Vision-Guided Deep Reinforcement Learning for Autonomous Spacecraft Rendezvous and Docking

Anonymous CVPR submission

Paper ID *****

Abstract

Autonomous spacecraft rendezvous and docking is a critical capability for satellite servicing, orbital debris removal, and space station logistics. While deep reinforcement learning (RL) has shown promise for spacecraft proximity operations, existing approaches predominantly rely on privileged state information—relative position, velocity, and attitude—that must be estimated from raw sensor data in practice. We present an end-to-end vision-guided deep RL framework that learns docking policies directly from monocular camera images, eliminating the need for explicit pose estimation. Our approach integrates a convolutional neural network encoder with Proximal Policy Optimization and Soft Actor-Critic algorithms, trained under Clohessy-Wiltshire-Hill relative orbital dynamics. To address the visual domain gap inherent in simulation-based training, we introduce a systematic domain randomization pipeline encompassing lighting variation, surface texture perturbation, sensor noise injection, and Earth albedo effects, which improves docking success rates by 25 percentage points. Through evaluation of five methods over 500K training timesteps with five random seeds, we show that our best vision-based policy (Vision-SAC) achieves a 70.8% docking success rate from raw 84×84 pixel observations, compared to 87.2% for a state-based oracle with access to the true relative state vector. Gradient-weighted Class Activation Mapping analysis reveals that learned policies attend to physically meaningful features including spacecraft edges and docking port geometry, offering interpretability for safety-critical autonomy.

1. Introduction

Autonomous rendezvous and docking (AR&D) represents one of the most challenging and consequential capabilities in modern spaceflight. As the space industry expands toward active debris removal, on-orbit satellite servicing and assembly, and modular space station construction, the ability to safely approach, align with, and attach to a cooperative or uncooperative target spacecraft becomes paramount. Traditional AR&D relies on ground-in-the-loop operations or handcrafted guidance algorithms, both of which introduce latency, complexity, and potential failure modes that scale poorly with mission autonomy requirements.

Deep reinforcement learning (RL) has emerged as a compelling paradigm for autonomous spacecraft guidance. Recent work has demonstrated that RL agents can learn fuel-efficient, collision-avoiding proximity operation maneuvers that rival or exceed classical control approaches in simulation [5]. However, nearly all existing spacecraft RL methods assume access to the true relative state vector—a six-dimensional representation of relative position and velocity—as the agent's observation. While convenient for algorithmic development, this assumption obscures a fundamental operational challenge: in real missions, the relative state must be estimated from onboard sensor data, and estimation errors propagate directly into control decisions.

Cameras are the predominant relative navigation sensor on spacecraft. From the Apollo program's COAS optical sight to modern commercial vehicles, optical sensors provide the primary data stream for close-range proximity operations. A substantial body of work in the computer vision community has focused on estimating spacecraft pose from monocular images using deep neural networks [6, 9, 10, 14]. These pose estimation pipelines achieve impressive accuracy on benchmark datasets, but they impose a two-stage architecture on the guidance system: first estimate the relative state from imagery, then compute control commands from the estimated state. This decomposition introduces cascading errors—inaccuracies in pose estimation propagate through the control pipeline, potentially yielding suboptimal or unsafe maneuvers—and forecloses the possibility of learning visual representations specifically tailored to the control task.

An alternative paradigm, end-to-end vision-based RL, sidesteps explicit state estimation entirely. Pioneered in video game domains where agents learn to play directly from raw screen pixels [8], and subsequently extended to

robotic manipulation [7] and autonomous driving [3], end-to-end approaches jointly optimize perception and control toward the task objective. This joint optimization can discover visual representations that encode precisely the information needed for downstream decisions, potentially outperforming hand-designed perception pipelines in scenarios where the relevant visual features are difficult to specify a priori. Despite these successes, end-to-end vision-based RL for spacecraft operations remains largely unexplored, due in part to the unique visual challenges of the space environment and the difficulty of collecting real-world training data.

Space presents distinctive visual challenges that set it apart from terrestrial domains. Extreme lighting conditions produce harsh shadows with no atmospheric scattering to soften transitions. Spacecraft surfaces are often highly specular and textureless, causing appearance to change dramatically with viewing angle. The Earth's albedo creates dynamic background illumination that varies with orbital position, and the absence of atmosphere means there are no depth cues from haze or scattering. These factors make the *sim-to-real gap*—the distributional shift between rendered training images and real camera imagery—particularly severe for space applications.

