Reinforcement Learning with Hybrid Action Representation for Autonomous Strip Imaging Task Scheduling in High-Resolution Super-Agile Satellites

Anaïs Cheval* and Hanspeter Schaub†
Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO 80303

This paper investigates the use of Deep Reinforcement Learning (DRL) to address the scheduling problem for strip imaging tasks in the context of Super-Agile Earth-Observing Satellites (SAEOS). While DRL has proven effective for discrete point-target scheduling, existing literature has not considered the additional complexities of strip imaging tasks. Unlike static point targets, strip imaging requires tracking a moving reference point along a strip's central line, imposing strict constraints on transition times between tasks to satisfy field-of-view, attitude, and acquisition-speed requirements. As a result, the scheduler must not only select the next task to image but also accurately estimate the transition time required before starting image acquisition. To address these challenges, the strip-imaging scheduling problem is formulated as a Partially Observable semi-Markov Decision Process (POsMDP) with a hybrid action space, and a DRL framework specifically tailored for this problem HOP-PPO (Hybrid Observation task Planning–oriented Proximal Policy Optimization) is introduced. HOP-PPO employs a dual-actor architecture to effectively handle the hybrid action space and incorporates a transformer-based observation encoder to capture structural dependencies among tasks. A custom training environment is constructed using BSK-RL, a Python package based on Basilisk, which enables the creation of Gymnasium environments tailored for spacecraft task scheduling. Experimental results demonstrate that HOP-PPO outperforms the heuristic baseline by 36.3% in mean cumulative reward and achieves at least 25.9% superior performance compared to discretization-based and standard hybrid DRL methods.

I. Introduction

Continuous monitoring of the Earth is essential for the timely detection and response to dynamic events such as floods, wildfires, and storms. Achieving this requires a combination of low-resolution and high-resolution scanning satellites, as exemplified by NASA's SensorWeb [1]. Low-resolution satellites rapidly identify areas of concern across large regions, mapping the entire Earth at least once per day (e.g., MODIS [2]). High-resolution satellites, in contrast, provide the detailed imagery needed for precise assessment and informed decision-making in the identified areas by the low resolutions satellites. To be effective, high-resolution satellites must exhibit super-agility, enabling them—through advanced attitude control systems—to decouple the scanning direction from their orbital path and the image acquisition rate from their orbital motion. Additionally, their task planning system must efficiently schedule area imaging tasks to maximize coverage of high-priority areas, while allowing rapid retasking in response to new identified areas through low-resolution observations.

Traditionally, planning for Earth-observing satellites (EOS) has been performed on the ground, where a sequence of imaging tasks is optimized and then uplinked to the spacecraft for open-loop execution. The most common optimization-based approach for this process is Mixed-Integer Linear Programming (MILP), which is favored for its ability to provide mathematically optimal solutions [3, 4]. Industry leaders such as Spire Global [5] and Planet [6] employ MILP-based methods to efficiently manage and schedule imaging tasks across their satellite constellations. However, any change in initial conditions or the introduction of new tasks requires full or partial replaning. MILP methods are computationally intensive and scale poorly with an increasing number of satellites and targets, making

*Ph.D. Student, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO 80303. Correspondence: anais.cheval@colorado.edu

†Distinguished Professor and Department Chair, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado Boulder, Boulder, CO 80303. Fellow of AIAA and AAS

them unsuitable for rapid retasking in real-time event monitoring. To support adaptive planning and overcome MILP's constraints, NASA's SensorWeb proposes an heuristic-based approach. Two complementary systems are used: ASPEN, which operates on the ground for non-time-critical planning, and CASPER, which functions onboard for real-time decision-making. Both systems utilize a stochastic local search algorithm combined with a portfolio of heuristics to iteratively repair and improve plans. While this heuristic-based approach offers significant advantages in responsiveness, it lacks optimality and relies on handcrafted heuristics that can be difficult to tune and generalize.

The use of Deep Reinforcement Learning (DRL) has been proposed to address these limitations. More specifically, DRL has shown the ability to effectively solve the scheduling problem for point imaging tasks for both individual Earth-observing satellites [7] and decentralized constellations [8, 9]. DRL agents are first trained offline on high-fidelity simulations to map states to actions to maximize a numerical reward function. A common practice is to train the algorithm using a variety of random initial conditions, and targets to generalize the policy. After the training step, the policy can be up-linked to the satellite for closed-loop on-board execution, responding to the real states of the environment, which means that re-planning is inherent to a DRL planning paradigm. The execution of trained policies is typically fast. Neural network approximations of the scheduling policy can be executed in milliseconds on modern computational hardware.

However, these prior DRL studies focus exclusively on discrete point-target imaging tasks, not considering the challenges of area imaging tasks. Large areas can be decomposed into a set of strips, framing the challenge as a sequential strip imaging problem. This research seeks to bridge this gap by applying DRL to address the scheduling problem for continuous strip imaging tasks in the context of high-resolution Super-Agile Earth Observing Satellites (SAEOSs). A critical difference between point and strip imaging lies in the need to explicitly determine transition times between consecutive tasks to ensure heterogeneous imaging requirements. In point imaging, the target is fixed in the Earth-fixed frame; the satellite simply slews to the target and tracks it until completion. In contrast, strip imaging involves tracking a virtual point that moves at the required acquisition speed along the strip's central line. These required scanning speeds vary from target to target. As a result, accurate estimation of the transition time is critical: the satellite must have completed its slewing maneuver to line up with the strip imaging trajectory to meet attitude and acquisition speed accuracy requirements and the strip must have entered its field of view before imaging can begin. Any deviation from this transition time prevents proper synchronization with the moving target, resulting in coverage gaps and failure to meet the imaging requirements.

In the context of strip imaging, the transition time estimation between tasks introduces an additional layer of complexity for decision making: the agent must not only choose which task to image next from a discrete set, but also determine an appropriate transition time from a continuous domain. As a result, the action space is no longer purely discrete, as in point-target imaging, but becomes a hybrid action space where a continuous decision is conditioned on a discrete decision. Handling such action spaces poses challenges because widely used DRL algorithms, such as Proximal Policy Optimization (PPO) [10], are typically designed for either fully discrete or fully continuous action spaces. A straightforward workaround is to discretize the continuous component [11], but this approach rapidly becomes impractical: it requires choosing a discretization granularity and leads to extremely large discrete action sets, ultimately harming scalability and performance.

Recent research has explored modifications of DRL algorithms to handle hybrid action spaces. PPO is of particular interest due to its demonstrated effectiveness in point imaging tasks, as confirmed by comparative studies [12]. Hybrid-PPO [13] (H-PPO) extends the standard PPO actor–critic architecture by employing two separate actors with a shared state encoder: one selects the discrete task, while the other outputs continuous parameters for each possible task. The final action sent to the environment is the combination of the chosen discrete task and its corresponding continuous parameter. However, the architecture of H-PPO does not align well with the structure of the Strip Imaging Scheduling problem. Specifically, it fails to exploit structural similarities between discrete tasks and relies on a shared state encoder for both actors, preventing the necessary actor specialization: the discrete actor requires a global view of all strips, whereas the continuous actor relies only on local, strip-specific features.

This paper first introduces the attitude guidance and control system required to execute strip imaging tasks in the context of Super-Agile Earth Observing Satellites (SAEOSs). It then formulates the strip imaging scheduling problem as a Partially Observable semi-Markov Decision Process (POsMDP) with a hybrid action space. A Hybrid Observation task Planning–oriented Proximal Policy Optimization framework (HOP-PPO) specifically tailored to this formulation is subsequently proposed. This framework uses a dual-actor architecture to effectively handle the hybrid action space and incorporates a transformer-based observation encoder to capture structural dependencies among tasks. The resulting policy is benchmarked against H-PPO, discretization-based methods, and classical heuristics to demonstrate the advantages of the approach.

