TINTIN: Thrust-Integrated Neural Touchdown with Reinforcement Learning and INertial Navigation

Qilong (Jerry) Cheng† N18422055, Juncheng Zhou† N16975306

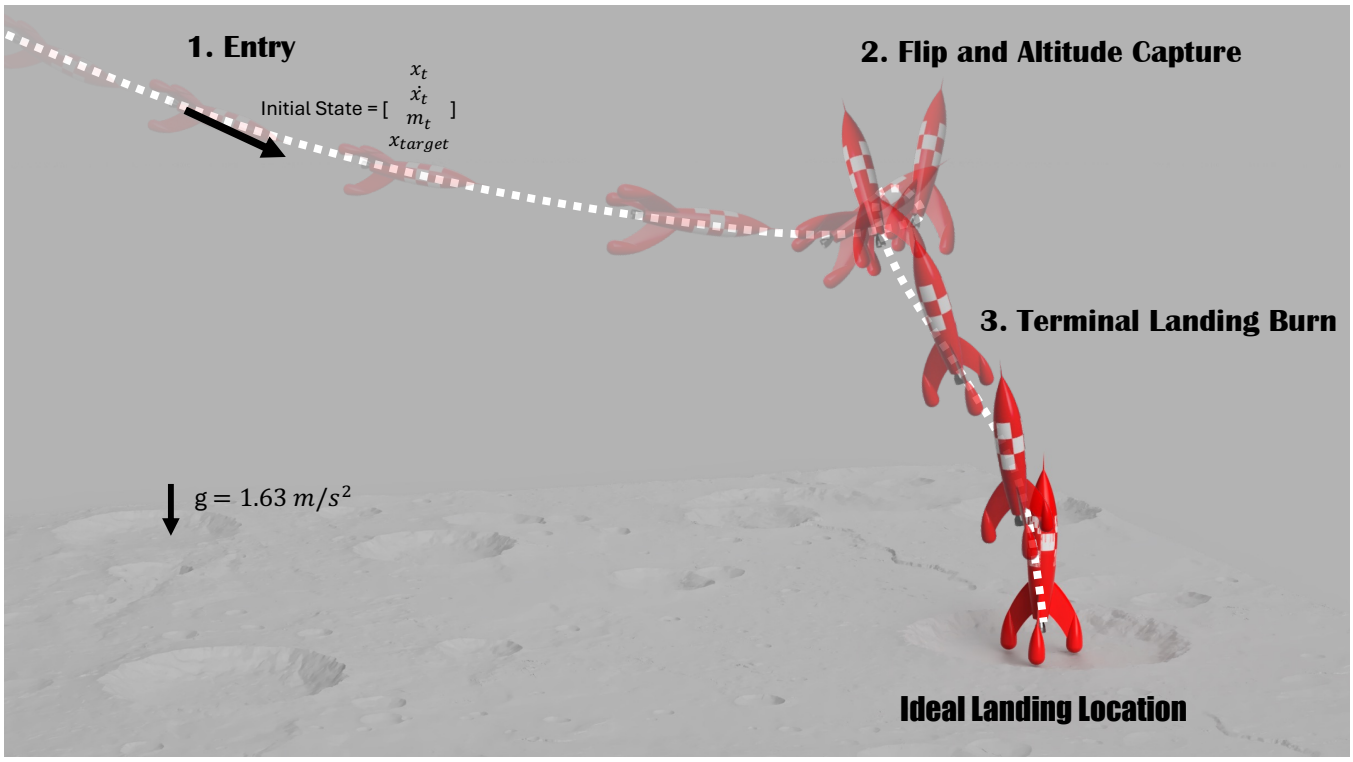

Fig. 1. System overview of the lunar rocket-landing pipeline. The trajectory comprises three phases: (1) *Entry*, (2) *Flip & attitude capture*, and (3) *Terminal landing burn* to a pre-specified target pad. An RL policy observes vehicle pose/velocity relative to the target and outputs throttle plus 2-DoF gimbal commands (pitch, yaw) to achieve pose-accurate touchdown while minimizing propellant under lunar gravity ($g \approx 1.62$ m/s^2).