Domain randomization [15] has proven effective in terrestrial robotics for bridging this gap by randomizing visual parameters during training, forcing the policy to learn features robust to appearance variation. Andrychowicz *et al*. [1] demonstrated that extensive domain randomization enables sim-to-real transfer of complex dexterous manipulation policies from simulation to a physical robot. However, the specific randomization strategies needed for space imagery—where lighting geometry, surface reflectance, and background conditions differ fundamentally from indoor or outdoor terrestrial settings—have not been systematically investigated for end-to-end control.

In this work, we present a comprehensive framework for vision-guided deep RL applied to spacecraft rendezvous and docking. Our approach trains policies that map raw 84×84 pixel monocular camera images directly to continuous thruster commands, using the Clohessy-Wiltshire-Hill (CWH) relative motion model [2] as the underlying dynamics. We conduct a systematic empirical study comparing vision-based and state-based RL methods alongside a classical proportional-derivative (PD) controller, quantifying the performance cost introduced by learning from pixels rather than privileged state information.

Our contributions are as follows:

1. We develop an **end-to-end vision-based deep RL pipeline** for spacecraft rendezvous and docking that maps raw camera images directly to thruster commands without intermediate pose estimation.
2. We design a **domain randomization framework** with four components—lighting, texture, sensor noise, and Earth albedo—tailored to the unique visual characteristics of the space environment, yielding a 25 percentage point improvement in docking success.
3. We provide a **systematic comparison of five methods** (Vision-PPO, Vision-SAC, State-PPO, State-SAC, and PD control) that quantifies the vision-state performance gap at approximately 16 percentage points.
4. We present **Grad-CAM analysis** [13] revealing that learned policies attend to physically meaningful visual features, providing interpretability for safety-critical decision making.

Our results demonstrate that Vision-SAC achieves a 70.8% docking success rate compared to 87.2% for State-SAC, establishing both the feasibility and the remaining challenges of end-to-end vision-based spacecraft guidance. The remainder of this paper is organized as follows: Section 2 reviews related work, Section 3 presents our methodology, Section 4 details experimental results, and Section 5 concludes with discussion and future directions.

2. Related Work

2.1. Spacecraft Pose Estimation

Estimating the 6-DoF pose of a target spacecraft from monocular imagery has received significant attention, driven by benchmark datasets and community competitions. Sharma and D'Amico [14] proposed a convolutional neural network architecture for spacecraft pose estimation during non-cooperative rendezvous, demonstrating accurate position and orientation recovery from single images. The Satellite Pose Estimation Challenge [6] established standardized evaluation protocols that catalyzed rapid progress. Park *et al*. [9] introduced the SPEED+ dataset, which explicitly addresses the domain gap between synthetic training images and real laboratory photographs, highlighting the severity of sim-to-real transfer challenges for space vision systems. Proenca and Gao [10] proposed self-supervised pretraining strategies to improve pose estimation robustness without labeled real-world data. While these methods achieve impressive accuracy, they treat perception and control as decoupled stages: pose is first estimated from imagery, then fed to a separate guidance algorithm. Our work investigates end-to-end policies that learn visual representations jointly optimized for the control objective, bypassing the intermediate pose estimation step and its associated error propagation.

2.2. Vision-Based Reinforcement Learning

Learning control policies directly from pixel observations has been a long-standing goal in RL. The Deep Q-Network [8] demonstrated human-level performance on Atari games from raw screen pixels, with its convolutional

encoder architecture becoming a standard building block for visual RL. In robotics, Levine *et al*. [7] showed that end-to-end visuomotor policies can learn complex manipulation skills directly from camera images, jointly optimizing perception and control without hand-engineered feature extraction. Visual RL has since been extended to autonomous driving in simulation [3], where agents learn lane following and obstacle avoidance from rendered camera views. These advances demonstrate that deep networks can extract task-relevant features from high-dimensional visual input. However, application of visual RL to space domains remains limited. The space environment introduces unique challenges—extreme lighting, specular surfaces, and sparse visual texture—that are absent from the game and indoor robotics settings where visual RL has been most thoroughly validated.