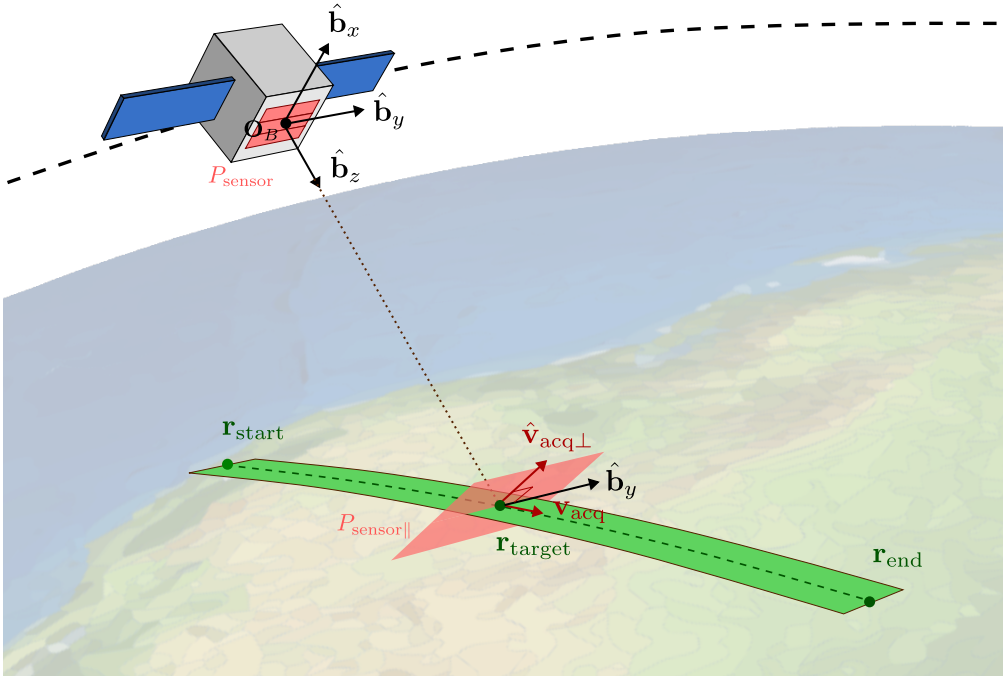


Fig. 1 Attitude Guidance for a Strip Imaging Task.

II. High-Fidelity Modelling of a strip imaging task

A. Strip Request Model

Imaging requests are modeled as a set $\mathcal{R}$ of strip targets, where each request $\rho \in \mathcal{R}$ is defined by a tuple $(\mathbf{r}_{\text{start}}, \mathbf{r}_{\text{end}}, p, v_{\text{acq}})$ representing a start point and end point, both defined as fixed locations in the planet-fixed frame, a priority level p, and a required acquisition speed v_{acq}. All imaging requests begin in the unfulfilled set $\mathcal{U}$ and are moved to the fulfilled set $\mathcal{F}$ once successfully imaged. After fulfillment, requests are considered complete and are not re-imaged, although operators may add new requests for the same location if necessary. If a strip can be imaged from both directions, an additional request is generated with the start and end points reversed but sharing the same priority and acquisition speed. These two requests are treated as linked: fulfilling either one causes both to be moved from $\mathcal{U}$ to $\mathcal{F}$.

In this work, synthetic imaging strip requests are generated by first selecting a random start point $\mathbf{r}_{\text{start}}$ uniformly distributed over the planet's surface. The corresponding end point $\mathbf{r}_{\text{end}}$ is computed by propagating the start point along a great circle. The direction is determined by a uniformly sampled azimuth, and the length of the strip is drawn uniformly from the range $[L_{\min}, L_{\max}]$. Each imaging request is assigned a priority value p randomly drawn from a uniform distribution over the interval $[0, 1]$, along with an acquisition speed v_{acq} sampled uniformly from the range $[v_{\text{acq-min}}, v_{\text{acq-max}}]$. For each scenario, the total number of imaging requests is randomly sampled from the range $[N_{\min}, N_{\max}]$. Although this specific distribution is used for simulation purposes, the proposed method is general and can accommodate any distribution of strip requests, including those derived from real mission data.

B. Attitude Guidance And Control Model For A Strip Imaging Task Considering A Super-Agile Satellite

The Earth-observing satellite is modeled as a small spacecraft with mass m and inertia I, operating in a fixed low-altitude circular orbit around Earth (radius $\mathbf{r}_{\text{Earth}}$) with inclination i and altitude a. It is equipped with a body-fixed scan-line camera for imaging purposes. The scan line sensor consists of a rectangular surface containing several rows of photodiodes. Attitude control is provided by a three-axis reaction wheel assembly, with wheels aligned along the spacecraft's principal body axes. The satellite is super-agile, capable of slewing simultaneously about all three axes—roll, pitch, and yaw—while actively imaging.

Compared to point-imaging tasks, strip-imaging tasks complicate the attitude guidance model by requiring a specific scanning direction at a desired scanning speed. While snapshot instruments permit the orientation around the view

direction to remain unspecified during target observation, scan line cameras demand that the scan line sensor aligns perpendicularly with the desired scan path. As a result, point-imaging retains one degree of freedom, while strip-imaging requires a fully constrained attitude.

To compute the attitude and rate references for strip imaging operations, three primary reference frames are used: the Earth-Centered Inertial frame N, the Earth-Fixed frame E, and the spacecraft Body frame B. The upper-left superscript specifies the reference frame in which the vector is represented. The unit vectors of the body frame $\{\mathbf{O}_B; \hat{\mathbf{b}}_x, \hat{\mathbf{b}}_y, \hat{\mathbf{b}}_z\}$ are defined with origin $\mathbf{O}_B$ at the center of the rectangular scan-line sensor, which lies in the plane P_{sensor}. The body frame is visualized in Figure 1, and its unit vectors are defined as follows:

- $\hat{\mathbf{b}}_z$ is the boresight axis of the imager. It is defined as the vector normal to P_{sensor} and passing through $\mathbf{O}_B$.
- $\hat{\mathbf{b}}_y$ is the cross-track axis, defined as the vector lying in P_{sensor}, aligned with the direction of the rows of photodiodes, and passing through $\mathbf{O}_B$. It is oriented to maintain a right-handed body frame.
- $\hat{\mathbf{b}}_x$ is the third body axis, introduced to complete the right-handed orthonormal frame. It is given by

$$\hat{\mathbf{b}}_x = \hat{\mathbf{b}}_y \times \hat{\mathbf{b}}_z.$$

The primary attitude requirement is to steer the boresight axis $\hat{\mathbf{b}}_z$ towards a virtual ground target $\mathbf{r}_{\text{target}}$, which moves along the strip's central line — from $\mathbf{r}_{\text{start}}$ to $\mathbf{r}_{\text{end}}$ — with a velocity $\mathbf{v}_{\text{acq}}$ of constant magnitude v_{acq}. The line-of-sight (LOS) vector pointing from the spacecraft to the target is given by:

$$\mathbf{r}_{LS} = \mathbf{r}_{\text{target}} - \mathbf{r}_B, \tag{1}$$

where $\mathbf{r}_B$ is the position of the spacecraft. The principal rotation angle to satisfy the primary attitude requirement is:

$$\phi_1 = \arccos\left(\hat{\mathbf{b}}_z \cdot {}^B\hat{\mathbf{r}}_{LS}\right). \tag{2}$$

The rotation error axis ${}^B\hat{\mathbf{e}}_1$ is defined as:

$${}^B\hat{\mathbf{e}}_1 = \begin{cases} \hat{\mathbf{b}}_y, & |\phi_1| < \epsilon \quad \text{or} \quad |\phi_1 - \pi| < \epsilon, \\ \dfrac{\hat{\mathbf{b}}_z \times {}^B\hat{\mathbf{r}}_{LS}}{\|\hat{\mathbf{b}}_z \times {}^B\hat{\mathbf{r}}_{LS}\|}, & \text{otherwise,} \end{cases} \tag{3}$$

where $\epsilon > 0$ is a small threshold introduced to avoid numerical instabilities. The attitude error between the current body frame B and the desired reference frame $R_1 = \{\mathbf{O}_B; \hat{\mathbf{q}}_x, \hat{\mathbf{q}}_y, \hat{\mathbf{q}}_z\}$ is expressed in Modified Rodrigues Parameters (MRPs) [14] vector as:

$$\boldsymbol{\sigma}_{BR_1} = -\tan\left(\frac{\phi_1}{4}\right) {}^B\hat{\mathbf{e}}_1. \tag{4}$$

The corresponding reference attitude, which represents the orientation of frame R_1 relative to the inertial frame N, is obtained through MRP composition:

$$\boldsymbol{\sigma}_{R_1N} = \boldsymbol{\sigma}_{BN} \oplus (-\boldsymbol{\sigma}_{BR_1}), \tag{5}$$

where $\oplus$ denotes the MRP addition operation including the shadow set switch to keep the result within the principal MRP domain.