***Abstract*—We present a deep reinforcement learning (DRL) approach for Integrated Guidance and Control (IGC) of a variable-mass 6-DoF lunar lander. A Soft Actor–Critic (SAC) policy maps full-state observations directly to thrust and gimbal commands, replacing traditional trajectory-planning and control loops. Training uses physics-informed reward shaping, curriculum learning, and domain randomization across a deployment ellipse. The resulting policy achieves robust soft landings with near-vertical glideslope, high Monte Carlo success rates, and quasi–fuel-optimal performance across wide variations in initial pose, velocity, and mass. These results demonstrate DRL-based IGC as a viable alternative to classical powered-descent guidance methods.**

This work is supported by Professor Benjamin Rivière's course ROB-GY 7863. The document is served as the Project 2 Report.

† denotes equal contribution.

I. Introduction

Precision powered descent is essential for future lunar missions, requiring controllers that remain robust and fuel-efficient under nonlinear 6-DoF dynamics, thrust–attitude coupling, actuator limits, and time-varying mass. Classical systems follow a decoupled GNC pipeline—navigation estimates state, guidance generates a trajectory, and a tracking controller executes it. Although effective for nominal missions, this decomposition can introduce thrust–vector cancellation and degraded robustness under large off-nominal states or rapid changes in mass and inertia.

Integrated Guidance and Control (IGC) removes this separation by learning both *what trajectory to follow* and *how to actuate* in a single mapping. Landing poses unique challenges: feasible descent depends strongly on velocity direction, yaw is unactuated, thrust–attitude coupling is highly

nonlinear, and the state space spans a wide deployment ellipse rather than a single initial pose. These factors make global planning and local stabilization inseparable and motivate learning-based approaches.

Model-based strategies such as iLQR/SLQ-MPC [1]–[4] and convex Powered-Descent Guidance (PDG) [5], [6] provide fuel-efficient trajectories but rely on a separate attitude controller and degrade under large disturbances or unmodeled mass variations. Recent RL approaches address some of these limitations: open-source 6-DoF landing simulators such as *Landing Starships* [7] and *Starship Landing Gym* [8], and recent high-fidelity 6-DoF studies [9], show that SAC/PPO can learn flip maneuvers and terminal descent. However, these works typically employ simplified mass models, limited initial-condition diversity, or reduced emphasis on global robustness across deployment ellipses.

In this work, we develop a DRL-based IGC system for a 6-DoF, variable-mass lunar lander trained over a deployment ellipse. A Soft Actor–Critic policy maps the full vehicle state directly to throttle and gimbal commands. Learning is enabled by a high-fidelity MuJoCo simulator, physics-informed reward shaping with attitude-gated terms, curriculum learning that expands the feasible set of initial conditions, and domain randomization over position, velocity, orientation, yaw, and mass.

Our contributions are:

- *RocketGym*: a high-fidelity 6-DoF MuJoCo environment modeling variable-mass rigid-body dynamics and thrust–attitude coupling.
- A DRL-based **integrated guidance and control** policy trained with curriculum learning and domain randomization for robustness across a deployment ellipse.
- Extensive Monte Carlo evaluation demonstrating accurate, robust, and quasi–fuel-optimal landings.

II. Modeling of the *RocketGym* Environment

We develop *RocketGym*, a MuJoCo-based simulator featuring a 6-DoF rocket body with a 2-DoF gimbaled engine and time-varying mass. The rocket geometry is adapted from the Adventures of TINTIN design, and the lunar terrain is imported from high-fidelity Unreal Engine assets. The lunar descent task is formulated as a fully observable MDP with full 6-DoF rigid-body dynamics, mass depletion, gimbaled thrust actuation, and environmental forces. The control objective is minimum-fuel landing while stabilizing position, attitude, and vertical descent toward the designated target zone.

