Reinforcement Learning for Space-to-Space Surveillance: Autonomous Scheduling for Resident Space Object Imaging

Daniel Huterer Prats
PhD Student, Department of Aerospace Engineering Sciences, University of Colorado Boulder, 3775 Discovery Dr., Boulder, CO 80303
Hanspeter Schaub
Professor, Department of Aerospace Engineering Sciences, University of Colorado Boulder

Chris Wheeler
Chief Technology Officer, Interactive Aptitude

ABSTRACT

The increasing number of resident space objects (RSO) in low Earth orbit poses significant challenges for autonomous Space Situational Awareness (SSA). Unlike Earth observation, space-based SSA requires agile imaging of fast-moving targets under stringent constraints on power, line-of-sight, and illumination. This work researches having a satellite taking images of space objects with known trajectories. The paper formulates the space-to-space RSO inspection problem as a partially observable Markov decision process and trains a reinforcement learning (RL) agent for simulatenous dynamic target selection and onboard resource management. Using the BSK-RL environment and the Basilisk high-fidelity spacecraft simulator, an actor-critic RL agent learns to autonomously image RSOs while maximizing coverage and adhering to subsystem limitations. Results show that the learned policy generalizes across orbital configurations, exploits eclipse periods for proactive downlinking, maintains energy margins quasi-autonomously, and achieves more timely and useful image delivery than a myopic heuristic. These findings support the potential of RL-enabled autonomy for future scalable SSA missions.

1. INTRODUCTION

The rapid growth in resident space objects (RSOs), fueled by the deployment of numerous low Earth orbit (LEO) constellations and the reduced barrier to space access, has led to a surge in cataloged space assets. Estimates indicate over 9700 active satellites currently orbiting in LEO, with projections of tens of thousands more in the next decade [1, 2]. This surge is creating significant strain on existing space domain awareness (SDA) capabilities and necessitating further advancements to keep up with the ever-growing number of space assets. While for decades satellite tracking and imageing has been predominantly conducted from ground-based telescopes, they are subject to various challenges such as weather, atmospheric effects, fixed field-of-regard (FOR) and limited to observations only during nighttime. These limitations have significant impact and in some cases only allow for 25% operability in some instances[3]. Conversely, space-based sensors for satellite tracking and imaging are not impacted by most of these and additionally, over the course of the orbit, possess an effective FOR covering the entire sky [4].

Ongoing research in the broader field of space-based space surveillance (SBSS) also addresses the challenging problem of orbit determination using short-arc optical measurements from narrow field-of-view (FoV) sensors. These measurements often span only a few seconds due to the high relative velocities between sensor and target, leading to significant uncertainty in orbital state estimation [5, 6, 7]. To mitigate this, recent work has explored advanced estimation techniques including batch and sequential estimators, genetic algorithms, and multiple shooting methods [8, 9, 10, 11].

If a space object has a well-estimated orbit then space-to-space imaging of this Resident Space Object (RSO) is possible. The observing satellite must carefully track the RSO object during the imaging to compensate for the fast relative motion. Scheduling such imaging tasks is challenging as the Earth orbits are in a broad range of orbit planes

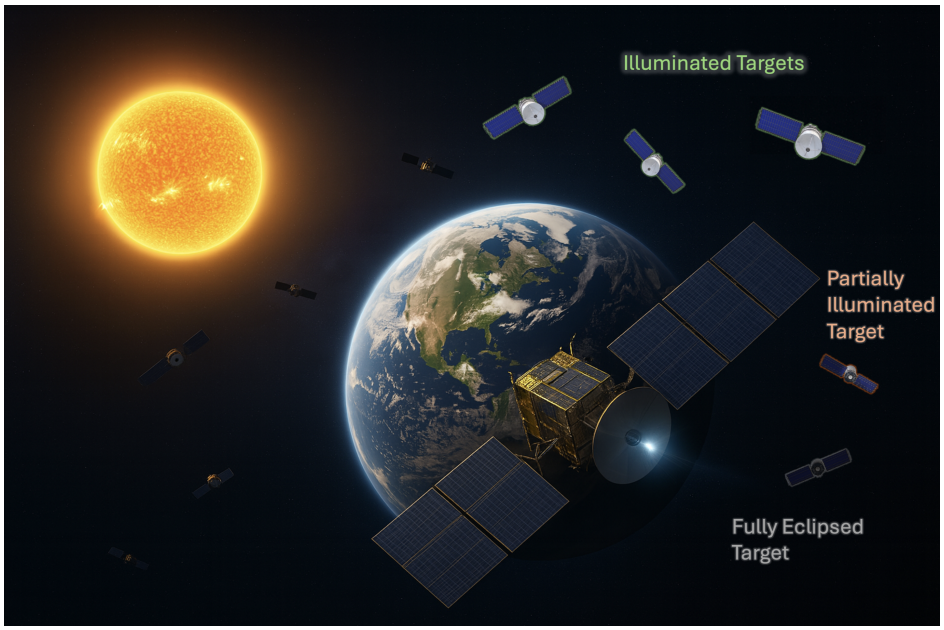

Fig. 1: Space-based RSO imaging under lighting constraints

and altitudes. The focus of this paper is developing autonomous on-board space-to-space imaging tasks using a shielded neural network to ensure satellite safety.

Scheduling the tasking of sensors for SSA and SBSS has been shown to be NP-hard, similar to the vehicle routing problem with time windows [12]. As such, exact optimization approaches, including mixed-integer programming and constraint satisfaction formulations [13], can only guarantee optimality for small instances. To address scalability, a variety of non-RL heuristic and metaheuristic approaches have been explored, including greedy algorithms [14], multi-objective genetic algorithms (NSGA-II) [15], and information-gain–driven schedulers [16], many of which have been applied to both ground- and space-based SSA contexts. While these methods can produce high-quality schedules, they typically require rerunning the optimization whenever new target opportunities or updated orbital data become available, limiting their responsiveness in highly dynamic environments.

In contrast, reinforcement learning (RL) can learn reusable policies that generalize across scenarios, enabling rapid real-time decision-making without solving a new optimization problem from scratch at each step[17]. RL methods can also directly incorporate complex operational constraints, stochastic events, and multi-objective trade-offs into the decision process, avoiding the need for simplifications often required in MILP or heuristic formulations. Hence, some research has also been conducted in optimizing the task scheduling of ground-based sensors using RL [18, 19, 20, 21, 22]. Nonetheless, the use of RL applied to space situational awareness is still in its early steps, particularly when applied to space-based sensor tasking [23]. The many advantages of space-based sensor tasking have been recognized for decades [24, 25, 26], yet its operational environment, characterized by rapidly changing fields-of-regard and dynamic target geometry, poses significantly greater scheduling challenges than for ground-based sensors with fixed FOR. These complexities make it an especially compelling domain for RL-based approaches, which can adapt to such variability in real time.

While RL has gained substantial traction in various space applications, previous research investigated the use of RL for onboard (space-based) sensor tasking focusing primarily on Earth observation satellites (EOS) [27, 28, 29, 30, 31, 32]. These missions typically involve scheduling imagery of static or slowly moving ground targets, and RL agents have been trained to manage complex trade-offs such as momentum buildup, limited power availability, and onboard data storage [33, 34]. In these scenarios, the relative motion between the spacecraft and the targets is predictable and constrained to a single hemisphere (only looking at targets below), simplifying both access modeling and decision-making.

