to induce a vertical loop, although the final policy often bypasses that maneuver. Our design draws on prior work in autonomous drone racing [2], [3], [4] and reactive aerobatic flight [5]. We also explored the Eureka framework [6] for reward tuning. The final system uses four elements: (i) a compact reward with dense progress shaping, sparse gate-pass bonuses, and a per-step time penalty, (ii) a 36-dimensional body-frame observation space, (iii) a five-stage reset curriculum, and (iv) domain randomization of dynamics parameters. The best policy, trained with 4096 parallel environments, completes three laps in 18.04 s.

II. Reward Function Design

We initially adopted the projection-based progress reward from Song et al. [2], which measures advancement along the line segment between adjacent gate centers. That formulation was sufficient for standard gate sequences, but it did not produce successful powerloop traversals. We then redesigned the reward by combining insights from prior work [3], [4], [5] and briefly explored Eureka [6] for reward tuning. The final reward has three active terms:

$$r(t) = w_{\text{prog}}\, r_{\text{prog}}(t) + w_{\text{gate}}\, r_{\text{gate}}(t) + w_{\text{time}}\, r_{\text{time}}. \tag{1}$$

Upon terminal failure, the reward is overridden by a death cost c_{death}. The active reward weights in the current implementation are listed in Table I.

TABLE I
Active reward component weights.

Component	Symbol	Weight
Gate-pass bonus (sparse)	w_{gate}	55.0
Progress reward (dense)	w_{prog}	15.0
Time penalty (per-step)	w_{time}	-0.08
Death cost (terminal)	c_{death}	-80.0

Global progress reward. The primary dense shaping signal measures the drone's advancement along the racing line. We define global progress as

$$G(t) = n_{\text{passed}}(t) + 1 - \tanh\!\left(\frac{d(t)}{\sigma_d}\right), \tag{2}$$

where $n_{\text{passed}}(t) \in \mathbb{Z}_{\geq 0}$ is the cumulative number of gates passed, $d(t) = \|\mathbf{p}_{\text{drone}}(t) - \mathbf{p}_{g_i}\|_2$ is the Euclidean distance to the current target gate center $\mathbf{p}_{g_i}$, and $\sigma_d = 3.0\,\text{m}$ is a distance scale. The per-step increment is

$$r_{\text{prog}}(t) = \text{clamp}\big(G(t) - G(t{-}1),\ -1,\ 1\big). \tag{3}$$

The $\tanh$ normalization keeps the signal smooth and bounded. On steps where a gate is crossed, r_{prog} is set to zero to avoid double-counting with the gate-pass bonus. For tight same-direction gate pairs in the powerloop segment, negative progress increments are clipped to zero.

Gate-pass bonus. A sparse reward $r_{\text{gate}}(t) \in \{0, 1\}$ fires when the drone crosses a gate. In the gate-local frame $\mathcal{F}_g$ with origin at the gate center and the normal $\hat{\mathbf{x}}_g$ pointing opposite to the approach direction, crossing is detected as

$$x_g(t{-}1) > 0.1 \ \ \wedge \ \ x_g(t) \leq 0.1, \tag{4}$$

subject to the aperture constraint $|y_g(t)| < \frac{s}{2}$ and $|z_g(t)| < \frac{s}{2}$, where $s = 1.0\,\text{m}$ is the gate side length. Upon a valid crossing, the waypoint index advances as $i \leftarrow (i + 1) \bmod N$, where $N = 7$.

Time penalty. A constant per-step cost $r_{\text{time}} = 1.0$ (scaled by $w_{\text{time}} = -0.08$) discourages hovering and implicitly incentivizes speed without an explicit velocity reward, which in early experiments caused the drone to fly fast in unproductive directions.

Ablated reward terms. During development we explored four additional reward components, all disabled in the final configuration. A *speed bonus* $r_{\text{speed}} = \tanh(\|\mathbf{v}\|/3.0)$ encouraged unproductive fast flight. An *action smoothness penalty* $r_{\text{smooth}} = \|\mathbf{a}(t) - \mathbf{a}(t{-}1)\|_2^2$ reduced the aggressive motions needed for the powerloop. A *crash penalty* triggered when the contact sensor measured a nonzero net force on the body after a 100-step warm-up. This made the policy too conservative near gate edges. An *entry half-plane reward* gave a one-time bonus when the drone returned to the correct approach side ($x_g > 0.15\,\text{m}$) before a tight same-direction gate pair. Mid-loop resets were more effective, so this term was removed.

III. Observation Space Design

Following Kaufmann et al. [3] and Song et al. [4], the policy receives a 36-dimensional observation vector $\mathbf{o}(t) \in \mathbb{R}^{36}$:

$$\mathbf{o}(t) = \big[\, \mathbf{v}^b,\ \mathrm{vec}(\mathbf{R}_{bw}),\ \mathbf{c}^b_{i,1:4},\ \mathbf{c}^b_{i+1,1:4} \,\big]. \tag{5}$$

The components are summarized in Table II.