Environment parameters: The rocket is modeled as a 100 m vehicle with a dry mass of 2×10^5 kg and a center of mass located mid-body, 50 m above the ground at touchdown. The gimbaled pitch–roll joint is positioned 30 m below the center of mass, and the landing target is fixed at $(0, 0, 0)$.

State: The rocket state is defined as

$$s_t = \left[\, \mathbf{p},\ \mathbf{v},\ \mathbf{q},\ \boldsymbol{\omega},\ m,\ \mathbf{I} \,\right]^{\mathsf{T}} \in \mathbb{R}^{17}, \tag{1}$$

Mass depletion and inertia scaling follow $\dot{m} = -T/(I_{sp} g_0)$ and $I(t) = I_0(m/m_0)$, where $I_{sp} = 400$ s approximates a high-performance vacuum engine for lunar descent and $g_0 = 9.81$ m/s^2 is standard gravity.

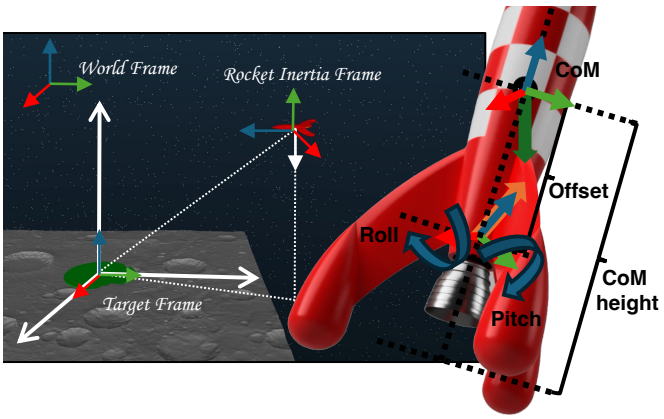

Figure 2: Illustration of the coordinate frames, initial pose, rocket twist (linear and angular velocities), and the commanded control inputs (gimbaled thrust and throttle).

Action: The control input is

$$u_t = [T, \theta_p, \theta_y]^{\mathsf{T}} \in \mathbb{R}^3, \tag{2}$$

and is obtained from normalized commands $[u_T, u_p, u_y] \in [-1, 1]^3$ via $T = T_{\max}\frac{u_T+1}{2}$,$\theta_p = 30°u_p$, $\theta_y = 30°u_y$. Here, $T_{\max}$ provides a thrust-to-weight ratio of approximately 3 (about 2.5×10^7 N). The gimbal angles are limited to $\pm 30°$. All remaining physical parameters follow the *RocketGym* specification in section A.

Initial Conditions: The rocket starts at a randomized pose within an envelope withhorizontal offset $x \in [500 \pm 20]$ m, lateral offset $y \in [0 \pm 20]$ m, altitude $z \in [500 \pm 50]$ m, and initial speed up to 50 m/s. Initial attitude, pitch and roll of the rocket, are also randomized within small bounds.

Termination Conditions: An episode ends if any of those conditions are met: 1) horizontal distance > 700 m, 2) speed > 200 m/s, 3) tilt angle $> 100°$, 4) CoM height is $< 55m$ or 5) a time limit of 2000 steps is reached.

Success Criteria: A landing is successful if the rocket touches down within 80 m of the target, with tilt $\leq 15°$ and vertical speed between 0 and 20 m/s.

III. Classical Control - PID Baseline

To benchmark our RL controller later, we implemented a simple 2D PID baseline controller with a reduced state

$$s = [\, x,\ \dot{x},\ y,\ \dot{y},\ \theta,\ \dot{\theta} \,]^{\mathsf{T}},$$

with thrust magnitude T and gimbal angle θ_g.