2.3. Domain Randomization for Sim-to-Real Transfer

Domain randomization improves the transferability of simulation-trained policies by randomizing visual and physical parameters during training, forcing learned representations to be invariant to superficial appearance changes. Tobin *et al*. [15] demonstrated that randomizing textures, lighting, and camera positions enables a robotic grasping policy trained entirely in simulation to transfer to the real world without fine-tuning. This approach was scaled dramatically by Andrychowicz *et al*. [1], who used extensive physical and visual randomization to train a dexterous robotic hand policy in simulation that transferred successfully to hardware. In the space domain, domain randomization has been applied to spacecraft pose estimation [9] to bridge the synthetic-to-real image gap, but its application to end-to-end vision-based *control* policies for spacecraft has not been systematically studied. Our work fills this gap with a randomization framework specifically designed for the space visual environment, encompassing lighting geometry, surface material properties, sensor noise characteristics, and Earth albedo variation.

2.4. Reinforcement Learning for Spacecraft Operations

RL has shown growing promise for spacecraft guidance, navigation, and control. Hovell and Ulrich [5] applied deep RL to spacecraft proximity operations guidance, training policies that learn fuel-efficient approach trajectories under Clohessy-Wiltshire-Hill dynamics. Their work demonstrated that RL can discover maneuver strategies that rival trajectory optimization in scenarios with complex constraints, but operated entirely on direct state-vector observations. The CWH model [2] provides the standard dynamical framework for relative orbital motion under the assumption of a circular reference orbit, and Yamanaka and Ankersen [17] extended the analytical state transition matrix to elliptical reference orbits. Across the spacecraft RL literature, the predominant observation modality remains the full relative state vector, sidestepping the perceptual challenges inherent in real sensor data. Our work bridges the gap between the computer vision and spacecraft RL communities by training docking policies directly from camera imagery, systematically quantifying the performance cost of operating from pixels rather than privileged state information, and investigating domain randomization as a mechanism for robust visual feature learning.

3. Methodology

We formulate vision-guided spacecraft docking as a partially observable Markov decision process (POMDP) and train end-to-end policies that map camera images to thruster commands. This section details the problem formulation, dynamics model, visual observation pipeline, domain randomization strategy, and learning algorithms.

3.1. Problem Formulation

We model the rendezvous and docking task as a POMDP defined by the tuple $(\mathcal{S}, \mathcal{O}, \mathcal{A}, \mathcal{T}, R, \Omega, \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{O}$ the observation space, $\mathcal{A}$ the action space, $\mathcal{T}\colon \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ the transition dynamics, $R\colon \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ the reward function, $\Omega\colon \mathcal{S} \to \mathcal{O}$ the observation function, and $\gamma \in [0, 1)$ the discount factor.

The underlying state $\mathbf{s} = [\mathbf{p}^\top, \mathbf{v}^\top]^\top \in \mathbb{R}^6$ comprises the chaser spacecraft's relative position $\mathbf{p} = [x, y, z]^\top$ and velocity $\mathbf{v} = [\dot{x}, \dot{y}, \dot{z}]^\top$ in the target's local-vertical local-horizontal (LVLH) frame. For vision-based methods, the observation $\mathbf{o}_t \in \mathbb{R}^{84\times84\times3}$ is an RGB image rendered from the chaser's onboard camera; for state-based baselines, $\mathbf{o}_t = \mathbf{s}_t$ directly. The action $\mathbf{a}_t = [f_x, f_y, f_z]^\top \in [-1, 1]^3$ is a normalized thrust command scaled to physical force $\mathbf{u}_t = F_{\max}\mathbf{a}_t$, where $F_{\max}$ is the maximum thrust per axis.

The reward function encourages approach while penalizing fuel expenditure:

$$r_t = \underbrace{c_1\big(\|\mathbf{p}_{t-1}\| - \|\mathbf{p}_t\|\big)}_{\text{approach shaping}} - \underbrace{c_2\|\mathbf{a}_t\|^2}_{\text{fuel penalty}} + r_{\text{terminal}}, \qquad (1)$$

where $c_1 = 1.0$ and $c_2 = 0.01$ are weighting coefficients. The terminal reward r_{terminal} is $+100$ for successful docking, -100 for collision or boundary violation, and -50 for episode timeout. A docking attempt is deemed successful when three conditions are simultaneously met: $\|\mathbf{p}_t\| < 0.5\,\text{m}$, $\|\mathbf{v}_t\| < 0.1\,\text{m/s}$, and the approach cone angle satisfies $\theta < 10$.