In contrast, space-based sensor tasking for space surveillance presents a significantly different challenge as targets in

space are themselves orbiting rapidly and may occupy entirely different orbital planes, leading to highly dynamic and transient imaging opportunities. To date, only limited RL-based approaches have addressed this problem. Notably, [23] applied RL for a LEO-based sensor imaging GEO targets with the aim to minimize the mean trace covariance across all RSOs. However, this setup benefits from the quasi-stationary nature of GEO targets, which reduces the relative motion complexity. Additionally, that work focused solely on target selection and did not incorporate realistic spacecraft safety or operational constraints such as battery depletion, eclipse (illumination status of targets), reaction wheel desaturation, or the need for downlink operations.

The focus of this paper is to research the more demanding scenario of LEO-to-LEO space-based imaging, where both the sensor and the targets are in fast-moving and known orbits. The agent must reason over short-lived windows of opportunity, rapid changes in line-of-sight (LOS), and tighter timing constraints. Furthermore, this study incorporates a rich set of spacecraft operational constraints, including energy management across eclipse cycles, momentum buildup and desaturation needs, and data storage limitations. The resulting on-board network is very responsive to a changing target environment as the tasking decisions are made at each decision interval with the latest current states. Short-notice injection of new high value targets can readily be addressed. The focus of this study is on the scheduling and onboard decision-making aspects of a space-based imaging system. The simulation framework assumes known target states and does not currently incorporate state uncertainty or simulate how those uncertainties might be reduced through the imaging actions taken by the agent (as is done in [23]). By using the Proximal Policy Optimization (PPO) algorithm, the policy does not make drastic changes in-between training steps and has the ability to learn in a stable fashion, with reduced risk of unlearning good behavior [35]. By tackling onboard scheduling with safety and operational constraints, this work helps to bridge the gap in the space-based SSA literature on deep RL for LEO-to-LEO imaging.

2. SPACE-BASED SPACE SURVEILLANCE ENVIRONMENT FORMULATION

Space-based SSA is inherently complex and cannot be accurately represented using static targets, as is commonly done in agile Earth observation satellite (AEOS) scenarios, which focus on fixed ground locations[36, 37, 38]. In contrast, a space-to-space scanning satellite must deal with the dynamic and hard-to-predict relative orbital motion of the RSOs in the catalog, especially with imperfect knowledge of their state. These targets move in and out of the field of view (FOV) from all directions due to their diverse inclinations and orbital regimes, ranging from low LEO to the edge of MEO in this study. The maneuver times to lock the camera bore-sight onto another RSO are not analytically predictable. As a result, it becomes significantly more difficult to anticipate which RSOs will be observable at any given time. This problem is further complicated when considering the fast and frequently changing lighting conditions, especially for LEO orbits. Consequently, the agent must dynamically update the list of visible unimaged targets at each decision step-a challenge that will be detailed further in the next section.

2.1 Simulation Environment

A custom SBSS environment is developed to simulate the space-to-space imaging task, built upon the high-fidelity Basilisk simulation package [39][1] and the BSK-RL framework [40] [2]. The simulation models a single imaging spacecraft positioned in a 500 km LEO tasked with observing a catalog of RSOs, focusing on imaging rather than orbit determination. The environment tracks the states of the spacecraft and RSOs, generates observations based on sensor parameters, and computes rewards based on successful imaging and downlink actions according to the agents reward function, while respecting operational constraints.

Each environment step corresponds to the spacecraft completing an action with a fixed duration, described in section 3.1. The RSOs are randomly initialized as described in Table 1 and it may downlink to any of the seven groundstations, detailed in Table 8 of the appendix. The slant ranges shown in Fig. 2 were computed as defined in [41] to give a visual representation of the visibility as seen from a 500 km LEO altitude.

Basilisk's modular, message-passing architecture allows for the assembly of a flight-like closed-loop simulation in which attitude, power, data handling, guidance, and actuation are explicitly coupled. Each block publishes and/or subscribes to typed messages (states, commands, health), so subsystem implementations can be swapped without changing interfaces. In our SBSS setup shown in Fig. 3, the `LocPointTask` is connected with the selected target's navigation message through the `imageRSO()` FSW Module. The computed attitude error and body-rate estimates are consumed

[1] `https://avslab.github.io/basilisk/`
[2] `https://avslab.github.io/bsk_rl/`

Table 1: Orbital Parameters for Scanning Satellite and Passive RSOs

Orbital Element	**Scanning Satellite**	**Passive RSOs (N=100)**
Semi-major axis (a)	6871 km[a]	6871 km to 8371 km[b]
Eccentricity (e)	0 (circular orbit)	[0.0, 0.02]
Inclination (i)	0° to 180°	0° to 180°
Right Ascension (Ω)	0° to 360°	0° to 360°
Argument of Periapsis (ω)	0° to 360°	0° to 360°
True Anomaly (f)	0° to 360°	0° to 360°

a Fixed at 500 km altitude above Earth's mean radius (6371 km).

b Corresponds to altitudes of approximately 979–1329 km.

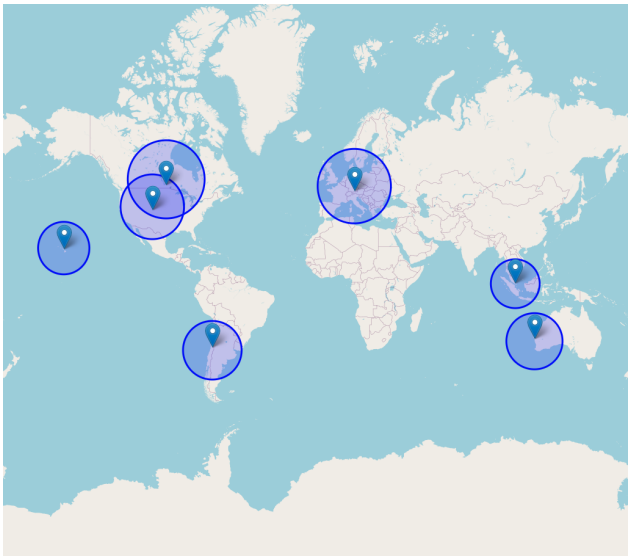

Fig. 2: Groundstation visibility for LEO imaging satellite at 500 km altitude

by an `mrpFeedback` controller[3], which outputs reaction-wheel motor torques subject to per-axis speed/torque limits. The wheel assembly returns the resulting bus torque to the rigid-body dynamics. Although this task focuses on space-to-space imaging (rather than ground-target scheduling), the overall architecture aligns with [42].

2.2 Satellite Dynamics and Control

The scanning satellite used for the imaging operations mimics the well-tested setup used for many AEOS research studies [43, 36]. The control law used to execute attitude maneuvers is Basilisk's `mrpFeedback` controller implemented as detailed in chapter 8 of [44]. To challenge the agent with the need to charge and downlink to maximize performance, the battery charge is initialized between 25% and 50% and the data storage buffer is sized to fit only half of the data needed to image all RSOs. This balance is chosen to neither make the problem trivially easy nor focus primarily on these tasks since the aim for the agent is to perform effectively in terms of imaging (optimizing which target to aim for next) and downlinking targets. The setup of the various satellite properties, controller gains, initial conditions and power drains are summarized in Table 2. Unlisted values follow the BSK-RL defaults.