1) Outer-loop PD control:

$$a_x^{\text{des}} = K_{p,x}(x^* - x) + K_{d,x}(0 - \dot{x}), \tag{3}$$

$$a_y^{\text{des}} = K_{p,y}(y^* - y) + K_{d,y}(0 - \dot{y}), \tag{4}$$

with gravity compensation:

$$a_x^T = a_x^{\text{des}}, \qquad a_y^T = a_y^{\text{des}} + g.$$

2) Conversion to thrust and direction:

$$T_{\text{des}} = m\sqrt{(a_x^T)^2 + (a_y^T)^2},$$

$$\phi_{\text{des}} = 2(a_y^T,\, a_x^T).$$

Gimbal command:

$$\theta_g = \phi_{\text{des}} - \theta - K_\theta \dot{\theta}.$$

3) Final PID control law:

$$u = \begin{bmatrix} T \\ \theta_g \end{bmatrix} = \begin{bmatrix} \text{clip}(T_{\text{des}}) \\ \text{clip}(\phi_{\text{des}} - \theta - K_\theta \dot{\theta}) \end{bmatrix}.$$

IV. Deep Reinforcement Learning for Integrated Guidance and Control

In this section, we formulated the rocket landing problem as a continuous-control MDP and evaluated several DRL algorithms for IGC landing.

The raw observation contains the full translational and rotational state,

$$o_t = [\,\mathbf{p}, \mathbf{v}, \mathbf{q}, \boldsymbol{\omega}, m\,] \in \mathbb{R}^{14},$$

However, due to the rotational symmetry of the landing problem—where approaching from any azimuth is equivalent—the translational state can be reduced to the lateral distance $r = \sqrt{x^2 + y^2}$ and altitude z. Exploiting this symmetry substantially lowers the effective state dimensionality while preserving full 6-DoF dynamics fidelity.

The action vector consists of commanded thrust and gimbal angles,

$$a_t = [T, \theta_p, \theta_y],$$

obtained by linearly mapping normalized policy outputs $u_t \in [-1, 1]^3$ to physical engine commands. All algorithms operate under this shared observation–action interface. For DQN training, the continuous action space was discretized by uniformly binning thrust, pitch, and roll into 3 levels each (thrust $\in [0, 1]$, pitch/roll $\in [-0.1, 0.1]$), yielding a discrete action set of $3 \times 3 \times 3 = 27$ actions. The agent then selected from this finite table rather than issuing continuous commands.

A. Reward Shaping

The reward is designed to balance continuous dense rewards with termination sparse rewards to speed up training porcess. At each timestep, the shaping terms encourage the rocket to: 1) minimize attitude tilt, 2) reduce vertical descent velocity, 3) decrease lateral distance to the target, and 4) shorten overall flight duration. Terminal bonuses heavily reward 1) successful soft landings, 2) fuel-efficient touchdowns, and 3) near-target hard landings. Penalties apply for boundary violations, 1) including excessive tilt, 2) large lateral error, or 3) high impact speed.

The full set of reward components and their weights is summarized in Appendix III.