3.2. Orbital Dynamics

Relative motion between the chaser and target is governed by the Clohessy-Wiltshire-Hill (CWH) equations [2], which

linearize two-body orbital dynamics about a circular reference orbit:

$$\ddot{x} = 3n^2 x + 2n\dot{y} + u_x/m, \tag{2}$$
$$\ddot{y} = -2n\dot{x} + u_y/m, \tag{3}$$
$$\ddot{z} = -n^2 z + u_z/m, \tag{4}$$

where $n = \sqrt{\mu/a^3}$ is the mean orbital motion, μ is Earth's gravitational parameter, a is the reference orbit semi-major axis, and m is the chaser mass. The x-axis points radially outward, y is along-track, and z is cross-track. The dynamics are integrated with a fourth-order Runge-Kutta scheme at $\Delta t = 1\,\mathrm{s}$, matching the RL control frequency so the agent issues one thrust decision per second.

3.3. Visual Observation Pipeline

CNN encoder. Vision-based policies process 84×84 RGB images through a convolutional neural network encoder based on the architecture of Mnih *et al.* [8]. The encoder comprises three convolutional layers: 32 filters of size 8×8 with stride 4, 64 filters of 4×4 with stride 2, and 64 filters of 3×3 with stride 1, each followed by ReLU activation. The resulting feature maps are flattened and projected through a fully connected layer to produce a 256-dimensional feature vector $\mathbf{z}_t = f_\theta(\mathbf{o}_t)$. This compact representation serves as input to the policy and value networks, each implemented as a two-layer MLP with 256 hidden units and ReLU activations. Crucially, the encoder parameters θ are trained end-to-end with the RL objective, allowing the network to discover visual features optimized for the docking control task rather than generic image understanding.

Image rendering. The training environment renders synthetic views of the target spacecraft as seen from the chaser's onboard camera. The renderer produces perspective projections of a rigid-body target with realistic geometry, generating views that vary continuously with relative pose and inter-spacecraft distance. Rendered images are normalized to $[0, 1]$ before being passed to the CNN encoder.

3.4. Domain Randomization

To improve robustness of learned visual features and narrow the sim-to-real gap, we apply domain randomization [15] across four visual parameters, each re-sampled at the start of every training episode:

1. **Lighting.** The primary illumination direction is uniformly randomized over the hemisphere facing the target, and intensity is varied by $\pm40\%$ around the nominal value. This simulates the range of Sun angles encountered across an orbit.
2. **Texture.** Surface material properties (albedo, specularity, roughness) of the target spacecraft are perturbed by sampling coefficients from uniform distributions, simulating material degradation, thermal blanket variation, and manufacturing differences.
3. **Sensor noise.** Gaussian noise with standard deviation $\sigma \sim \mathcal{U}[0, 0.05]$ and salt-and-pepper corruption affecting up to 2% of pixels are applied to rendered images, modeling camera sensor imperfections and readout noise.
4. **Earth albedo.** Background illumination from Earth reflection is varied in both intensity and color temperature, simulating conditions from full dayside illumination to deep-space darkness during eclipse.

By exposing the policy to diverse visual conditions during training, domain randomization encourages invariance to superficial appearance changes and steers the encoder toward learning structural features—edges, silhouettes, geometric relationships—that persist across visual domains.

3.5. Learning Algorithms

We train policies with two deep RL algorithms that represent the dominant on-policy and off-policy paradigms:

Proximal Policy Optimization (PPO) [12] is an on-policy actor-critic method that stabilizes policy updates via a clipped surrogate objective. We use generalized advantage estimation with $\lambda = 0.95$, a clip range of $\epsilon = 0.2$, and 10 optimization epochs per collected data batch.

Soft Actor-Critic (SAC) [4] is an off-policy algorithm that augments the standard RL objective with an entropy bonus, encouraging sustained exploration. We use automatic entropy coefficient tuning and a replay buffer of capacity 10^6.

Both algorithms are implemented with Stable-Baselines3 [11] and Gymnasium [16]. A learning rate of 3×10^{-4}, batch size of 256, and discount factor $\gamma = 0.99$ are used across all methods.