2.3 Targeting, Imaging and Downlink Constrains

At each action step, the environment identifies the unimaged RSOs within the spacecraft's current FOR, defined by a minimum elevation angle ε_{min}. For a successful image action the targeting of the sensor needs to satisfy, the attitude error requirement, the attitude rate requirement as well as be within for one evaluation of the flight software algorithm (operating at 2 Hz).

[3] `https://hanspeterschaub.info/basilisk/Documentation/fswAlgorithms/attControl/mrpFeedback/mrpFeedback.html`

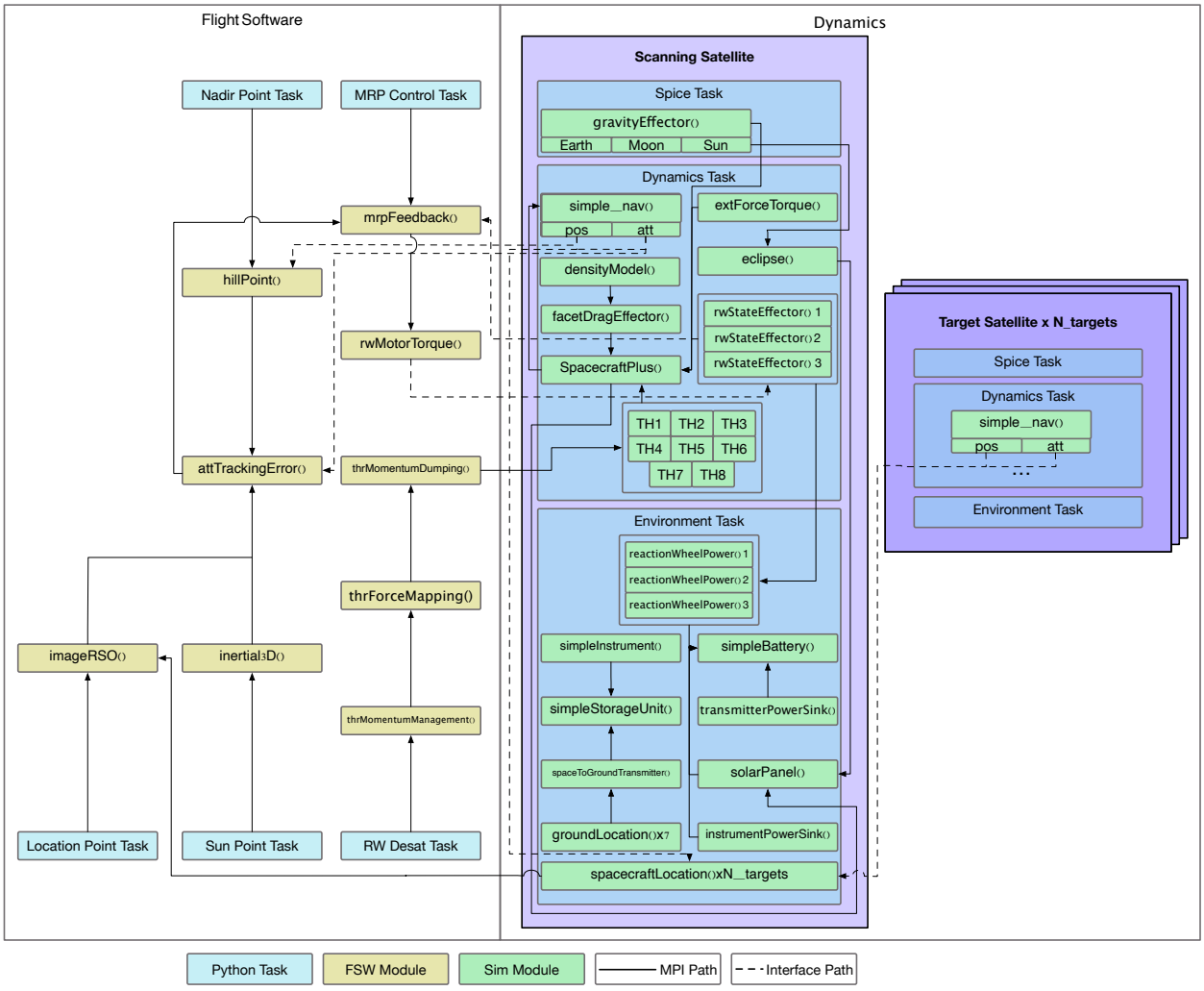

Fig. 3: Basilisk simulation architecture

The various targeting, imaging, and downlink constraints are listed in Table 3.

In this study, the eclipse status was not considered as a binary (shadowed or illuminated) but as a continuum, to also take into account the penumbra effect of a partially eclipsed sun as seen from the RSO, illustrated in Fig. 4.

2.4 Problem Objective

The objective of the satellite is to image and downlink as many RSOs in a given episode duration, while adhering to battery, angular momentum and data storage constraints. For this work, the total time for a full simulation (episode duration) was set to 150% of the time required to image all RSOs (150 imaging actions), which for 100 targets corresponds to 45,000 seconds, around 8 orbits. This design variable is chosen to provide sufficient time for the satellite to be in LOS with all the RSOs (if not the large majority) during at least one step of the episode. While for any given moment only a small percentage of all the RSOs will be visible from the imaging spacecraft, over the course of the entire episode, it essentially sees all (or most of) the targets once or more. Nonetheless, as the eclipse status of the RSOs at every timestep is also considered in this work, the optimal performance given the fixed episode duration is typically less than the total number of RSOs in the simulation.

Given an imaged RSO, the environment evaluates whether imaging constraints (e.g., LOS, eclipse conditions) are satisfied at time of imaging, and assigns a reward accordingly. The episode rollout is limited to a finite horizon, typically spanning several orbits, to evaluate the performance over extended durations. This setup ensures that the environment captures the dynamic and constrained nature of space-based SDA, providing a realistic platform for

Table 2: Spacecraft and control parameters

Parameter	**Value(s)**
Physical	
Mass, Inertia	$m = 330\,\text{kg}, \quad [I_{xx}, I_{yy}, I_{zz}] = [82.1, 98.4, 121.0]\ \text{kg}\,\text{m}^2$
Actuators	$3\times$ Reaction Wheels (orthogonal axes)
RW max torque	$u_{\max} = 0.4\,\text{N}\,\text{m}$ (per-axis)
RW speed limit	6000 RPM
Initial RW Speeds	± 500 RPM
Initial Body Rates	$<0.0001\ \text{rad}\,\text{s}^{-1}$ (random tumble)
Battery Capacity	500 W h
Initial Stored Charge	25–50% of capacity
Solar panel size	$1\ \text{m}^2$ (efficieny 20 %)
Power Drains	
Base Power Draw	10 W
Instrument Power Draw	30 W
Transmitter Power Draw	25 W
Thruster Power Draw	80 W
Other Properties	
Sensor-boresight-axis	Spacecraft z-axis
Desaturation Attitude	Sun-pointing
Control (mrpFeedback)	
Steering gains	$K = 7.0, \ \ P = 35.0$

Table 3: Targeting, Imaging, and Downlink Constraints for the SDA Environment

Constraint	**Value**	**Unit/Description**
Attitude Error Requirement	≤ 0.01	MRP
Attitude Rate Requirement	≤ 0.05	rad/s Eclipse Threshold e_{thresh} 0.5
Single Image Data Size	0.5	Mb
Storage Capacity	25	Mb (50 images)
Initial Storage Fill Level	0	Mb

training and evaluating the RL agent.