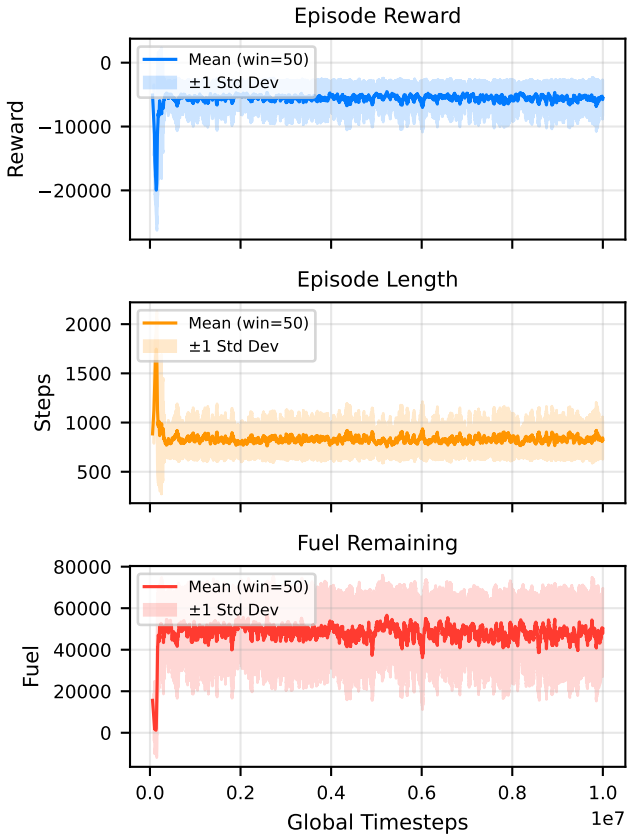

Figure 3: Training performance of the SAC policy over time.

B. Curriculum Learning

Curriculum learning is implemented by progressively tightening the success criteria. Early stages permit large landing radius, higher descent velocities, and larger tilt angles; these thresholds are gradually reduced once the policy achieves an 80% success rate at each level. This staged progression improves convergence and mitigates local-minimum failures.

C. Domain Randomization

Domain randomization is applied by varying the initial conditions of each training episode, exposing the policy to diverse poses, velocities, and tilts. This improves robustness and generalization to unseen states. All randomization is handled within the *RocketGym* reset() function.

D. Algorithmic Comparison

All three algorithms, : DQN, PPO, SAC, are implemented in *stable-baselines3* with vectorized environments (see section A). The goal is to identify which algorithm best handles 6-DoF lander dynamics, with the most stable training, highest success rate, and best fuel-efficient guidance under identical environment and rewards.

V. Results

A. Overall Performance

SAC achieved the highest success rate and most consistent soft landings, and it is also the one that converges to the optimal solution the fastest. PPO converged to stable after over 10 million total timesteps, but to suboptimal thrusting behavior as the reward still does not stablize. And there is catastrogic forgetting issues at the early stage of the training. DQN fails to produce a reliable 6-DoF controller, likely due to coarse action discretization, finer action grids may improve performance in future work.

B. Single Episode Rollout RL vs PI

The SAC policy produces smooth, coordinated 6-DoF trajectories, achieving stable glideslopes and controlled terminal descent. In contrast, the PID controller exhibits strong cross-coupling effects, overshoot, and late-stage instability—largely due to mass depletion and nonlinear attitude dynamics. SAC implicitly adapts to these variations, whereas fixed-gain PI control fails to regulate tilt and velocity simultaneously, leading to oscillatory or divergent behavior near touchdown.

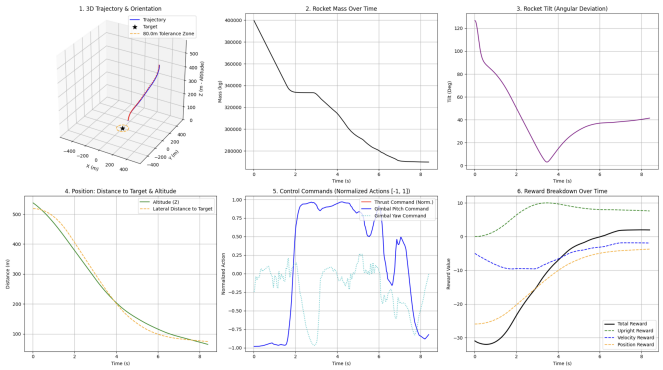

Figure 4: (a) Testing results for a random episode using the SAC RL controller.