TABLE II
Observation vector components ($\dim = 36$).

Group	Component	Dim	Frame
Ego-state	Linear velocity $\mathbf{v}^b$	3	Body
	Rotation matrix $\mathrm{vec}(\mathbf{R}_{bw})$	9	B→W
Gate info	Current gate corners $\mathbf{c}^b_{i,1:4}$	12	Body
	Next gate corners $\mathbf{c}^b_{i+1,1:4}$	12	Body
	Total	**36**	

Ego-state (12D). The linear velocity in the body frame $\mathbf{v}^b \in \mathbb{R}^3$ is read from the robot's center-of-mass velocity. The body-to-world rotation matrix $\mathbf{R}_{bw} \in SO(3)$ is converted from the world-frame quaternion $\mathbf{q}_w = (q_w, q_x, q_y, q_z)$ and flattened row-major into $\mathrm{vec}(\mathbf{R}_{bw}) \in \mathbb{R}^9$. We use the full rotation matrix rather than the quaternion to avoid sign ambiguity and to provide a smoother representation.

Gate geometry (24D). For each of the current (i) and next ($i{+}1 \bmod N$) target gates, the four corner positions in the body frame are computed as

$$\mathbf{c}^b_{j,k} = \mathbf{R}_{bw}^{\top}\big(\mathbf{c}^w_{j,k} - \mathbf{p}_{\text{drone}}\big), \quad k = 1, \ldots, 4, \tag{6}$$

where $\mathbf{c}^w_{j,k}$ are the world-frame corner positions and $\mathbf{p}_{\text{drone}}$ is the drone position. Each gate is a $1\,\text{m} \times 1\,\text{m}$ square with local corners at $\mathbf{c}^{\text{local}}_k \in \{(0, \pm 0.5, \pm 0.5)\}$. The corners implicitly encode gate position, orientation, distance, and size. Absolute world position is omitted to reduce overfitting to track-specific coordinates. The implementation also supports an optional privileged critic that appends body-frame angular velocity and world position, yielding a 42-dimensional critic observation during training.

IV. Reset Strategy and Curriculum Learning

Since the progress reward alone could not produce useful powerloop states, early on-policy rollouts rarely reached the gate 2→3 segment. We therefore used curriculum learning to reshape the initial-state distribution over training. Early stages focus on stable starts and the first gate. Later stages unlock harder mid-track states while keeping a nonzero fraction of gate-0 starts so the policy does not forget end-to-end racing from the beginning of the track. Inspired by Han et al. [5], we designed a five-stage curriculum indexed by the training progress ratio $\rho = k/K$, where k is the current iteration and $K = 1500$ for the reported run. The stage schedule is listed in Table III.

At each stage, the curriculum controls two quantities. First, U denotes the number of unlocked reset targets, so resets may start from gates $\{0, \ldots, U-1\}$. Second, p_0 denotes the probability of resetting at gate 0. We decrease p_0 gradually from 1.0 to 0.35 rather than driving it to zero, because later training still benefits from revisiting takeoff and early-gate approach behavior.

Position randomization. At each gate g_i with yaw ψ_i, the drone is placed 0.5–3.0 m behind the gate along its normal, with lateral offset ± 1.0 m and height noise ± 0.3 m, then rotated into world coordinates via ψ_i. The orientation faces the gate with additive noise: $\Delta\phi, \Delta\theta \sim \mathcal{U}(-0.1, 0.1)$ rad and $\Delta\psi \sim \mathcal{U}(-0.2, 0.2)$ rad. Mid-track resets initialize forward velocity $v_0 \sim \mathcal{U}(1.5, 4.0)$ m/s along the gate approach direction $-\hat{\mathbf{n}}_{g_i}$.

Powerloop mid-loop resets. After $\rho > 0.10$, a separate apex-reset branch is activated for the powerloop region, i.e., the gate 2→3 segment. Up to 30% of all training resets are reassigned to this branch. A continuous phase $\alpha \sim \mathcal{U}(0, 1)$ parameterizes the sampled loop state, where $\alpha = 0$ corresponds to loop-entry states just after gate 2 and $\alpha = 1$ corresponds to near-apex states before descending toward gate 3. The sampled height is $z = 0.8 + 1.5\alpha \pm 0.15$ m, the sampled loop pitch is $\vartheta_{\text{loop}} = \alpha \cdot \vartheta_{\max}$, where $\vartheta_{\max} = \frac{2\pi}{3}$ for $\rho < 0.25$ and $\vartheta_{\max} = \pi$ thereafter, and the sampled loop speed is $v_{\text{loop}} = 2.0(1 - 0.7\alpha) \pm 0.3$ m/s. This branch exposes the policy to mid-maneuver states that on-policy rollouts rarely reach.

Domain randomization. Per-environment perturbations are applied once at construction to improve sim-to-real robustness. The randomized parameters and ranges are listed in Table IV.