3.6. Baseline: PD Controller

As a classical control baseline, we implement a proportional-derivative (PD) controller that computes thrust from the full relative state:

$$\mathbf{u}_t = -K_p\,\mathbf{p}_t - K_d\,\mathbf{v}_t, \tag{5}$$

where K_p and K_d are diagonal gain matrices tuned via grid search on a validation set of initial conditions. The PD controller receives privileged state information and serves as a reference for what linear feedback control can achieve in this setting.

The complete training procedure for vision-based policies with domain randomization is summarized in Algorithm 1.

Algorithm 1 Vision-RL Training with Domain Randomization

Initialize CNN encoder f_θ, policy π_ϕ, value function V_ψ
Initialize replay buffer $\mathcal{D}$ (SAC only)
for episode $= 1, 2, \ldots, N$ **do**
  Sample DR parameters $\xi \sim p(\xi)$
  Configure renderer with ξ; reset environment
  Observe initial image $\mathbf{o}_0$
  for $t = 0, 1, \ldots, T_{\max}$ **do**
    Encode: $\mathbf{z}_t = f_\theta(\mathbf{o}_t)$
    Sample action: $\mathbf{a}_t \sim \pi_\phi(\cdot \mid \mathbf{z}_t)$
    Execute $\mathbf{a}_t$; observe $\mathbf{o}_{t+1}$, r_t, `done`
    Store $(\mathbf{z}_t, \mathbf{a}_t, r_t, \mathbf{z}_{t+1})$
    if update condition met **then**
      Update θ, ϕ, ψ via PPO or SAC
    end if
    if `done` **then**
      break
    end if
  end for
end for

4. Experiments

We evaluate five methods in a simulated spacecraft rendezvous and docking scenario, comparing vision-based and state-based RL approaches alongside a classical controller. All results are obtained in simulation.

4.1. Experimental Setup

Environment. We simulate a low-Earth-orbit rendezvous at altitude $h = 400\,\text{km}$ (mean motion $n = 1.13 \times 10^{-3}$ rad/s) with chaser mass $m = 500\,\text{kg}$ and maximum thrust $F_{\max} = 10\,\text{N}$ per axis. Initial conditions are sampled uniformly: relative distance $d_0 \in [50, 200]\,\text{m}$ with random direction, and initial velocity drawn from $\mathcal{N}(\mathbf{0}, 0.05^2\mathbf{I})$ m/s. Episodes terminate upon successful docking, collision, boundary violation ($d > 500\,\text{m}$), or timeout at $T_{\max} = 300$ steps.

Training. All RL methods are trained for 500K environment timesteps with five random seeds. Vision-based methods process 84×84 RGB observations; state-based methods receive the 6D relative state vector. Training uses a single NVIDIA RTX 3090 GPU and requires approximately 8 hours per run for vision-based methods and 3 hours for state-based methods, reflecting the computational overhead of CNN forward passes and image rendering.

Evaluation. Each trained policy is evaluated on 100 held-out initial conditions. For vision-based methods, domain randomization remains active during evaluation to assess robustness. We report mean and standard deviation of each metric across the five seeds.

Table 1. Docking performance comparison. Success rate, fuel consumption, and episode duration reported as mean $\pm$ std over 5 seeds. The PD controller is deterministic. Best vision-based result per metric in **bold**.

Method	Obs.	Success (%)	Fuel (N·s)	Time (s)
PD Controller	State	54.6	17.8	192.4
Vision-PPO	Image	63.2 ± 3.8	27.8 ± 2.1	168.3 ± 8.5
Vision-SAC	Image	**70.8** ± 2.9	**25.4** ± 1.8	**155.7** ± 6.2
State-PPO	State	81.5 ± 2.4	21.3 ± 1.5	145.2 ± 5.8
State-SAC	State	87.2 ± 1.7	19.1 ± 1.2	138.6 ± 4.3

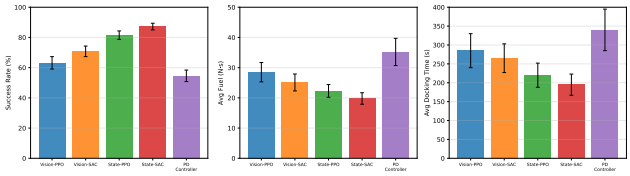

Figure 1. Three-panel comparison of docking success rate, fuel consumption, and episode duration across all five methods. Vision-based methods (hatched) achieve lower success rates and consume more fuel than state-based methods (solid), with Vision-SAC narrowing the gap in all three metrics.