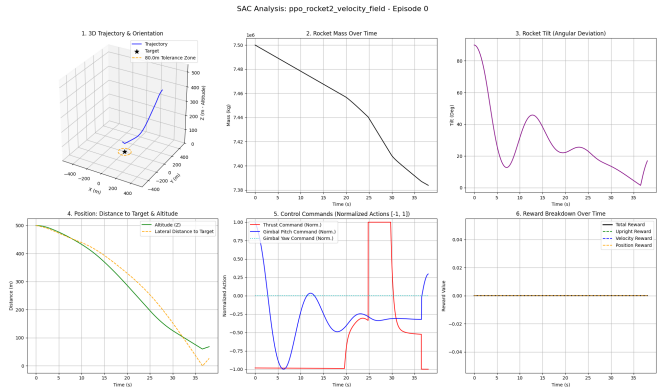

Figure 5: (b) Testing results for a random episode using the PID controller.

C. Monte Carlo Touchdown Statistics

We evaluate policy robustness over 100 randomized initial conditions.

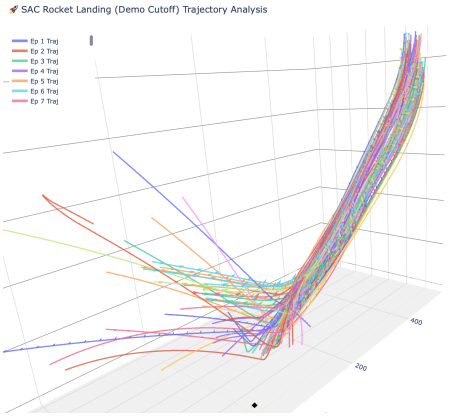

Figure 6: Monte Carlo landing trajectories over 100 runs.

The trajectory bundle exhibits consistent approach geometry, with most runs converging to a near-vertical glideslope before touchdown. Failure analysis shows that 43% of episodes terminate due to excessive tilt, 27% due to lateral drift, 19% from high terminal velocity, and 11% from timeouts. These results indicate that the SAC policy reliably handles the global descent and flip maneuvers but struggles with late-phase attitude stabilization. Improving terminal shaping or integrating a model-based terminal controller may substantially increase success rates.

In summary, while the policy shows strong global guidance behavior, limited terminal stability remains the primary source of failure. Future work will focus on improving late-stage attitude and velocity handling through refined shaping, adaptive gains, or a model-based terminal landing module.

VI. Conclusion

We presented a DRL-based IGC system for 6-DoF planetary powered descent. Using SAC with curriculum learning, domain randomization, and a high-fidelity variable-mass dynamics model, the policy learns to regulate the coupled translational and rotational dynamics of a single-gimbaled lander. These results provide a proof of concept for applying learning-based controllers in aerospace guidance and control. Despite strong global performance, the dominant failures arise in the terminal phase, where attitude stabilization and velocity regulation remain difficult. Future work will integrate physics-model priors and residual RL to improve training stability, inject model-based structure into the policy, and further increase touchdown reliability.

References

[1] S. Tassa, T. Erez, and E. Todorov, "Synthesis and stabilization of complex behaviors through online trajectory optimization," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 4906–4913, 2012.

[2] S. Tassa, N. Mansard, and E. Todorov, "Control-limited differential dynamic programming," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1168–1175, 2014.

[3] J. Neunert, M. Giftthaler, C. Semini, and J. Buchli, "Fast nonlinear model predictive control for unified trajectory optimization and tracking," in *IEEE Int. Conf. on Robotics and Automation (ICRA)*, pp. 1398–1404, 2016.
[4] M. Giftthaler, M. Neunert, M. Stäuble, M. Frigerio, C. Semini, and J. Buchli, "A family of iterative gauss–newton shooting methods for nonlinear optimal control," *The International Journal of Robotics Research*, vol. 38, no. 2-3, pp. 324–348, 2018.
[5] B. Açıkmeşe and J. M. Ploen, "Convex programming approach to powered descent guidance for mars landing," *Journal of Guidance, Control, and Dynamics*, vol. 30, no. 5, pp. 1353–1366, 2007.
[6] M. Szmuk, T. P. Reynolds, and B. Açıkmeşe, "Successive convexification for real-time 6-dof powered descent guidance with state-triggered constraints," in *AIAA Scitech 2019 Forum*, 2019.
[7] A. T. L., "Landing starships: Open-source reinforcement learning for 6-dof rocket landing." https://github.com/alxndrTL/Landing-Starships, 2024.
[8] A. Pellerin, "Starship landing gym: A 6-dof reinforcement learning environment for rocket landing." https://github.com/Armandpl/starship-landing-gym, 2024.
[9] A. Authors, "Deep reinforcement learning for 6-dof rocket landing with thrust-vector control," *arXiv preprint arXiv:2407.15083*, 2024.