Once the first attitude reference requirement is met, the second requirement ensures that the reference cross track axis remains perpendicular to the desired scan path. This is achieved by applying a corrective rotation about the reference boresight axis, resulting in a final attitude reference frame R_2. To ensure valid scan geometry, the scan path must lie in the sensor plane—orthogonal to the boresight axis—since the perpendicularity between the reference cross-track axis and the scan path is only meaningful when both vectors lie within the same plane. To remove any component of $\hat{\mathbf{v}}_{\text{acq}}$ aligned with the reference boresight axis $\hat{\mathbf{r}}_{LS}$, this vector is projected onto the plane normal to $\hat{\mathbf{r}}_{LS}$:

$$\hat{\mathbf{v}}_{\text{acq}\perp} = \hat{\mathbf{v}}_{\text{acq}} - \left(\hat{\mathbf{v}}_{\text{acq}} \cdot \hat{\mathbf{r}}_{LS}\right) \hat{\mathbf{r}}_{LS}. \tag{6}$$

The reference cross-track axis is given by the normalized vector:

$$\hat{\mathbf{r}}_\perp = \begin{cases} \hat{\mathbf{v}}_t = \dfrac{\hat{\mathbf{r}}_{\text{target}} \times \hat{\mathbf{v}}_{\text{acq}}}{\left\|\hat{\mathbf{r}}_{\text{target}} \times \hat{\mathbf{v}}_{\text{acq}}\right\|}, & \text{if } \left\|\hat{\mathbf{v}}_{\text{acq}\perp}\right\| < \epsilon \\ \dfrac{\hat{\mathbf{r}}_{LS} \times \hat{\mathbf{v}}_{\text{acq}\perp}}{\left\|\hat{\mathbf{r}}_{LS} \times \hat{\mathbf{v}}_{\text{acq}\perp}\right\|}, & \text{otherwise} \end{cases} \tag{7}$$

where $\hat{\mathbf{v}}_t$ is a unit vector lying in the plane tangent to the Earth's surface at the virtual target location $\hat{\mathbf{r}}_{\text{target}}$ and perpendicular to the velocity vector $\hat{\mathbf{v}}_{\text{acq}}$. The rotation angle ϕ_2 required to align the current cross-track axis $\hat{\mathbf{q}}_y$ with $\hat{\mathbf{r}}_\perp$ is computed as:

$$\phi_2 = \text{sign}\left[-\left(\hat{\mathbf{q}}_y \times {}^{R_1}\hat{\mathbf{r}}_\perp\right)_z\right] \arccos\left(\hat{\mathbf{q}}_y \cdot {}^{R_1}\hat{\mathbf{r}}_\perp\right). \tag{8}$$

This corrective rotation about the boresight axis ${}^{R_1}\hat{\mathbf{r}}_{LS}$ is represented as an MRP set:

$$\boldsymbol{\sigma}_{R_2R_1} = -\tan\left(\frac{\phi_2}{4}\right) {}^{R_1}\hat{\mathbf{r}}_{LS}. \tag{9}$$

Finally, the overall reference attitude $\boldsymbol{\sigma}_{R_2N}$, expressed relative to the inertial frame N, is obtained by composing the first reference attitude $\boldsymbol{\sigma}_{R_1N}$ with this corrective rotation:

$$\boldsymbol{\sigma}_{R_2N} = \boldsymbol{\sigma}_{R_1N} \oplus \boldsymbol{\sigma}_{R_2R_1}. \tag{10}$$

The spacecraft's attitude error relative to this final reference is then:

$$\boldsymbol{\sigma}_{BR_2} = \boldsymbol{\sigma}_{BN} \ominus \boldsymbol{\sigma}_{R_2N}, \tag{11}$$

where $\ominus$ denotes MRP subtraction including the shadow set switch to keep the result within the principal MRP.

The tracking error rate $\dot{\boldsymbol{\sigma}}_{BR_2}$ is computed via numerical differentiation of the attitude error $\boldsymbol{\sigma}_{BR_2}$ over time. When no prior data point is available, numerical differencing is not feasible. In such cases, the error rate is initialized to zero.

Closed-loop attitude control during a strip imaging task is performed using an exponentially stable MRP-based steering controller [15], in combination with rate servos driving the three reaction wheels. The reaction wheels are subject to actuation constraints, with commanded torques limited to a maximum $u_{\max}$. The control system operates at a frequency f and receives, at each control step, the attitude error $\boldsymbol{\sigma}_{BR_2}$ and the attitude error rate $\dot{\boldsymbol{\sigma}}_{BR_2}$.

C. Imaging Requirements

Imaging requests are subject to operational hard constraints, primarily related to geometric visibility and sensor limitations such as minimum illumination requirements and sensor saturation thresholds. This work focuses on the view angle constraint, which ensures that the virtual target $\mathbf{r}_{\text{target}}$ is at any time within the sensor's field of regard, but the method is general enough to account for other constraints. Specifically, during a strip imaging task, the spacecraft must maintain:

$$\left|\angle\left(\mathbf{r}_{\text{LS}}, \hat{\mathbf{r}}_{\text{target}}\right)\right| < \frac{\pi}{2} - \theta_{\min}, \tag{12}$$

where $\mathbf{r}_{\text{LS}}$ is the LOS vector and $\theta_{\min}$ the minimum required elevation angle above the local horizon. Due to the spacecraft's motion and orbital geometry, only specific time intervals are suitable for initiating the imaging of a strip while satisfying this constraint during the entire task. Starting time opportunity windows are defined as the intervals during which the imaging of a given strip can begin such that the entire strip can subsequently be imaged without violating the view angle constraint. Each of these windows is represented as a time interval $[t_{\text{start},1}, t_{\text{start},2}] = w \in \mathcal{W}_i$, where $\mathcal{W}_i$ denotes the set of all feasible starting time opportunity windows for imaging request i.

In addition to operational hard constraints, dynamic performance requirements must also be met to ensure high-quality imaging. Specifically, during a strip imaging task, the spacecraft's attitude must closely track the guidance profile. This is enforced by bounding the attitude error $\boldsymbol{\sigma}_{BR_2}$ during the entire strip imaging task:

$$\|\boldsymbol{\sigma}_{BR_2}\| < \sigma_{\max} \tag{13}$$

where $\sigma_{\max}$ represents the maximum tolerable attitude error.

At the end of a strip imaging task, the operational hard constraints and the dynamic performance requirements for the next strip are not necessarily satisfied. In particular, dynamic performance discrepancies arise because the controller requires a finite transition time to align the spacecraft with the guidance profile. To ensure that these requirements are satisfied from the beginning of the next strip, a transition phase of duration T_{pre} is introduced before each strip imaging task ρ. To incorporate this transition interval into the simulation process, the central line of the strip is interpolated along the Earth's surface prior to the nominal start location $\mathbf{r}_{\text{start}}$, over a distance equivalent to $v_{\text{acq}}T_{\text{pre}}$. This interpolation yields a virtual pre-imaging trajectory that precedes the actual imaging path.

III. Formulation of the Strip Imaging Scheduling Problem

A. Strip Imaging Scheduling Problem

The strip-imaging scheduling problem addressed in this paper seeks to construct an ordered sequence of strip imaging tasks

$$u = (a_1, a_2, \ldots, a_K),$$

where each task a_k takes the form

$$a_k = (\rho, T_{\text{pre}}),$$

with $\rho = (r_{\text{start}}, r_{\text{end}}, p, v_{\text{acq}}) \in R$ specifying an imaging request, and T_{pre} representing a chosen transition duration from $[0, T_{\text{pre-max}}]$.

The objective is to maximize the cumulative sum of the priorities of imaging requests that are successfully completed for the first time within the mission horizon T.