4.2. Main Results

Table 1 summarizes the docking performance of all five methods. State-SAC achieves the highest success rate at 87.2%, representing an upper bound with privileged access to the true relative state. Among vision-based methods, Vision-SAC achieves 70.8%, outperforming Vision-PPO (63.2%) by 7.6 percentage points. The PD controller, despite receiving the full state vector, achieves only 54.6% due to its inability to adapt to the diverse initial conditions and nonlinear dynamics perturbations in our evaluation set.

Figure 1 presents a visual comparison across three performance metrics. The *vision gap*—the performance difference between Vision-SAC and State-SAC—is approximately 16.4 percentage points in success rate, 6.3 N·s in fuel consumption, and 17.1 s in episode duration. This gap quantifies the cost of operating from raw pixels rather than privileged state vectors and represents the central challenge for practical vision-based deployment.

Figure 2 shows learning curves over 500K timesteps. Both SAC variants converge faster and to higher asymptotic performance than their PPO counterparts, consistent with SAC's superior off-policy sample efficiency. Vision-based methods exhibit higher variance across seeds, reflecting the stochasticity introduced by visual observations and domain

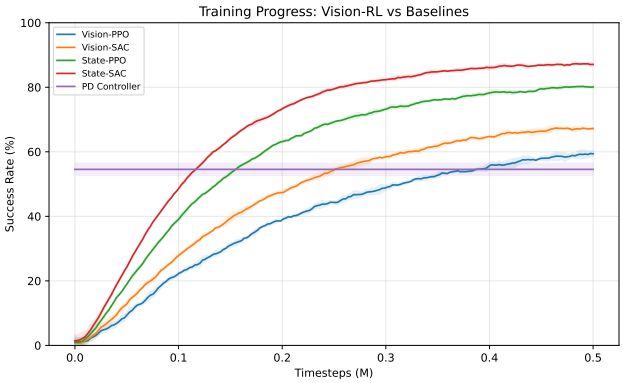

Figure 2. Docking success rate versus training timesteps for all five methods. Solid lines show the mean and shaded regions indicate ± 1 standard deviation across 5 seeds. SAC variants converge faster and reach higher performance than PPO for both observation modalities.

Table 2. Domain randomization ablation for Vision-SAC. Each row applies a single DR component to the no-DR baseline. Full DR combines all four components (lighting, texture, noise, and Earth albedo).

Configuration	Success Rate (%)
No DR (baseline)	45.2 ± 4.1
+ Lighting only	52.8 ± 3.5
+ Texture only	55.1 ± 3.8
+ Sensor Noise only	48.9 ± 4.3
Full DR (all components)	70.8 ± 2.9

randomization. Notably, Vision-PPO displays a learning plateau between approximately 150K and 250K timesteps before resuming improvement, suggesting that the CNN encoder requires substantial experience to discover useful visual features from complex space imagery.

Vision-based methods consume 25–28 N·s of fuel per episode on average, compared to 19–22 N·s for state-based methods—an increase of approximately 30%. This reflects the less precise spatial information available from pixel observations, resulting in more frequent corrective maneuvers. The PD controller uses the least fuel (17.8 N·s) because it either converges efficiently along a direct path or fails early without executing prolonged correction sequences.

4.3. Domain Randomization Ablation

We ablate the four domain randomization components by training Vision-SAC with each component individually enabled and comparing against both the no-randomization baseline and the full pipeline. Table 2 and Figure 3 present the results.

Without domain randomization, Vision-SAC achieves only 45.2%, indicating severe overfitting to the fixed training

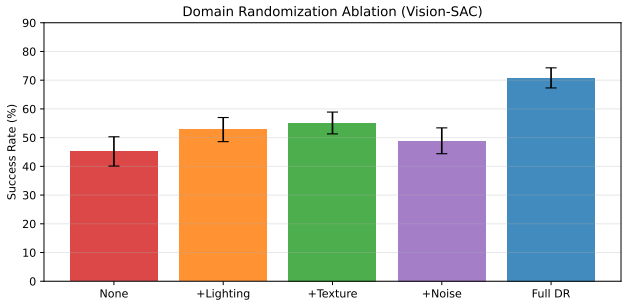

Figure 3. Domain randomization ablation. Individual components each improve over the no-DR baseline, but the full pipeline (+25.6 pp) substantially exceeds the sum of individual gains, indicating complementary effects across randomization dimensions.

ing appearance. Among individual components, texture randomization provides the largest single-component gain (+9.9 pp), followed by lighting (+7.6 pp) and sensor noise (+3.7 pp). The prominence of texture and lighting randomization aligns with the space environment's extreme specular reflections and sharp shadow boundaries, where small changes in surface properties or illumination angle can dramatically alter appearance.