APPENDIX

ROCKET MODEL PARAMETERS

The rocket engine is sized to achieve a thrust-to-weight ratio (TWR) of 3 on the lunar surface, providing sufficient authority for descent, flip maneuvers, and terminal landing burns. A vacuum specific impulse of $I_{sp} = 400\,\mathrm{s}$ is selected, representative of a high-performance hydrolox engine operating in near-vacuum conditions. With a wet mass of $5{,}000$ tons and a dry-mass fraction of 10%, the propellant load is $4{,}500$ tons. The maximum thrust, mass flow rate, and burn duration follow from $F = I_{sp}\dot{m}g_0$ and $t_{\text{burn}} = m_{\text{prop}}/\dot{m}$. The engine is mounted 30 m below the center of mass for attitude authority, and the vehicle height is approximately 100 m.

Table I: Rocket Physical and Propulsion Parameters (Lunar Environment)

Parameter	Value
Total wet mass m_0	5.00×10^6 kg
Dry mass fraction	10%
Dry mass m_{dry}	5.00×10^5 kg
Propellant mass m_{prop}	4.50×10^6 kg
Specific impulse (vacuum) I_{sp}	400 s
Standard gravity g_0	9.81 m/s^2
Lunar gravity g_{moon}	1.62 m/s^2
Chosen TWR (Moon)	3
Maximum thrust $F_{\max}$	2.43×10^7 N
Mass flow rate $\dot{m}$	6.20×10^3 kg/s
Burn time t_{burn}	7.26×10^2 s
Rocket height	100 m
Engine offset from CoM	30 m

HYPERPARAMETERS

Table II summarizes the main training hyperparameters used for SAC in *RocketGym*. These settings follow standard stable-baselines3 defaults with minor tuning for stability and runtime.

Table II: Training Hyperparameters for SAC

Parameter	Value
Algorithm	SAC (Stable-Baselines3)
Policy network	[256, 256] MLP
Learning rate	3×10^{-4}
Batch size	256
Replay buffer	1×10^6 transitions
Discount factor γ	0.95
Target smoothing τ	0.005
Entropy coefficient	auto
Parallel envs	16
Total steps	5M
Training time	1h 34m (M1 Max)

REWARD SHAPING

Table III lists the continuous shaping terms and terminal rewards used in *RocketGym*. These terms balance stabilization, fuel efficiency, and safety, and correspond directly to the implementation described in Sec. IV.

Table III: Reward Components and Weights Used in *RocketGym*

Reward Term	Equation
Upright alignment (encourages vertical posture)	$15\,\cos(\theta)$
Vertical velocity shaping (penalizes v_z)	$-1\,v_z$
Lateral distance penalty (reduces horizontal error)	$-0.05\,d_\perp$
Step penalty (discourages long episodes)	-0.01
Terminal Rewards	
Successful landing bonus + fuel preservation	$1000 + 5\left(\frac{m_{\text{fuel}}}{m_0}\right)$
Boundary violation penalty (tilt $> 100°$, $d_\perp > 700$ m, $\|\mathbf{v}\| > 500$ m/s)	-500
Crash penalty (non-boundary crash)	-100
Crash near target zone bonus ($d_\perp < 100$ m)	$-100 + 10$

TRAINING RESULTS FOR DQN AND PPO

Figures 7 and 8 show representative training curves for PPO and DQN, respectively. Both algorithms exhibit significantly slower convergence and higher variance compared to SAC, reflecting their limited sample efficiency for the 6-DoF landing task.