The numerical objective function is to maximize the fraction of successfully imaged targets $\mathscr{I}_{\text{ill}}$ out of all the targets $\mathscr{T}$:

$$\text{maximize} \quad \frac{\mathscr{I}_{\text{ill}}}{\mathscr{T}} \tag{1}$$

subject to the mission dynamics, target visibility constraints, and spacecraft constraints (including battery and limited onboard data storage). When an image is taken, the imaging reward is only given if the target RSO is sufficiently lit, defined by $\boldsymbol{s}_i > e_{\text{thresh}}$, where $\boldsymbol{s}_i$ is the i-th target illumination factor. Illuminated and non-illuminated images are both downlinked as they occupy space in the buffer (and it is assumed in this study that images are not analyzed on board), but only those taken under valid illumination contribute to the reward.

The numerical objective in 1 is to maximize the number of RSO targets that are both imaged and successfully downlinked and therefore the task-scheduling must balance data collection and communication opportunities with limitations on energy availability and attitude control authority. This constrained SDA environment formulation encourages planning strategies that achieve high mission performance without compromising long-term operability, and episodes terminate immediately upon critical constraint violation such as battery or reaction wheel failure.

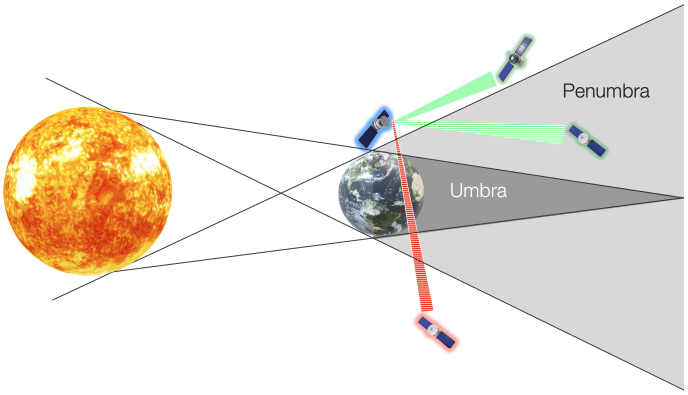

Fig. 4: Space-based RSO imaging under eclipse and LOS restrictions. Green represents a successful imaging, whereas shaded may give partial reward to the agent as it is partially eclipsed. Red would violate the LOS constraint and hence no reward would be given for imaging that target.

3. REINFORCEMENT LEARNING SETUP

Autonomous space-based tasking in a dynamic orbital environment presents a natural setting for RL, where an agent must make sequential decisions to maximize long-term returns under uncertainty[17]. In this study, we formalize the autonomous imaging and downlink scheduling problem for RSO observation as a partially observable Markov decision process (POMDP). This framework allows the agent to learn policies that reason over incomplete state information, adapt to changing target visibility, and manage limited onboard resources and operational constraints.

A POMDP models decision-making under uncertainty about the system state. It consists of a state space $\mathscr{S}$, action space $\mathscr{A}$, transition function $\mathscr{T}$, and reward function $\mathscr{R}$. Because the true state is not fully observable, the agent receives observations from $\mathscr{O}$, governed by the observation model $\mathscr{Z}$. The discount factor $\gamma \in [0,1]$ balances immediate and future rewards. The objective is to choose actions that maximize the expected cumulative reward over time, taking into account the stochastic dynamics of the environment.

Because the true system state is not fully observable, the agent must learn to infer sufficient information from observations $\boldsymbol{o}_t \in \mathscr{O}$ to select actions $a_k \in \mathscr{A}$ that maximize the expected discounted sum of future returns:

$$\mathbb{E}\left[\sum_{k=0}^{T} \gamma^k r_k\right], \tag{2}$$

where k is the step of the environment.

The specific challenges this work addressed include:

- Adapting to orbital dynamics and Earth shadow (eclipse) conditions.
- Operating under limited energy, momentum, and data buffer resources.
- Respecting geometric constraints such as LOS and FOV.
- Prioritizing targets based on policy goals.

3.1 POMDP Formulation

The elements of the POMDP tuple for the inspection task are as follows:

- **State Space:** The underlying simulator state provides the generative model for the MDP and includes the physical dynamics of the inspector spacecraft and RSOs, internal subsystem states (e.g., battery level, data storage, wheel momentum), and external environmental states (e.g., lighting, eclipse windoes, visibility windows for ground stations).

- **Observation Space:** The agent observes a partial view of the full simulator state, composed of normalized quantities relevant to the imaging and planning tasks. The components of the observation vector are listed in Table 4.

Table 4: Observation space elements provided to the agent at each timestep.

Element	Description	Dim.
s_{data}	Fraction of onboard data storage used	1
s_{batt}	Normalized battery charge level	1
s_{mom}	Normalized wheel momentum	1
$\boldsymbol{\varepsilon}_i$	Elevation angles of visible target i	N
$\boldsymbol{r}_{BR,i}^{\mathscr{H}}$	Relative position vectors to target i in Hill frame, $\mathscr{H}$	$3 \times N$
$\boldsymbol{\theta}_i$	Angle between boresight and target i	N
$\boldsymbol{d}_i$	Distance to target i	N
$\boldsymbol{s}_i$	Illumination factor (shadowing value) for target i	N
$\boldsymbol{e}_{\text{start}}$, $\boldsymbol{e}_{\text{end}}$	Eclipse Normalized start/end times	2
$\boldsymbol{g}_{\text{open}}$, $\boldsymbol{g}_{\text{close}}$	Normalized ground stations window open/close times	2x5

- **Action Space:** At each timestep, the agent selects one action from a discrete action set:
  - Image(i) – Image a selected RSO from $N = 10$ visible targets (for 5 minutes),
  - Charge – Enter charging mode for 5 minutes,
  - Downlink – Initiate data downlink for 3 minutes,
  - Desat – Perform momentum desaturation for 2.5 minutes.

  The imaging action targets one of the top-N RSOs sorted by elevation angle. Other actions are global spacecraft operations.

- **Reward Function:** The reward $r(t)$ at time t is determined by whether the selected imaging target was successfully imaged under valid illumination conditions and whether a previously collected image was downlinked. Rewards are only granted for illuminated captures (not shadowed ones), and downlinked data is only valuable if the original image was illuminated.

$$r(t) = \mathrm{w}_i \tag{3}$$

$$\text{s.t.} \quad \boldsymbol{s}_i > e_{\text{thresh}}, \tag{4}$$

$$\text{LOS}_i, \tag{5}$$

$$\angle\left(\hat{\boldsymbol{z}}_b, \hat{\boldsymbol{\rho}}_i\right) \leq \text{MRP}(0.01) \tag{6}$$

  here:
  - $\boldsymbol{w}_i$ is the priority of the i-th target,
  - $\boldsymbol{s}_i$ is the illumination factor of the i-th target,
  - $\mathrm{e}_{\text{thresh}}$ is the illumination threshold to be considered sufficiently lit,
  - LOS_i is the binary measure of whether line-of-sight is present between the scanning satellite and the i-th target,
  - $\hat{\boldsymbol{z}}_b$ is the spacecraft body z-axis,
  - $\hat{\boldsymbol{\rho}}_i$ is the unit vector from the spacecraft to target i,
  - the angle constraint enforces alignment within 0.01 Modified-Rodriguez Parameters (equivalently, $\leq 2.29^\circ$).