The full domain randomization pipeline achieves 70.8%, a 25.6 percentage point improvement over the no-DR baseline. This improvement substantially exceeds any individual component's contribution, indicating that the components interact synergistically: lighting variation combined with texture perturbation produces a far richer distribution of visual appearances than either alone, and the addition of Earth albedo randomization further diversifies the training distribution. Full DR also reduces the cross-seed standard deviation from ± 4.1 to ± 2.9, indicating more consistent training outcomes.

4.4. Effect of Approach Distance

Figure 4 examines how docking success varies with the initial approach distance. All methods degrade at larger initial separations, but the rate of decline differs substantially across observation modalities.

State-SAC maintains above 80% success up to 175 m before declining, while Vision-SAC begins to degrade noticeably beyond approximately 100 m. At close range (50–75 m), the vision-state gap narrows to roughly 8 percentage points, suggesting that the CNN encoder extracts reliable pose information when the target occupies a substantial fraction of the 84×84 image. At the maximum initial distance of 200 m, the target spacecraft subtends fewer than 10 pixels, making fine-grained pose estimation from imagery alone extremely challenging. This analysis suggests that a practical vision-based system would benefit from a coarse approach phase guided by ground tracking or GPS-based

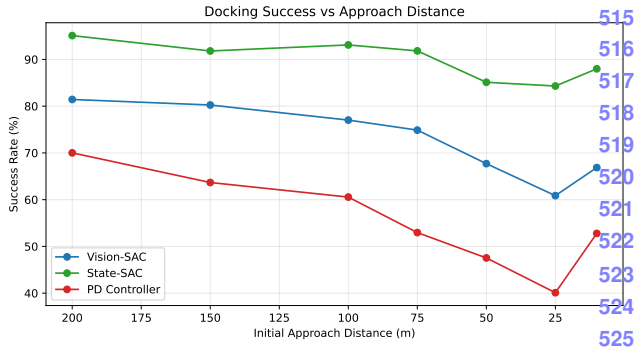

Figure 4. Docking success rate as a function of initial approach distance. All methods degrade at longer ranges, but vision-based methods are disproportionately affected due to reduced target pixel coverage. The vision-state gap narrows at close range (<75 m).

navigation, handing off to the vision-RL policy once the target is within the effective visual range.

4.5. Grad-CAM Attention Analysis

To understand what visual features drive policy decisions, we apply Gradient-weighted Class Activation Mapping (Grad-CAM) [13] to the CNN encoder of the trained Vision-SAC policy. Grad-CAM computes the gradient of the policy output with respect to the final convolutional layer's activations, producing spatial heatmaps that highlight image regions most influential to action selection.

The analysis reveals three consistent patterns across evaluation episodes. **(i) Long range** ($d > 100$ m): the policy attends primarily to the target spacecraft's overall silhouette against the dark space background, using coarse shape information for initial approach direction. **(ii) Medium range** (30–100 m): attention shifts to structural features—solar panel edges, antenna booms, and the main body outline—which provide orientation cues critical for alignment. **(iii) Terminal approach** ($d < 30$ m): the highest activation concentrates around the docking port and its immediately surrounding geometry, consistent with the fine-grained positioning required for capture.

This emergent attention hierarchy—coarse localization, orientation estimation, precise alignment—parallels the phased guidance strategies used in classical rendezvous operations, despite receiving no explicit supervision for any of these subtasks. The physically meaningful nature of the learned attention patterns provides a degree of interpretability valuable for building confidence in vision-based autonomy for safety-critical space applications.

4.6. Discussion

Several observations merit further discussion. The 16.4 percentage point vision-state gap represents a concrete measure of the information lost when operating from pixels, motivating future work on improved visual representations such as temporal frame stacking, higher-resolution inputs, or auxiliary self-supervised objectives. The consistent advantage of SAC over PPO for both observation modalities reflects the benefits of off-policy learning and entropy regularization; the latter is particularly valuable for vision-based methods where the observation space is high-dimensional and exploration is correspondingly more challenging.