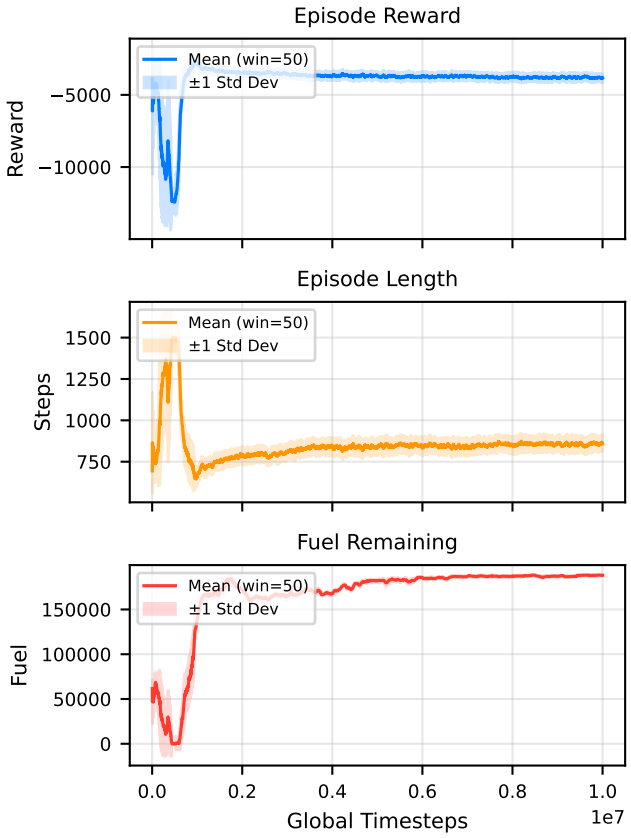

Figure 7: PPO training performance for a representative run.

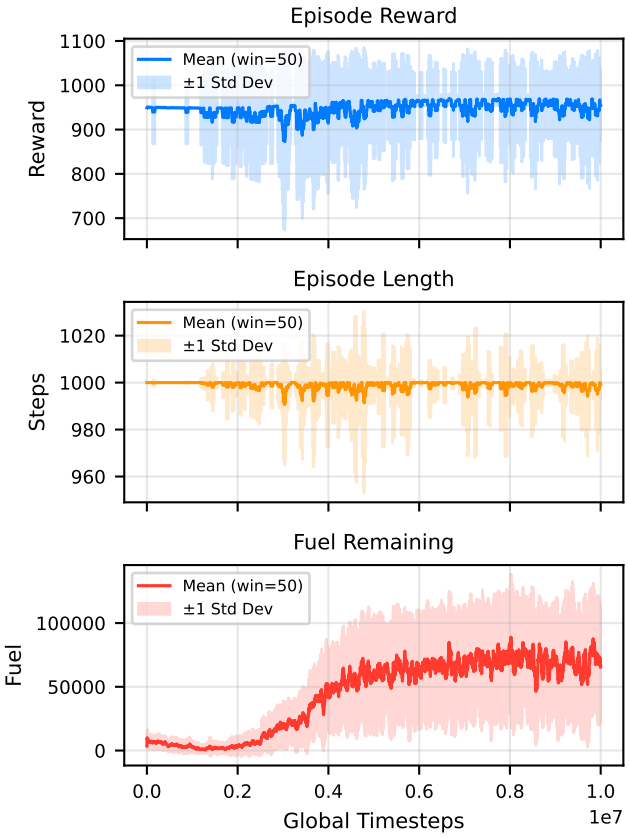

Figure 8: DQN training performance for a representative run.