B. Partially Observable semi-Markov Decision Processes

The strip imaging scheduling problem is formalized as a Partially Observable semi-Markov Decision Process [16] (POsMDP) to be solved with reinforcement learning. Although the mission horizon T is finite, it is assumed to be sufficiently long to justify modeling the problem using an infinite-horizon POsMDP formulation.

A POsMDP provides a framework for sequential decision-making in environments with partial observability and variable-duration actions. At each decision step, the environment is in a hidden state $s \in \mathcal{S}$, the agent selects an action $a \in \mathcal{A}$, and the environment transitions to a new state $s' \in \mathcal{S}$ according to the transition probability function $T(s' \mid s, a)$. The duration of this transition is governed by the step-duration function $F(s, a, s')$. The agent receives a scalar reward $r = R(s, a, s')$ that quantifies the immediate benefit or cost of taking action a in state s and arriving at state s'. Because the true state s' is not directly observable, the agent instead receives an observation $o \in O$ drawn from the observation function $Z(o \mid s', a)$.

In a partially observable setting, the RL agent seeks to learn a policy $\pi(a_t \mid h_t)$ that maps a history of observations and actions $h_t = (o_0, a_0, o_1, a_1, \ldots, o_t)$ to a distribution over actions. The objective is to maximize the expected cumulative discounted reward starting from the initial history h_0:

$$V(h_0) = \sum_{t=0}^{\infty} \gamma^{\sum_{i=0}^{t} \Delta t_i} \, r_t, \tag{14}$$

where $\gamma \in [0, 1)$ is the discount factor, Δt_i is the duration of step i, and r_t is the reward obtained at decision step t.

C. POsMDP Formulation For The Strip Imaging Scheduling Problem

The elements of the POsMDP $(\mathcal{S}, \mathcal{A}, T, F, R, O, Z)$ for the strip imaging scheduling problem are defined as follows:

- **State space** $\mathcal{S}$ is the complete space of simulator states required to maintain the Markov assumption. It includes satellite dynamic states, flight software states, and environment states.
- **Action space** $\mathcal{A}$ combines discrete imaging requests with a continuous range of possible transition times that the agent can select at each decision step. To avoid an excessively large action space, not all imaging requests are considered simultaneously. Instead, only the next N unfulfilled requests are included, ordered by the remaining time until the closing of their respective starting time opportunity windows. At each step, the agent selects an action:

  $$a = (r, T_{\text{pre}}) \in \mathcal{A} = [\![1, N]\!] \times [0, T_{\text{pre-max}}],$$

  where r denotes a request index among the selected N unfulfilled requests, and T_{pre} is a transition time chosen from the continuous interval $[0, T_{\text{pre-max}}]$.
- **Transition probability function** T is deterministic and defined by a generative model G such that $G(s, a)$ returns the next state s'. Then,

  $$T(s, a, s') = \begin{cases} 1 & \text{if } s' = G(s, a), \\ 0 & \text{otherwise.} \end{cases}$$

Table 1 Definition of the observation space features for a single look-ahead strip n. The total observation vector consists of these features concatenated for $n = 1 \dots N$.

Parameter	Norm.	Dim.	Description
p_n	-	1	Priority of the unfulfilled request n
${}^{H}\mathbf{r}_{\text{start,n}}$	$\mathbf{r}_{\text{Earth}}$	3	Starting point of the unfulfilled request n in the Hill frame
${}^{H}\mathbf{r}_{\text{end,n}}$	$\mathbf{r}_{\text{Earth}}$	3	Ending point of the unfulfilled request n in the Hill frame
l_n	$L_{\max}$	1	Length of request n
$v_{\text{acq},n}$	$v_{\text{acq,max}}$	1	Required acquisition speed for request n
$\boldsymbol{\theta}_{BR_2,\,n}$	π rad	1	Principal rotation angle between B and R_2 if the spacecraft starts imaging without pre-imaging for request n
$\lVert\dot{\boldsymbol{\sigma}}_{BR_2,n}\rVert$	0.05 rad/s	1	MRP attitude rate error norm between B and R_2 if the spacecraft starts imaging without pre-imaging for request n
$w_{\text{relative},n}$	300 s	2	Next starting time opportunity window for request n expressed relative to the current simulation time

- **Step-duration function** F is deterministic and depends solely on the chosen action a. Specifically, F maps each action to its associated execution time, defined by

$$F\big(a = (r,\, T_{\text{pre}})\big) = T_{\text{pre}} + T_{\text{acq},r},$$

  where $T_{\text{acq},r}$ is the time required to image request ρ_r excluding the transition phase.
- **Reward function** R yields the priority of the request if it is fulfilled for the first time, and zero otherwise:

$$R\big(s,\, a = (r,\, T_{\text{pre}}),\, s'\big) = \begin{cases} p_r & \text{if } \rho_r \in \mathcal{U}(s), \text{ and } \rho_r \in \mathcal{F}(s'), \\ 0 & \text{otherwise,} \end{cases}$$

  where p_r is the priority of image request ρ_r.
- **Observation space** O detailed in Table 1 is constructed by selecting and transforming relevant dimensions from the full state space, guided by expert knowledge and ablation studies. It includes key information about the spacecraft and the next N upcoming unfulfilled requests. These observations are limited to data that the satellite can reasonably obtain onboard with minimal uncertainty, supporting reliable closed-loop decision-making. All observation elements are normalized to approximately lie within the range $[-1, 1]$ to enhance the performance of RL algorithms.
- **Observation function** Z is deterministic since the satellite is assumed to observe the observation space perfectly.

The POsMDP and underlying generative model are implemented in BSK-RL*, a modular, open-source package for creating spacecraft tasking RL environments. BSK-RL uses the standard Gymnasium API for RL environments, making the package compatible with all major RL frameworks. Internally, the spacecraft and environment dynamics are modeled in the Basilisk† spacecraft simulation framework.

IV. Reinforcement Learning Framework Tailored to Strip Scheduling

This framework is referred to as HOP-PPO, a Hybrid Observation task Planning–oriented Proximal Policy Optimization framework.

*https://github.com/AVSLab/bsk_rl
†https://github.com/AVSLab/basilisk

A. Proximal Policy Optimization

Proximal Policy Optimization [10] (PPO) is an actor-critic RL algorithm in which the actor learns a policy $\pi_\theta(a_t \mid h_t)$ while the critic estimates the state-value function $V_\phi(h_t)$, representing the expected return from a given history. The critic's estimates are used to compute an advantage function $\hat{A}_t$, which quantifies the discrepancy between the observed rewards and the critic's predictions. In PPO, the advantage is commonly estimated using Generalized Advantage Estimation [17] (GAE):

$$\hat{A}_t^{\mathrm{GAE}} = \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\phi(h_{t+1}) - V_\phi(h_t), \tag{15}$$

where γ is the discount factor and $\lambda \in [0, 1]$ controls the bias–variance tradeoff.

The stochastic policy π_θ is optimized via stochastic gradient ascent by maximizing the following clipped surrogate objective:

$$L_{\mathrm{CLIP}}(\theta) = \mathbb{E}_t \Big[\min \left(r_t(\theta)\hat{A}_t^{\mathrm{GAE}}, \, \mathrm{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t^{\mathrm{GAE}} \right) \Big], \tag{16}$$

where the probability ratio

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid h_t)}{\pi_{\mathrm{old}}(a_t \mid h_t)}, \tag{17}$$

measures the deviation of the current policy from the one used to collect the data, and ϵ is a clipping hyperparameter that stabilizes training. By combining the critic's value estimates with the actor's policy updates, PPO achieves stable and efficient learning within a stochastic gradient actor–critic framework.

Building on prior work in point imaging scheduling, the full history h_t is simplified by the current observation o_t to reduce problem complexity, under the assumption that o_t contains sufficient information to enable near-optimal decisions for the Strip Imaging Scheduling Problem.