The agent's goal is to maximize the cumulative reward over the episode as defined in Eq. 2, subject to operational constraints like energy availability, data storage, momentum limits, and imaging cadence.

- **Transition Model:** The environment dynamics are implemented as a deterministic generative model rather than a probabilistic transition function. At each timestep, the agent selects one discrete action (e.g., imaging, charging, downlinking), which is then executed over a fixed time interval. The environment simulates the spacecraft's dynamics, onboard resource state updates, orbital lighting conditions, and RSO visibility during that interval. The environment state is then used to calculate the associated reward for the agent as well as passing it the next observation.

3.2 RL Training

Due to the high dimensionality of spacecraft dynamics and the complexity of sensor-based observation models, deep RL with the Proximal Policy Optimization (PPO) algorithm [35] is employed. PPO is an actor-critic architecture, where the actor represents the policy network that chooses actions based on current observations and the critic evaluates the performance of the policy and compares it to its own predictions. To improve the policy over iterations, the critic is used as feedback for the actor to improve, while the prediction error is used in turn to update the critic. This algorithm is a leading on-policy method that achieves a favorable trade-off between sample efficiency and training stability. It does so by limiting how much the policy is allowed to change at each step, using a clipped surrogate objective to prevent overly aggressive updates that could destabilize learning. Training was done using the ray "RLlib" library, which can interface with BSK-RL, on an Apple Silicon M4 CPU typically over a span of 24 hours, until 20 million environment steps have been completed. As part of the training optimization, a range of hyperparameter were varied including the learning rate, discount factor, training batch size, network size, clip parameters of the PPO as well as the gradient. The set of hyperparameter used is found in Table 9 of the appendix (otherwise the default Rllib values were used. Given the different duration of the actions, a time discounted generalized advantage estimate was used, meaning the discount factor is applied per second and not per step. It is worth noting that the training of the policies was a big challenge and required a large search of hyperparameters.

4. RESULTS

To evaluate the performance of the proposed RL scheduling approach, we compare it against a myopic heuristic policy. The heuristic policy is provided with the full list of unimaged targets currently within LOS. Out of those targets, it chooses the one with the smallest angular pointing error, without consideration for future opportunities or resource states. Two sepatate comparisons are made. Firstly, only the imaging performance is tested with consideration of just the eclipse threshold e_{thresh} to obtain illuminated images, with otherwise unlimited spacecraft resources. Secondly, another comparison with the inclusion of restricted resources and need to downlink to specified ground stations is made. In this second test case, both approaches operate under an identical safety shield that intervenes when the onboard data storage exceeds 99% capacity or when the battery state-of-charge drops below 20%. In those instances the shield overrides the agent's decision to enforce downlinking or charging actions, respectively. In the scenario where the battery as well as the storage shield intervention is triggered,the charging action will be tasked to keep the spacecraft alive. Notably, for the chosen episode duration used for training and testing, the spacecraft did not accumulate a large enough amount of angular momentum to require desat maneuvers. It has been shown that when the agent is exposed to artificial external torques in training, it is able to learn to utilize the desat action to keep the agent alive [45]. For the comparisons shown in this paper, no artificial external torque is applied.

4.1 Baseline Comparison

First the sensing problem is isolated from spacecraft resource management by removing all operational constraints (no battery, storage, or desaturation limits). In this setting the shield is inactive for both methods, and downlink/charging actions are never required. As a result, performance is dominated by line-of-sight (LOS) geometry and target selection alone. For this study a separate RL-policy was trained with exactly this environment setup. The main metrics of this study are summarized in Table 5.

As expected, both methods achieve similar end-to-end performance when resource limits are removed. Over 100 Monte Carlo episodes, the heuristic attains an average of 89.54 ± 2.99 (mean$\pm$std) successful images, while the RL-policy reaches 90.47 ± 2.95, a small average gain of around 0.9 images per run.

Table 5: Baseline (imaging-only) comparison over 100 runs. Values are mean±std. No resource actions or shield interventions occur in this setting.

Metric	Heuristic	RL policy
Illuminated images (count)	89.54 ± 2.99	90.47 ± 2.95
Illumination fraction (%)	71.28 ± 2.90	88.97 ± 3.79

Although overall performance is close, the RL-policy exhibits a clear preference for better lighting. The *illumination fraction* (fraction of targets that are illuminated when selected) increases from $71.28\% \pm 2.90\%$ (heuristic) to $88.97\% \pm 3.79\%$ (RL-policy).

Qualitatively, this manifests as the RL-policy biasing its pointing toward better-lit geometry especially around eclipse transitions: near eclipse entry the agent often "looks back" to capture still-lit targets, and near exit it "looks forward" to acquire the first illuminated opportunities. In this constraint-free regime, no charging/downlink/desat events or shield interventions occur for either method, as designed.

4.2 Aggregate Results under Eclipse Constraints and Resource Limits

Next, the two approaches are compared again with eclipse constraints but also under resources restrictions, namely: battery, data storage and angular momentum constraints. This makes the task more challenging as the agent needs to ensure it keeps its battery alive, while also necessitating downlinks when in range to groundstations, to free up storage and allow for more targets to be imaged. Again, 100 Monte Carlo (MC) episodes per policy were run in the restricted resource environment with the same intervention shield (storage $\geq 99\% \Rightarrow$ downlink; battery $\leq 20\% \Rightarrow$ charge) for both approaches. The resulting statistic are shown reported in Table 6 .

Table 6: Restricted-resources, eclipse-constrained MC study (100 runs each). Values are mean $\pm$ std unless noted. Downlink *yield* is useful/total.

Metric	Heuristic + Shield	RL + Shield
Illuminated images (mean)	86.85 ± 4.43	84.45 ± 3.71
Illumination fraction (%)	71.96 ± 3.38	87.23 ± 3.61
Total downlinks	77.15 ± 8.85	78.42 ± 13.69
Useful downlinks	69.80 ± 9.13	71.08 ± 13.02
Downlink yield (%)	90.47	90.64
Charge events over 100 runs	96	16
Shield interventions / run	22.82	3.26

Across the MC population, the heuristic yields +2.8% more successful images than the RL-policy (86.85 ± 4.43 vs. 84.45 ± 3.71).

Despite imaging more targets on average, the heuristic downlinks slightly fewer useful images than the RL-policy (69.80 ± 9.13 vs. 71.08 ± 13.02), with a nearly identical downlink yield (90.47% vs. 90.64%). Notably, the standard deviation in the number of useful downlinks is significantly higher than the equivalent for the illuminated images. This is particularly because the groundstations in the environment are in fixed locations and therefore, for some randomly generated orbit of the scanning satellite, provide potentially minimal coverage. On the other hand, since the targets are also generated randomly, there is no large deviation in the imaging performance when the scanning satellite is placed on a different orbit. This robustness of imaging to the orbit plane of the agent was also demonstrated in [23]. Moreover, while the heuristic outperforms the RL-policy, it also requires on average six times more charging actions compared to the trained model.