V. Lessons Learned and Key Design Decisions

The dense progress reward r_{prog} was the most important design choice. Without it, the policy received almost no useful signal before the first gate pass. The $\tanh$-based shaping provided a stable long-range gradient without overwhelming the sparse gate bonus. Curriculum learning was the second key factor. Staged resets and mid-loop samples exposed the policy to states that on-policy rollouts rarely reached early in training.

Another useful finding was that the final policy often bypassed the intended vertical loop and flew around the powerloop gates instead. This detour was faster on our track. Disabling the crash penalty and action smoothness penalty also improved performance. The policy became more aggressive near gate edges, while the quadrotor dynamics already provided sufficient smoothness. The full curriculum and hyperparameters are listed in the Appendix.

A natural next step is to explore additional physics-informed reward heuristics. Examples include velocity alignment with the current gate direction, rewards that favor dynamically efficient high-speed flight, and terms that better capture the coupling among attitude, thrust, and future gate geometry. Such heuristics may encourage more time-efficient trajectories and further reduce lap time.

VI. Conclusion

We presented a PPO policy for seven-gate drone racing in Isaac Sim. The final system combines dense global progress,

sparse gate bonuses, a time penalty, a compact 36-dimensional observation, a staged reset curriculum, and domain randomization. In evaluation, the best policy completes three laps in 18.04 s.

REFERENCES

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

[2] Y. Song, M. Steinweg, E. Kaufmann, and D. Scaramuzza, "Autonomous drone racing with deep reinforcement learning," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, 2021, pp. 1205–1212.

[3] 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, 2023.

[4] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, vol. 8, no. 82, p. eadg1462, 2023.

[5] Z. Han, X. Huang, Z. Xu, J. Zhang, Y. Wu, M. Wang, T. Wu, and F. Gao, "Reactive aerobatic flight via reinforcement learning," *IEEE Robot. Autom. Lett.*, 2025.

[6] Y. J. Ma, W. Liang, G. Wang, D.-A. Huang, O. Bastani, D. Jayaraman, Y. Zhu, L. Fan, and A. Anandkumar, "Eureka: Human-level reward design via coding large language models," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2024.

APPENDIX

TABLE III
CURRICULUM STAGES. U: NUMBER OF UNLOCKED RESET TARGETS, CORRESPONDING TO GATES $\{0, \ldots, U-1\}$; p_0: PROBABILITY OF RESETTING AT GATE 0.

Stage	ρ	U	p_0	Focus
1	$[0,\ 0.05)$	1	1.00	Stable starts and gate 0 acquisition
2	$[0.05,\ 0.12)$	2	0.55	Introduce gate 1 resets
3	$[0.12,\ 0.25)$	4	0.45	Add powerloop entry and exit practice
4	$[0.25,\ 0.50)$	6	0.40	Extend to later gates and chicane entry
5	$[0.50,\ 1.00]$	7	0.35	Full-track training with residual gate 0 starts

TABLE IV
DOMAIN RANDOMIZATION PARAMETERS. NOMINAL VALUES ARE FROM THE CRAZYFLIE 2.X CONFIGURATION.

Parameter	Nominal	Range
Thrust-to-weight T/W	3.15	$\times\,(1 \pm 0.05)$
Aero drag $C_{d,xy}$	9.18×10^{-7}	$\times\,\mathcal{U}(0.5,\ 2.0)$
Aero drag $C_{d,z}$	1.03×10^{-6}	$\times\,\mathcal{U}(0.5,\ 2.0)$
PID K_p (roll/pitch)	250.0	$\times\,(1 \pm 0.15)$
PID K_i (roll/pitch)	500.0	$\times\,(1 \pm 0.15)$
PID K_d (roll/pitch)	2.5	$\times\,(1 \pm 0.30)$
PID K_p (yaw)	120.0	$\times\,(1 \pm 0.15)$
PID K_i (yaw)	16.7	$\times\,(1 \pm 0.15)$
Motor time constant τ_m	0.005 s	fixed

TABLE V
PPO AND NETWORK HYPERPARAMETERS.

Hyperparameter	Value
Discount factor γ	0.99
GAE λ	0.95
Clip ratio ϵ	0.2
Learning rate η	3×10^{-4} (adaptive)
Target KL divergence D_{KL}	0.01
Entropy coefficient	0
Max gradient norm	1.0
Steps per env per iteration	24
Number of parallel envs	4096
Mini-batches / Epochs	4 / 5
Batch size	$4096 \times 24 = 98{,}304$
Total iterations K	1500 (reported run)
Actor hidden layers	$[128, 128]$, ELU
Critic hidden layers	$[512, 256, 128, 128]$, ELU
Initial noise std	1.0
Simulation rate	500 Hz ($dt = 0.002$ s)
Policy rate	50 Hz (decimation = 10)
Episode length	30 s ($\sim$1500 steps)
Action space	4 (collective thrust + body-rate commands)