B. Hybrid-Action Proximal Policy Optimization Architecture For Strip Scheduling

In standard RL, the policy π_θ handles either discrete action spaces, modeled via a Categorical distribution, or continuous action spaces, modeled via a Squashed Gaussian distribution. In the Strip Scheduling Problem, however, the agent must output both a discrete action and a continuous action, with the later conditioned on the selected discrete choice. To accommodate this, a hybrid architecture with two actors, illustrated in Figure 2, is proposed:

- The Discrete Actor π_{θ_D} receives the observation o_t and outputs a discrete action $a_{d,t}$, representing the choice of which strip to image. The output is modeled as a categorical distribution over the available strips:

$$a_{d,t} \sim \pi_{\theta_d}(a_{d,t} \mid o_t). \tag{18}$$

- The Continuous Actor π_{θ_C} produces a continuous action $a_{c,t}$ specifying the necessary transition time before imaging the selected strip. Its input is a subset of the observation o_t containing only the information relevant to estimate the transition time of the chosen strip $a_{d,t}$, denoted $S(o_t, a_{d,t})$. The output is modeled as a Squashed Gaussian distribution to ensure bounded continuous actions:

$$a_{c,t} \sim \pi_{\theta_c}(a_{c,t} \mid S(o_t, a_{d,t})). \tag{19}$$

To integrate both actors into a single decision-making mechanism, they are trained with a shared critic and a unified surrogate objective. This objective is evaluated under the joint policy:

$$\pi_\theta(a_t \mid o_t) = \pi_{\theta_d}(a_{d,t} \mid o_t) \, \pi_{\theta_c}(a_{c,t} \mid S(o_t, a_{d,t})). \tag{20}$$

with $a_t = (a_{d,t}, a_{c,t})$ and $\theta = (\theta_d, \theta_c)$.

In the proposed architecture, both actors and the shared critic are implemented as multi-layer perceptrons [18] (MLPs).

C. Observation Encoding With Self-Attention Layers

In the field of robotics, attention mechanisms have been successfully applied to task scheduling problems, particularly in scenarios where capturing contextual relationships between tasks and robots is crucial [19, 20]. This idea is adapted to the strip imaging scheduling problem as follows.

The raw observation vector o_t is formed by concatenating each strip's feature group. The groups are ordered according to the remaining time until the closing of the corresponding strip's opportunity window. In the proposed encoder, each strip is treated as a token. Each token $x_n \in \mathbb{R}^{d_{emb}}$ is obtained by projecting the strip's feature group into an embedding space of dimension d_{emb}. To capture both the relative and absolute positions of the feature groups in the raw observation vector, a sinusoidal positional encoding [21] is added to each token. This is defined component-wise as :

$$PE_{n,2i} = \sin\left(\frac{n}{10000^{2i/d_{\text{emb}}}}\right), \quad PE_{n,2i+1} = \cos\left(\frac{n}{10000^{2i/d_{\text{emb}}}}\right), \tag{21}$$

for $i = 0, 1, \ldots, d_{\text{emb}}/2 - 1$, where n is the strip index in the discrete part of the action space. The final token representation with positional encoding is :

$$z_{\text{n}} = x_{\text{n}} + PE_{\text{n}}. \tag{22}$$

After incorporating positional encoding, the tokens are passed through a Transformer [21] encoder using the implementation from PyTorch[22]. This encoder is composed of N_L stacked layers, each combining layer normalization, multi-head self-attention, and feedforward networks. The output is a sequence of tokens that are no longer independent but contextually informed by the other tokens in the sequence. The dimensionality of the tokens remains unchanged.

The resulting context-aware tokens serve as input to both the shared critic and the discrete actor. In contrast, the continuous actor relies solely on the raw features of the selected strip and does not use these tokens as input, since it does not perform comparisons across strips.

The overall architecture of the observation encoder is depicted in Figure 2.

D. Duration-Aware Advantage Function With Infinite-Horizon Bootstrapping

In the strip imaging scheduling problem, actions occur over variable time intervals, in accordance to the POsMDP formulation. To address this, the GAE function [Eq. 15] is adapted to account for the elapsed time between decisions [23] :

$$\hat{A}_t^{\text{GAE,semi}} = \sum_{i=0}^{\infty} \lambda^i \gamma^{\sum_{j=0}^{i} \Delta_{t+j}} \left(r_{t+i} + \gamma^{\Delta_{t+i+1}} V(h_{t+i+1}) - V(h_{t+i}) \right), \tag{23}$$

where Δ_{t+j} is the time duration between steps $t + j$ and $t + j + 1$. This formulation ensures that rewards and value estimates are discounted according to the actual time elapsed rather than the number of steps.

An infinite time horizon formulation is considered, as the overall mission duration T is assumed to be large. Training is performed using truncated episodes for tractability, while the value function beyond the truncation point is estimated using the default RLlib [24] bootstrapping method. This approach allows the GAE to incorporate information about rewards and state values beyond the immediate horizon, while still leveraging finite-length episodes for efficient training.

Table 2 Simulation Parameters

Parameter	Value	Parameter	Value
Spacecraft Properties		**Imaging Requirements**	
(a, i, e)	(520 km, 45°, 0)	$\sigma_{\max}$	0.1
m	330 kg	$\theta_{\min}$	10°
I	$[82.1, 98.4, 121.0]$ kg · m^2	**Request Properties**	
$u_{\max}$	0.3 N m	$[L_{\min}, L_{\max}]$	[100, 500] km
Steering Controller		$[v_{\text{acq,min}}, v_{\text{acq,max}}]$	[2, 4] km/s
$(K_1, K_3, \omega_{\max})$	(0.25, 3, 5 rad/s)	$[N_{\min}, N_{\max}]$	[1000, 5000]
$(K_{i,\text{servo}}, P_{\text{servo}})$	(5.0, 30)		

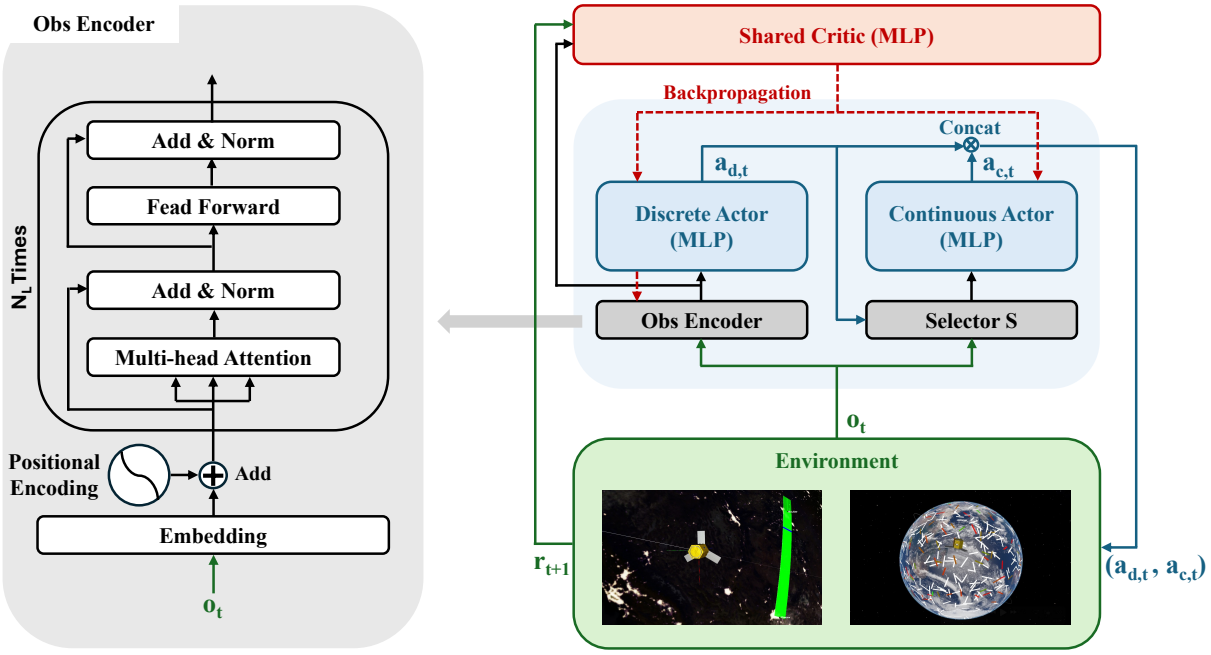


Fig. 2 HOP-PPO Framework for Strip Imaging Scheduling.