Overall, the MC study shows that the learned policy internalizes resource-aware scheduling and timely downlinking, yielding comparable illuminated collection (slightly lower than the updated heuristic baseline here) while improving delivery yield and autonomy (substantially fewer shield interventions).

4.3 Case Study for Representative Episode

In this section, a single episode is analyzed in more detail to highlight the different behaviour between the heuristic and the RL-policy.

4.3.1 Qualitative Behavior

Fig. 5 shows the temporal evolution of battery and storage usage, cumulative imaging and downlink counts, and total accumulated reward for the RL-policy in a representative simulation run. Key mission phases, including eclipse (umbra) and ground station visibility windows, are marked for reference. The RL-policy consistently exploits eclipse periods, when many LEO-to-LEO targets are also in darkness and therefore unobservable, to prioritize data downlinking. This behavior emerges naturally from training and results in productive use of otherwise idle time.

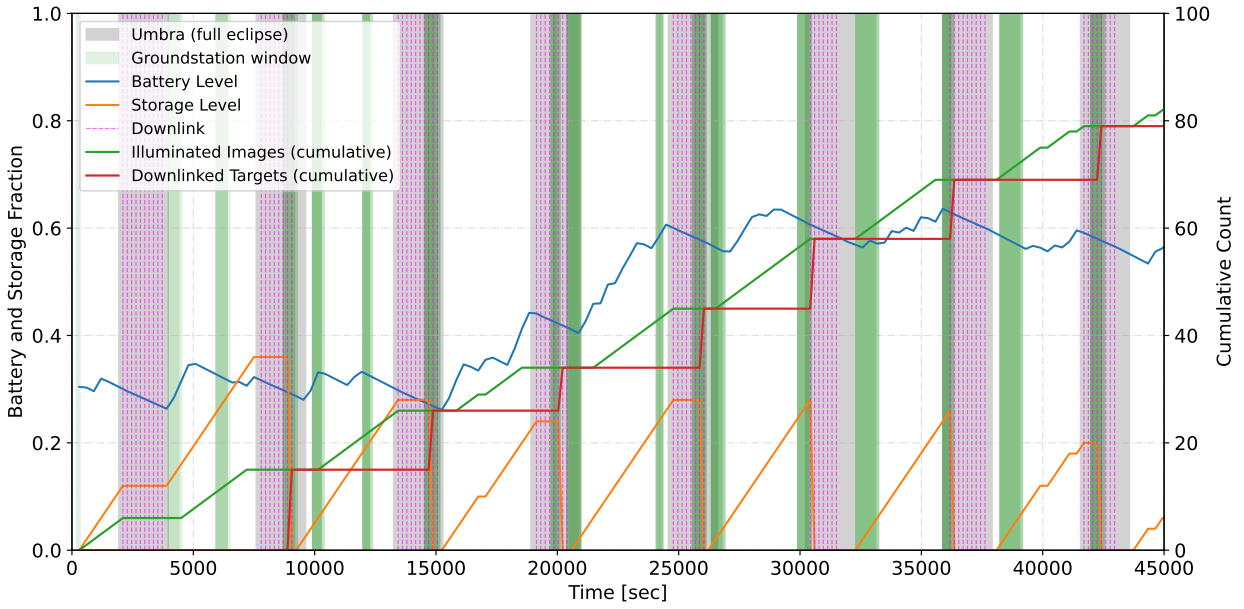

Fig. 5: RL-policy time series: battery/storage fractions, cumulative imaging and downlink counts, reward, with eclipse (umbra), and ground-station windows annotated.

Fig. 6 illustrates that the heuristic policy is fully dependent on shield interventions to avoid critical battery depletion. Rather than preemptively scheduling downlinks or aligning observations with energy margins, it continues to select imaging actions until the shield forces corrective measures. As a result, the heuristic spends less time in effective downlink operations and approaches high storage levels that limit further imaging. This behaviour entails that the downlinked images are typically older since they were taken longer ago. In contrast, the RL-agent learns to downlink every eclipse, which means more recent, and therefore valuable, images are sent to the ground.

A notable emergent strategy in the RL-agent is *dual-use target selection*, rarely requiring a dedicated charging action. In several intervals, the agent selects imaging opportunities that are spatially and temporally aligned with favorable solar illumination, permitting battery recharge while collecting images. This hybrid action profile maintains higher battery levels without requiring explicit charging actions in this example.

4.3.2 Quantitative Performance

Across multiple seeds, the RL-policy maintains higher autonomy (fewer shield interventions) and timelier delivery to ground stations relative to the heuristic baseline as well as better battery management. For the representative single episode shown in Fig. 5–6, the RL-policy produced fewer illuminated images than the heuristic (82 vs. 90) but achieved a higher imaging success rate (74.55% vs. 68.18%), performed more downlink operations (93 total; 79 useful vs. 83

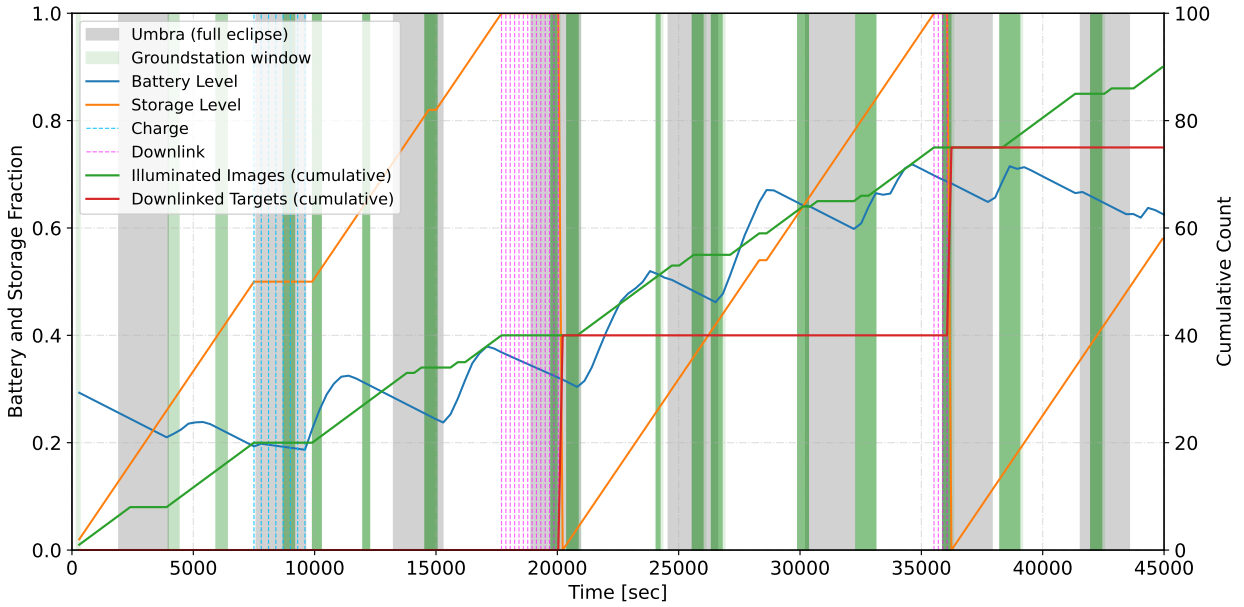

Fig. 6: Heuristic policy time series: the policy relies fully on shield interventions for charging and downlinking, leading to delayed deliveries and lower energy margins.

total; 75 useful), and required *zero* shield interventions (vs. 26 for the heuristic). Importantly, the RL-policy also improves *timeliness*: images taken during the day are typically downlinked in the following nighttime pass, i.e., within roughly one orbit. Conversely, the heuristic relies on the shield to initiate downlinks, when storage is entirely full. This delays the delivery and in this case only resulted in two successful donwlink action, meaning the resulting imagery is older on arrival at the ground and thus of reduced operational value.