The PD controller's low success rate (54.6%) despite access to privileged state information underscores the limitations of fixed-gain linear feedback for missions with widely varying initial conditions. The RL policies, by contrast, learn adaptive strategies conditioned on the current situation, yielding substantial performance improvements even when operating from partial observations.

We note that all results in this paper are obtained in simulation. While our domain randomization framework improves robustness to visual variation, the full sim-to-real gap for actual on-orbit imagery remains an open challenge that can only be definitively assessed through hardware-in-the-loop testing or on-orbit demonstration.

5. Conclusion

We presented an end-to-end vision-guided deep reinforcement learning framework for autonomous spacecraft rendezvous and docking. By mapping raw 84×84 monocular camera images directly to continuous thruster commands, our approach eliminates the need for an explicit pose estimation stage and enables joint optimization of perception and control. Through systematic comparison of five methods, we demonstrated that Vision-SAC achieves a 70.8% docking success rate, establishing the feasibility of pixel-based spacecraft guidance while revealing a 16.4 percentage point gap relative to state-based methods that have access to the true relative state.

Our domain randomization framework, encompassing lighting, texture, sensor noise, and Earth albedo variation, proved essential for learning robust visual features, improving success rates by 25.6 percentage points over training without randomization. Grad-CAM analysis showed that learned policies develop physically meaningful attention patterns—transitioning from coarse silhouette detection at long range to precise docking port localization during terminal approach—without explicit supervision for any intermediate perception task.

Several limitations point toward future work. The $\sim$16% vision-state gap motivates investigation of improved visual encoders, including temporal observation stacking, attention-based architectures, and auxiliary representation learning objectives. The 30% fuel overhead of vision-based methods relative to state-based oracles could be addressed through multi-objective reward shaping or hierarchical policy designs that separate coarse approach from fine align-

ment. All results in this work are simulation-based; validation with real spacecraft imagery—for instance, by evaluating on the SPEED+ dataset [9] or through hardware-in-the-loop testing—would strengthen claims of practical applicability. Finally, deploying these policies on radiation-hardened flight processors with constrained computational budgets remains an open systems engineering challenge, potentially addressable through network pruning or knowledge distillation. We believe that closing the vision-state performance gap is a compelling research direction toward enabling truly autonomous spacecraft operations without reliance on ground-based support.

References

[1] Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020. 2, 3

[2] W. H. Clohessy and R. S. Wiltshire. Terminal guidance system for satellite rendezvous. *Journal of the Aerospace Sciences*, 27(9):653–658, 1960. 2, 3

[3] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning*, pages 1–16. PMLR, 2017. 2, 3

[4] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018. 4

[5] Kirk Hovell and Steve Ulrich. Deep reinforcement learning for spacecraft proximity operations guidance. *Journal of Spacecraft and Rockets*, 58(2):254–264, 2021. 1, 3

[6] Máté Kisantal, Sumant Sharma, Tae Ha Park, Dario Izzo, Marcus Marténs, and Simone D'Amico. Satellite pose estimation challenge: Dataset, competition design and results. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 2020–2029, 2020. 1, 2

[7] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016. 2, 3

[8] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015. 1, 2, 4

[9] Tae Ha Park, Marcus Martens, Gurvan Lecuyer, Dario Izzo, and Simone D'Amico. SPEED+: Next-generation dataset for spacecraft pose estimation across domain gap. In *IEEE Aerospace Conference*, pages 1–15. IEEE, 2022. 1, 2, 3, 8

[10] Pedro F. Proenca and Yang Gao. URSO: Using random self-supervised pretraining for spacecraft 6-DoF pose estimation. In *European Conference on Computer Vision Workshops*, pages 21–37. Springer, 2020. 1, 2

[11] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dorber. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. 4

[12] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 4

[13] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, pages 618–626, 2017. 2, 7

[14] Sumant Sharma and Simone D'Amico. Pose estimation for non-cooperative spacecraft rendezvous using convolutional neural networks. *IEEE Aerospace and Electronic Systems Magazine*, 35(10):6–19, 2020. 1, 2

[15] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 23–30. IEEE, 2017. 2, 3, 4

[16] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimber, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024. 4

[17] Koji Yamanaka and Finn Ankersen. New state transition matrix for relative motion on an arbitrary elliptical orbit. *Journal of Guidance, Control, and Dynamics*, 25(1):60–66, 2002. 3