V. Numerical Performance Analysis

A. Baseline Methods

The HOP-PPO RL Policy is evaluated against several baseline methods:

- **A Random Policy**: At each step, this policy chooses a strip uniformly at random from the next N upcoming strips and selects a transition time drawn uniformly from $[0, T_{\text{pre-max}}]$.
- **An Heuristic Policy**: A transition time period T_{heur}, fixed for all strip imaging tasks, is picked from the interval $[0, T_{\text{pre-max}}]$. The policy selects all strip imaging requests ρ for which there exists a starting time opening-window $w(\rho)$ such that $T_{\text{heur}} \in w(\rho)$. The resulting set of selected strips is denoted by $\mathcal{S}_{\text{feasible}}$. Among these feasible strips, the policy chooses the request ρ that maximizes the ratio of its priority $p(\rho)$ to its duration $T_{\text{acq}}(\rho) + T_{\text{heur}}$:

$$\rho^{\star} = \arg \max_{\rho \in \mathcal{S}_{\text{feasible}}} \frac{p(\rho)}{T_{\text{acq}}(\rho) + T_{\text{heur}}}.$$

  With this heuristic, a task never fails due to the view-angle constraint; failures occur only when T_{heur} is not large enough to meet the dynamic performance requirements. This heuristic is evaluated for multiple fixed values of T_{heur}. The results are reported both for the optimal fixed transition value $T^{\star}_{\text{heur}}$ which maximizes cumulative reward, and for $T_{\text{heur}} = T_{\text{pre-max}}$. Here, $T_{\text{pre-max}}$ is a practically accessible upper bound, chosen as the minimal transition time for which more than 99% of the strips can fulfill the dynamic performance requirements.
- **A Discrete RL PPO Policy**: The continuous part of the hybrid action space, corresponding to $[0, T_{\text{pre-max}}]$, is discretized into d evenly spaced values. This yields a purely discrete action space of size $N \times d$. A standard actor–critic architecture with one MLP discrete actor and one MLP critic is then used to learn a policy over this expanded action set. Several discretization granularities are tested over the set $\mathcal{G} = \{15\text{s}, 30\text{s}, 60\text{s}\}$. The performances of both the best-performing granularity, denoted by $g^{\star}$, and the worst-performing granularity, denoted by g^{-}, after 5 million environment steps are reported.
- **A H-PPO[13] RL Policy**: The standard hybrid PPO extension in the litterature, but not specifically tailored to the structure of the Strip Imaging Scheduling Problem. This architecture uses two actors—a discrete MLP actor selecting the discrete action component and a continuous MLP actor selecting continuous parameters for all possible discrete choices. The final action sent to the environment combines the chosen discrete task with its associated continuous parameter. Both actors share a common MLP observation encoder, which is not shared with the critic. A single MLP critic is used, with separate objectives optimized for each actor.

- **A HOP-PPO RL Policy with MLP Encoder**: This baseline is the same as HOP-PPO, but replaces the transformer-based observation encoder with a MLP. This allows isolation of the effects of the transformer-based observation encoder and the overall hybrid architecture.

All MLP-based neural network components in the RL methods (actors, critics, and observation encoders) are implemented as 2-layer MLPs, with neurons optimized from $\{512, 1024, 2048, 4096\}$. For HOP-PPO, the transformer-based observation encoder is tuned to maximize performance while ensuring that its total number of parameters does not exceed that of a two-layer MLP with 4096 units per layer. All hyperparameters, aside from the network architectures described above, are kept the same across all RL-based methods.

Table 3 Training Hyperparameters

Common Training Parameters			
Number of workers	32	GAE parameter (λ)	0.95
Learning rate	3×10^{-5}	Gradient clipping	0.2
Discount factor (γ)	0.999	PPO clipping	0.5
Training batch size	3000	SGD iterations	10
N	40		

Network Architectures

Method	Shared Encoder	Actor MLP Units		Critic MLP Units
		Discrete	Continuous	
Discrete	None	[2048, 2048]	N/A	[2048, 2048]
HOP-PPO	Transformer† (Dim: 128, Heads: 8, Layers: 4)	[1024, 1024]	[1024, 1024]	[1024, 1024]
HOP-PPO MLP	MLP [2048, 2048]†	[1024, 1024]	[1024, 1024]	[1024, 1024]
H-PPO	MLP [2048, 2048]‡	[1024, 1024]	[1024, 1024]	[2048, 2048]

† Encoder shared by Critic and Discrete Actor.

‡ Encoder shared by Discrete Actor and Continuous Actor.

B. Comparative Performance Analysis of HOP-PPO and Baseline Methods

The training environment for all RL-based methods was configured using the simulation parameters listed in Table 2 and the training hyperparameters in Table 3, with unlisted values defaulting to the standard settings in BSK-RL v1.2.0 and RLlib v2.6.3. Training was conducted on the University of Colorado's Research Computing (CURC) infrastructure using 32 cores. All RL-based methods were trained for 3 million steps, except for the Discrete PPO method and its different granularities, which required 5 million steps to converge.

To assess the stability of the training, each method was trained three times with independently generated random seeds. Figure 3 shows the mean and standard deviation of the three training curves for each method, with each training curve representing the cumulative reward over the truncated episodes (6 orbits) as a function of the training steps. The mean convergence value over the three runs is reported along with its standard deviation, defining the convergence value of a training curve as the mean cumulative reward over the last 0.5 million steps. For all RL methods, the mean training curves exhibit clear performance improvement prior to reaching convergence. The variability across the three independent training runs remains low. In particular, the coefficient of variation—computed as the ratio of the standard deviation to the mean convergence value—remains below 2% for all RL methods, indicating consistent convergence characteristics across random seeds.

To benchmark the performance of the RL methods against the baseline heuristics, each policy is evaluated in a 24-hour Earth-observation mission scenario (15 orbits). For every evaluation episode, the initial number of strip-imaging

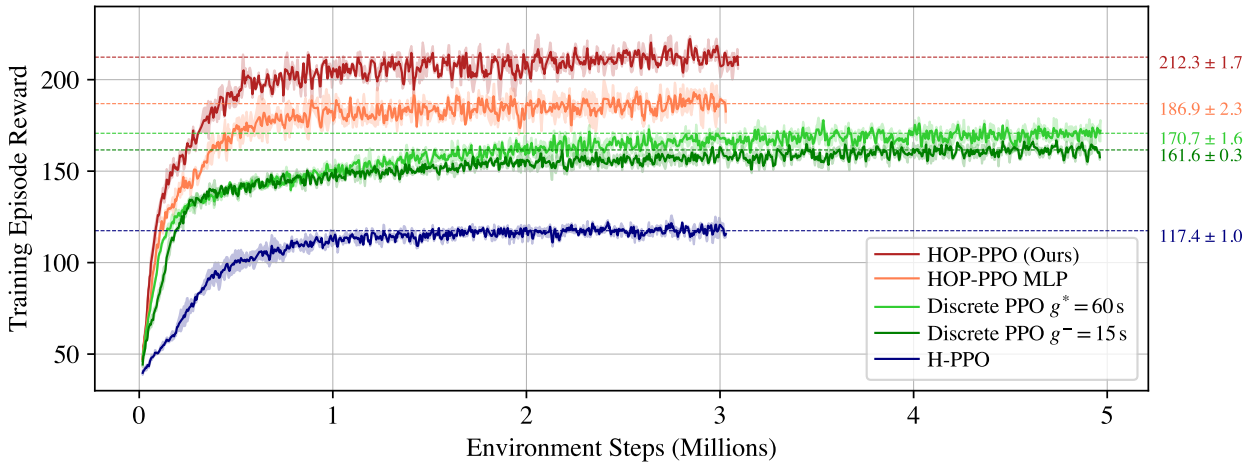


Fig. 3 Training Curves Averaged Over 3 Runs for RL Methods with Shaded Standard Deviation.

targets is uniformly sampled between 2,000 and 5,000. As the mission progresses, no new targets are introduced, requiring the satellites to adapt to the gradually decreasing number of available targets. All other scenario parameters not explicitly mentioned are kept identical to those used during training. Performance is quantified in terms of the mission reward obtained per orbit. For each method, performance is evaluated over 100 episodes generated using a fixed set of independently sampled random seeds. The resulting performance distributions are summarized in Figure 4.