Table 7: Single-episode performance summary comparing the RL-policy (with shield) to the myopic heuristic (with shield).[a]

Metric	**RL + Shield**	**Heuristic + Shield**
Illuminated images (count)	82	90
Imaging success rate (%)	74.55	68.18
Useful downlinks (count)	79	75
Mean target illumination fraction (%)	82.73	72.79

[a] "Useful downlinks" are those that deliver an illuminated image to the ground. Imaging success rate is computed as illuminated images divided by the total number of imaging actions.

Table 7 summarizes key metrics for this individual episode. The heuristic captured more illuminated images (90 vs. 82), yet the RL policy was more efficient (74.55% vs. 68.18% imaging success) and delivered more useful imagery to the ground (79 vs. 75). Notably, the RL-policy did not require any charging actions, while in the heuristic case, the shield triggered charging eight times. Furthermore, successful downlink actions occurred six times compared to twice for the heuristic. Both of these highlight the utility of the RL-policy's eclipse-time downlink strategy and dual-use imaging action behaviour.

4.4 Interpretation

These results indicate that the RL policy internalizes two complementary behaviors: (i) proactive eclipse-time downlinking, which preserves daytime imaging opportunities and improves timeliness to the ground, and (ii) resource-aware target selection that simultaneously supports battery charging during imaging. Together, these behaviors reduce reliance on the safety shield, sustain higher energy margins, while still yielding a comparable number of illuminated images to the myopic baseline. Both the RL policies trained (one to just image with unlimited resources and the other

with spacecraft constraints), avoid imaging actions in the eclipse phase that target other nearby targets that are also eclipsed. In those scenarios, both RL policies exhibit preferential behavior to farther targets which are beyond the eclipse and hence still illuminated. Therefore, these perform better than the heuristic in terms of the illumination fraction and similarly overall, despite a smaller number of imaging actions taken in the resource restricted environment.

5. CONCLUSIONS

With the growing population of RSOs in LEO, autonomous planning of imaging and downlink is essential for scalable SDA. This work addressed space-to-space imaging from a LEO platform, first in an unconstrained setting and then under strict operational constraints, by casting the problem as a partially observable Markov decision process and training a PPO-based deep RL agent in a high-fidelity simulation environment. The agent balanced illuminated image capture and timely ground delivery while respecting limits on energy, angular momentum, and data storage. Evaluation over 100 MC runs showed comparable number of illuminated image taken under resource limits but preferential downlink cadence was shown by the RL-policy. In the representative episode shown, the heuristic captured more illuminated images (90 vs. 82), while the RL-policy delivered more useful images to the ground (79 vs. 75) more frequently and operated with zero shield interventions (vs. 26), indicating greater autonomy and timeliness. The learned policy exhibits behaviour that exploits eclipse periods for proactive downlink and selects targets that supported passive battery charging during imaging, reducing reliance on safety overrides and improving use of observation opportunities in a dynamic LEO-to-LEO environment.

The results highlight the potential for RL-based onboard autonomy to improve the responsiveness and efficiency of space-based SSA missions. While this study assumed known target states and did not address orbital state uncertainty, the demonstrated framework provides a foundation for integrating estimation processes, handling uncertainty, and coordinating multiple spacecraft via Multi-Agent RL. Expanding to such scenarios could further enhance the scalability and robustness of autonomous space surveillance, contributing to more sustainable and resilient space operations.

Future work will allow variable-duration imaging actions to relax fixed step lengths. Training will integrate safety shields into the policy and encourage immediate downlink upon ground-station contact to avoid wasted actions. Robustness will be evaluated under varying RSO counts, longer horizons than training, and external torque disturbances. Finally, incorporating state uncertainty with onboard estimation and extending to multi-satellite coordination would bring the approach closer to operations and improve scalability and resilience.

6. ACKNOWLEDGMENTS

This work is supported by the Air Force STTR Phase 1 project with award No. FA9550-25-P-B003. This work utilized the Alpine high-performance computing resource at the University of Colorado Boulder. Alpine is jointly funded by the University of Colorado Boulder, the University of Colorado Anschutz, Colorado State University, and the National Science Foundation (award 2201538).

7. REFERENCES

[1] Andrew Williams, Olivier Hainaut, Angel Otarola, Gie Han Tan, Andrew Biggs, Neil Phillips, and Giuliana Rotola. A report to eso council on the impact of satellite constellations. Technical report, European Southern Observatory (ESO), 2021.

[2] Jose Alberto Hernandez and Pedro Reviriego. A brief introduction to satellite communications for NTN. *arXiv*, 2023.

[3] Mark R Ackermann, Colonel Rex R Kiziah, Peter C Zimmer, J T McGraw, John T McGraw, J T McGraw, and David D Cox. A Systematic Examination Of Ground-Based And Space-Based Approaches To Optical Detection And Tracking Of Satellites.

[4] Bin Jia, Khanh D. Pham, Erik Blasch, Dan Shen, Zhonghai Wang, and Genshe Chen. Cooperative space object tracking using space-based optical sensors via consensus-based filters. 52(4):1908–1936.

[5] L. Ansalone and F. Curti. A genetic algorithm for initial orbit determination from a too short arc optical observation. *Advances in Space Research*, 52(3):477–489, 2013.

[6] D. A. Vallado and S. S. Carter. Accurate orbit determination from short-arc dense observational data. *Journal of the Astronautical Sciences*, 46(2):195–213, 1998.

[7] T. Flohrer, H. Krag, H. Klinkrad, and T. Schildknecht. Feasibility of performing space surveillance tasks with a proposed space-based optical architecture. *Advances in Space Research*, 47(6):1029–1042, 2011.

[8] J.-S. Ardaens and G. Gaias. A numerical approach to the problem of angles-only initial relative orbit determination in low earth orbit. *Advances in Space Research*, 63(12):3884–3899, 2019.

[9] S. M. Lenz, H. G. Bock, J. P. Schlöder, E. A. Kostina, G. Gienger, and G. Ziegler. Multiple shooting method for initial satellite orbit determination. *Journal of Guidance, Control, and Dynamics*, 2012.

[10] G. Sciré, F. Santoni, and F. Piergentili. Analysis of orbit determination for space based optical space surveillance system. *Advances in Space Research*, 56(3):421–428, 2015.

[11] G. M. Goff, J. T. Black, and J. A. Beck. Tracking maneuvering spacecraft with filter-through approaches using interacting multiple models. *Acta Astronautica*, 114:152–163, 2015.

[12] Erik Hedenström. Tracking sensor network schedule optimization for space surveillance. Master's thesis, KTH Royal Institute of Technology, 2025.