For the heuristic baseline, results are reported for two fixed transition-time configurations: $T_{\text{pre-max}}$ and the reward-maximizing value T^*. A high transition time, $T_{\text{pre-max}}$, ensures successful imaging for at least 99% of the strip imaging tasks attempted by the heuristic policy, but at the expense of extended transitions for most of the tasks. By evaluating a range of fixed transition times $T_{heur,k} = 10k$ s, $k = 1, \ldots, 12$, the reward-maximizing value is determined to be $T^* = 80$ s. Although this setting achieves only a 81% success rate, it results in a higher mean cumulative reward. The heuristic policy T^* is subsequently used as a benchmark for evaluating the RL policies.

All RL policies outperform the Heuristic T^*, except H-PPO. This weaker performance arises from a mismatch between the H-PPO architecture and the structure of the Strip Imaging Scheduling problem. First, the environment features a hybrid action space, whose discrete component is large ($N = 40$). In this setting, the H-PPO continuous actor estimates a Squashed Gaussian distribution for each possible discrete action, producing $2N$ outputs. H-PPO does not scale well with large discrete action space and fails to exploit the structural similarities across discrete tasks in this problem, particularly when computing the continuous parameter for each discrete action. Second, H-PPO employs a shared observation encoder for both the continuous and discrete actors, which does not align well with the structure of this problem. The discrete actor requires a global overview of all available strips to effectively compare them, whereas the continuous actor only depends on the independent features of each strip and does not perform such comparisons.

The Discrete PPO method was evaluated with different granularities $\mathcal{G} = \{15\text{s}, 30\text{s}, 60\text{s}\}$. The best performance is achieved with $g^* = 60$ s, yielding a 10.4% improvement in mean cumulative reward over the Heuristic T^*. The worst performance occurs with $g^- = 15$ s , achieving only a 6% improvement. Decreasing the granularity, from 60 s to 15 s, which enlarges the action space from $3N$ to $9N$, harms performance. This suggests that discretization in this problem inherently leads to suboptimal solutions, as a coarse granularity must be maintained to control the size of the action space.

The HOP-PPO MLP method achieves a 20.7% improvement in mean cumulative reward over the Heuristic T^*, outperforming both Discrete PPO and H-PPO. This result isolates the actor–critic architectural benefits of the HOP-PPO framework, not considering the advantages of the transformer-based observation encoder, which is replaced in this variant by a simple MLP. Compared to Discrete PPO, HOP-PPO employs two separate actors—one discrete and one continuous—avoiding the need to discretize the continuous part of the hybrid action space and consequently to deal with large action spaces. Compared to H-PPO, HOP-PPO further exploits the structural similarities between strip imaging tasks, with a continuous actor that estimates a single Squashed Gaussian distribution for the selected strip. Additionally, the discrete and continuous actors do not share an observation encoder, allowing each to specialize on distinct input features: the discrete actor focuses on comparing strips globally, while the continuous actor processes only the features relevant to the chosen strip.

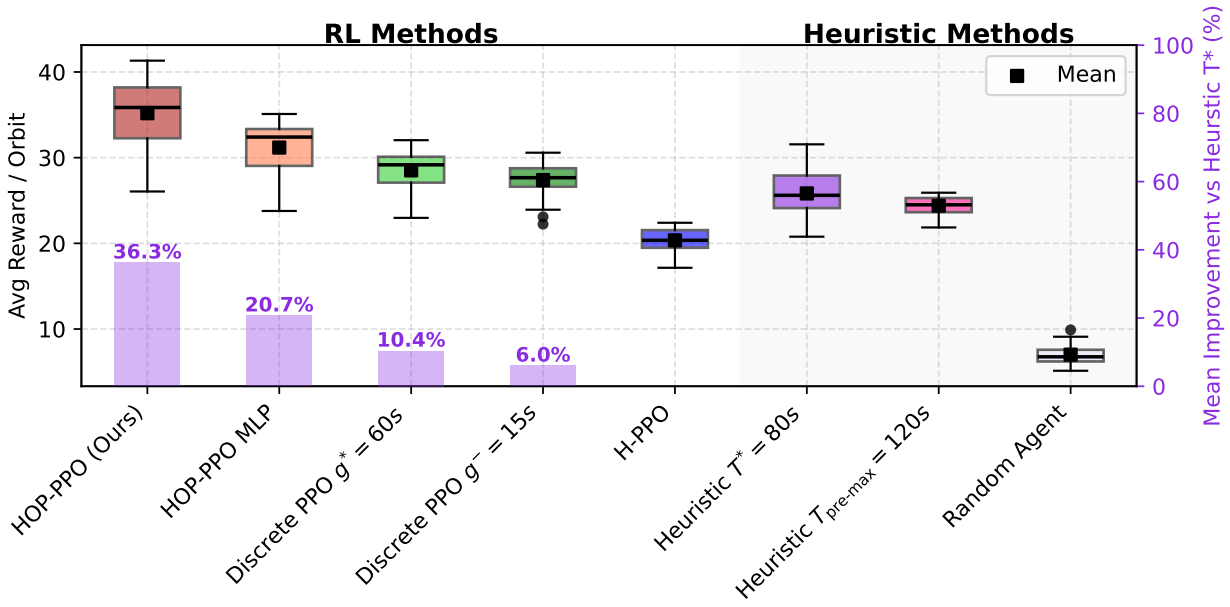


Fig. 4 Performance Comparison over a 24-Hour Earth-Observation Mission Scenario.

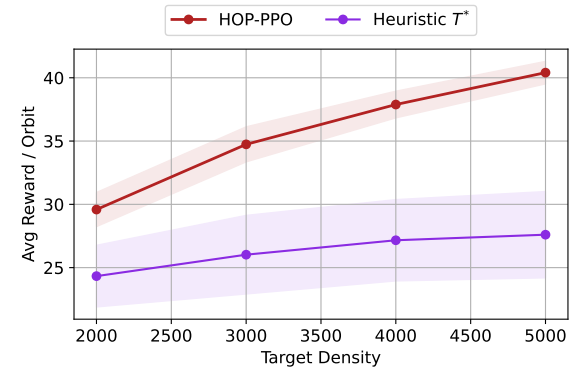
The HOP-PPO method achieves the best performance among all approaches, with a 36.3% increase in mean cumulative reward over the heuristic T^*. In particular, the 15.6% improvement over HOP-PPO MLP demonstrates the benefits of a transformer-based observation encoder within the proposed actor–critic architecture. The transformer-based encoder enables the model to focus on the most relevant strip tokens by integrating, in each of them, information on their contextual importance. This approach allows for more precise assessment and selection of the optimal strip, providing a clear advantage over a traditional MLP encoder.

C. Evaluation of HOP-PPO Policy Behavior Compared to a Baseline Heuristic

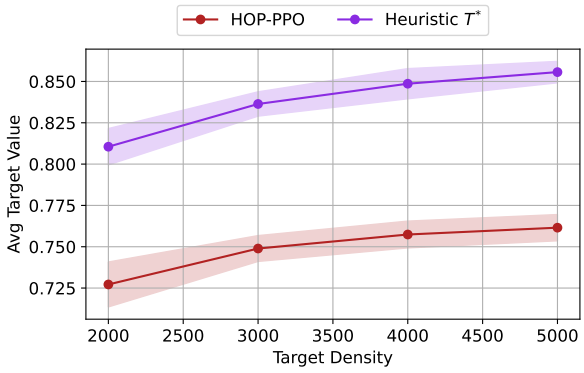
The HOP-PPO policy's behavior is analyzed and compared against the best-performing baseline heuristic T^* in the same 24-hour Earth-Observation Mission scenario described in the previous section, under different initial request densities [2000, 3000, 4000, 5000]. For each combination of policy and request density, a set of 30 test cases generated using different random seeds is executed to ensure statistically reliable results.

Figure 5 reports, for each target density, the average total reward per orbit (with standard deviation), the average target value (with standard deviation), the average task duration, and the average number of attempts per orbit. For both policies, the average cumulative reward per orbit increases as the request density grows. This behavior aligns with expectations as a higher density provides a broader set of requests that satisfy the view-angle constraint at an earlier stage. With more options available, the policies become increasingly selective. However, the way this selectivity is expressed varies across policies.