[13] Neil K. Dhingra, Cameron DeJac, Alex Herz, and Roderick Green. Space domain awareness sensor scheduling with optimality certificates. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, Maui, HI, USA, 2023. Accessed 2025-08-17.

[14] N. Herz and R. Wimmer-Schweingruber. Scheduling algorithms for ground-based optical space surveillance. *Acta Astronautica*, 87:1–11, 2013.

[15] Thomas Hinze, Jan Strohmer, and Benjamin Bastida Virgili. A genetic algorithm for optimizing geo object observation schedules. In *Proceedings of the AMOS Technical Conference*, 2012.

[16] Boris Shteinman and Moriba K. Jah. Information-gain driven auction-based scheduling for space surveillance sensors. In *Proceedings of the AMOS Technical Conference*, 2018.

[17] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition, 2018.

[18] Daniel Jang, Peng Mun Siew, David Gondelach, and Richard Linares. Space situational awareness tasking for narrow field of view sensors: A deep reinforcement learning approach. 71st International Astronautical Congress. International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law.

[19] Richard Linares and Roberto Furfaro. An Autonomous Sensor Tasking Approach for Large Scale Space Object Cataloging.

[20] Peng Mun Siew, Tory Smith, Ravi Ponmalai, and Richard Linares. Scalable Multi-Agent Sensor Tasking Using Deep Reinforcement Learning.

[21] Richard Linares and Roberto Furfaro. Dynamic Sensor Tasking for Space Situational Awareness via Reinforcement Learning.

[22] Benedict Oakes, Jason F Ralph, and Jordi Barr. Deep Reinforcement Learning Applications to Space Situational Awareness Scenarios.

[23] Peng Mun Siew, Daniel Jang, Thomas G. Roberts, and Richard Linares. Space-Based Sensor Tasking Using Deep Reinforcement Learning. 69(6):1855–1892.

[24] Grant Stokes, Curt Vo, Ramaswamy Sridharan, and Jayant Sharma. The space-based visible program. In *Space 2000 Conference and Exposition*. American Institute of Aeronautics and Astronautics.

[25] Michael Gaposchkin, prefix=von useprefix=true family=Braun, given=Curt, and Jayant Sharma. Space-Based Space Surveillance with the Space-Based Visible. 23(1):148–160.

[26] Peng Mun Siew, Tory Smith, Ravi Ponmalai, and Richard Linares. Scalable multi-agent sensor tasking using deep reinforcement learning. In *Proceedings of the Advanced Maui Optical and Space Surveillance Technologies (AMOS) Conference*, Maui, HI, USA, 2023. Program listing; paper appears in AMOS 2023 proceedings.

[27] M. Nazari, A. Oroojlooy, L. Snyder, and M. Takac. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[28] A. Harris, T. Teil, and H. Schaub. Spacecraft decision-making autonomy using deep reinforcement learning. In *AAS Spaceflight Mechanics Meeting*.

[29] A. Hadj-Salah, R. Verdier, C. Caron, M. Picard, and M. Capelle. Schedule earth observation satellites with deep reinforcement learning.

[30] D. Eddy and M. Kochenderfer. Markov decision processes for multi-objective satellite task planning. In *2020 IEEE Aerospace Conference*, pages 1–12. IEEE.

[31] A. Harris, T. Valade, T. Teil, and H. Schaub. Generation of spacecraft operations procedures using deep reinforcement learning. 59:611–626.
[32] A. Herrmann and H. Schaub. Reinforcement learning for the agile earth-observing satellite scheduling problem. pages 1–13.
[33] Lorenzzo Quevedo Mantovani, Yumeka Nagano, and Hanspeter Schaub. Reinforcement Learning For Satellite Autonomy Under Different Cloud Coverage Probability Observations.
[34] Mark Stephenson and Hanspeter Schaub. Reinforcement Learning For Earth-Observing Satellite Autonomy With Event-Based Task Intervals.
[35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.
[36] L. Q. Mantovani, Y. Nagano, and H. Schaub. Reinforcement learning for satellite autonomy under different cloud coverage probability observations. *AAS/AIAA Astrodynamics Conference*, 2023.
[37] M. Stephenson and H. Schaub. Reinforcement learning for earth-observing satellite autonomy with event-based task intervals. *AAS/AIAA Space Flight Mechanics Conference*, 2024.
[38] Mark Stephenson, Daniel Huterer Prats, and Hanspeter Schaub. Autonomous satellite inspection in low earth orbit with optimization-based safety guarantees. In *International Workshop on Planning Scheduling for Space*, Toulouse, France, April 28–30 2025.
[39] Patrick W. Kenneally, Scott Piggott, and Hanspeter Schaub. Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework. *Journal of Aerospace Information Systems*, 17(9):496–507, September 2020.
[40] Mark A Stephenson and Hanspeter Schaub. BSK-RL: Modular, High-Fidelity Reinforcement Learning Environments for Spacecraft Tasking. In *75th International Astronautical Congress*, Milan, Italy, October 2024. IAF.
[41] Shkelzen Cakaj, Bexhet Kamo, Vladi Koliçi, and Olimpjon Shurdi. The Range and Horizon Plane Simulation for Ground Stations of Low Earth Orbiting (LEO) Satellites. 04(09):585–589.
[42] Adam P. Herrmann and Hanspeter Schaub. Monte carlo tree search methods for the earth-observing satellite scheduling problem. *Journal of Aerospace Information Systems*, 19(1), January 2022.
[43] Mark A. Stephenson and Hanspeter Schaub. Optimal Agile Satellite Target Scheduling with Learned Dynamics. *Journal of Spacecraft and Rockets*, 62(3):793–804.
[44] Hanspeter Schaub and John Junkins. Nonlinear Spacecraft Stability and Control. In *Analytical Mechanics of Space Systems, Fourth Edition*, AIAA Education Series, pages 387–518. American Institute of Aeronautics and Astronautics, Inc., 4 edition.
[45] Lorenzzo Quevedo Mantovani and Hanspeter Schaub. Improving Robustness Of Autonomous Spacecraft Scheduling Using Curriculum Learning.

8. APPENDIX

Table 8: Ground Station Locations with ECEF Coordinates (in km), Geodetic Latitude/Longitude. The Minimum Elevation Angle is 10° for all stations.

Ground Station	**Latitude [°]**	**Longitude [°]**	**X [km]**	**Y [km]**	**Z [km]**	**Slant Range [km]**
Boulder	40.01	-105.25	−1284.8	−4714.5	4101.7	1690.92
Merritt	50.11	-97.26	910.7	−5540.5	3025.6	1694.73
Singapore	1.34	103.81	−1523.1	6191.8	150.5	1694.71
Weilheim	47.82	11.09	4200.6	827.3	4728.4	1693.52
Santiago	-33.45	-70.67	1761.8	−5022.2	−3515.9	1693.38
Dongara	-29.25	114.87	−2346.0	5046.5	−3116.0	1694.81
Hawaii	20.71	-156.25	−5461.0	−2479.2	2170.7	1694.71

Table 9: RL Training Parameters

Name	**Value**
Learning rate	1×10^{-6}
Discount factor (γ)	0.9997
Gradient clip	1.0
PPO clip parameter (ε)	0.15
Training batch size	3200