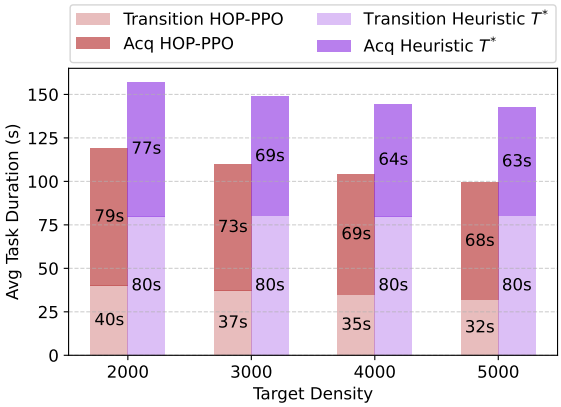
The heuristic T^* prioritizes strips with higher values and shorter acquisition times to allow for more attempts, as density increases. HOP-PPO also tends to select higher-value targets, but target value is not the primary focus, as it consistently achieves lower average target values than the heuristic. It improves performance by reducing the total task duration, which includes both the acquisition time and the transition time between strips. Importantly, the average transition time of HOP-PPO is 50% of the heuristic's transition time in low-density scenarios and 40% in high-density scenarios. This substantially shorter transition time allows HOP-PPO to image significantly more strips per orbit—35% more in low-density settings and 64% more in high-density ones. Overall, by balancing transition time, acquisition time, and target priority, HOP-PPO consistently achieves higher reward per orbit than the heuristic, with particularly pronounced gains at high request densities.



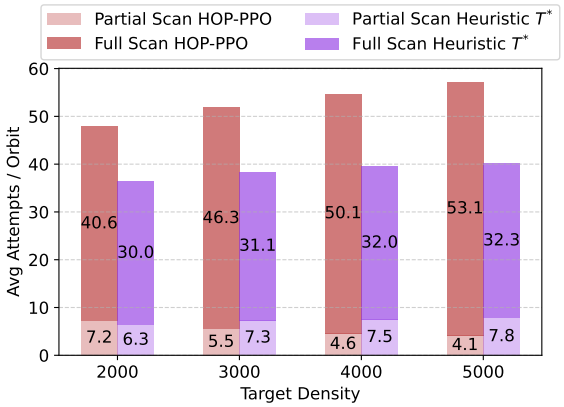
(a) Average Total Reward per Orbit.



(b) Average Target Value.



(c) Average Task Duration.



(d) Average Attempts per Orbit.

Fig. 5 HOP-PPO vs. Heuristic T^* Behavior Across Varying Initial Target Densities.

VI. Conclusion

This paper demonstrates the potential of Deep Reinforcement Learning (DRL) for the Strip Imaging Scheduling Problem considering Super-Agile Earth Observing Satellites. The proposed approach models the problem as a Partially Observable semi-Markov Decision Process and introduces the dual-actor, transformer-based HOP-PPO framework to efficiently handle strip imaging task selection and transition time estimation. Experimental results show that HOP-PPO outperforms existing hybrid DRL methods by at least 25.9% in cumulative reward and improves upon traditional heuristics by 36.3%. Further analysis of the policy reveals that it reduces transition durations by roughly 50% compared to heuristics, enabling a 35–64% increase in the number of imaged strips. Future work will extend the framework to incorporate resource management, including battery usage, reaction wheel desaturation, and data downlinking.

References

[1] Chien, S., Cichy, B., Davies, A., Tran, D., Rabideau, G., Castano, R., Sherwood, R., Mandl, D., Frye, S., Shulman, S., et al., "An autonomous earth-observing sensorweb," *IEEE Intelligent Systems*, Vol. 20, No. 3, 2005, pp. 16–24.

[2] Salomonson, V. V., Barnes, W., Maymon, P. W., Montgomery, H. E., and Ostrow, H., "MODIS: Advanced facility instrument for studies of the Earth as a system," *IEEE Transactions on geoscience and remote sensing*, Vol. 27, No. 2, 1989, pp. 145–153.

[3] Spangelo, S., Cutler, J., Gilson, K., and Cohn, A., "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers & Operations Research*, Vol. 57, 2015, pp. 1–16.

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

[5] Cappaert, J., Foston, F., Heras, P. S., King, B., Pascucci, N., Reilly, J., Brown, C., Pitzo, J., and Tallhamm, M., "Constellation modelling, performance prediction and operations management for the spire constellation," 2021.

[6] Shah, V., Vittaldev, V., Stepan, L., and Foster, C., "Scheduling the world's largest earth-observing fleet of medium-resolution imaging satellites," *International Workshop on Planning and Scheduling for Space*, Organization for the 2019 International Workshop on Planning and Scheduling, 2019, pp. 156–161.

[7] Herrmann, A., and Schaub, H., "Reinforcement learning for the agile earth-observing satellite scheduling problem," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 59, No. 5, 2023, pp. 5235–5247.

[8] Herrmann, A., Carneiro, J. a. V., and Schaub, H., "Reinforcement Learning for The Multi-Satellite Earth-Observing Scheduling Problem," *Proceedings of the 44th Annual American Astronautical Society Guidance, Navigation, and Control Conference, 2022*, Springer, 2022, pp. 1351–1368.

[9] Herrmann, A., Stephenson, M. A., and Schaub, H., "Single-Agent Reinforcement Learning for Scalable Earth-Observing Satellite Constellation Operations," *Journal of Spacecraft and Rockets*, Vol. 61, No. 1, 2024, pp. 114–132.

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

[11] Sherstov, A. A., and Stone, P., "Function approximation via tile coding: Automating parameter choice," *International symposium on abstraction, reformulation, and approximation*, Springer, 2005, pp. 194–205.

[12] Herrmann, A., and Schaub, H., "A comparative analysis of reinforcement learning algorithms for earth-observing satellite scheduling," *Frontiers in Space Technologies*, Vol. 4, 2023, p. 1263489.

[13] Fan, Z., Su, R., Zhang, W., and Yu, Y., "Hybrid actor-critic reinforcement learning in parameterized action space," *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 2279–2285.

[14] Schaub, H., and Junkins, J. L., *Analytical mechanics of space systems*, Aiaa, 2003.

[15] Schaub, H., and Piggott, S., "Speed-constrained three-axes attitude control using kinematic steering," *Acta Astronautica*, Vol. 147, 2018, pp. 1–8.

[16] Sondik, E. J., *The optimal control of partially observable Markov processes*, Stanford University, 1971.

[17] Schulman, J., Moritz, P., Levine, S., Jordan, M., and Abbeel, P., "High-dimensional continuous control using generalized advantage estimation," *arXiv preprint arXiv:1506.02438*, 2015.

[18] Rumelhart, D. E., Hinton, G. E., and Williams, R. J., "Learning representations by back-propagating errors," *nature*, Vol. 323, No. 6088, 1986, pp. 533–536.

[19] Dai, W., Rai, U., Chiun, J., Yuhong, C., and Sartoretti, G., "Heterogeneous multi-robot task allocation and scheduling via reinforcement learning," *IEEE Robotics and Automation Letters*, 2025.

[20] Bichler, J., Matoses Gimenez, A., and Alonso-Mora, J., "SADCHER: Scheduling using Attention-based Dynamic Coalitions of Heterogeneous Robots in Real-Time," *Proceedings of the IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2025.

[21] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I., "Attention is all you need," *Advances in neural information processing systems*, Vol. 30, 2017.

[22] Imambi, S., Prakash, K. B., and Kanagachidambaresan, G., "PyTorch," *Programming with TensorFlow: solution for edge computing applications*, Springer, 2021, pp. 87–104.

[23] Stephenson, M. A., Mantovani, L. Q., and Schaub, H., "Learning Policies for Autonomous Earth-Observing Satellite Scheduling over Semi-Markov Decision Processes," *Journal of Aerospace Information Systems*, 2025, pp. 1–11.

[24] Liang, E., Liaw, R., Nishihara, R., Moritz, P., Fox, R., Gonzalez, J., Goldberg, K., and Stoica, I., "Ray rllib: A composable and scalable reinforcement learning library," *arXiv preprint arXiv:1712.09381*, Vol. 85, 2017, p. 245.