Autonomous Tip-and-Cue Earth-Observing Constellation Tasking with Reinforcement Learning

Mark Stephenson
University of Colorado, Boulder
3775 Discovery Drive
Boulder, CO, 80308
mark.a.stephenson@colorado.edu

Hanspeter Schaub
University of Colorado, Boulder
3775 Discovery Drive
Boulder, CO, 80308
hanspeter.schaub@colorado.edu

***Abstract*—** While Earth-observing constellations often collect images from an *a priori* request list, this paradigm greatly limits the phenomena that can be observed: Emergent and unpredictable events are often valuable imaging targets. Although tip-and-cue architectures exist to image such events—usually with a "tipping" leader satellite that cues observations by the follower satellite(s)—these lack the flexibility or capacity desired from modern Earth-observing constellations. In this work, reinforcement learning is demonstrated as a way of autonomously and scalably tasking a homogenous constellation of satellites with scanning and imaging instruments. A per-agent policy is learned that is executable onboard each satellite and able to respond to the high-uncertainty environment, solving a problem that traditional pre-planning approaches cannot handle and demonstrating collaborative behavior between agents. As satellites are added to a constellation, the performance of the satellites working together grows faster than the number of satellites. Depending on the location of a satellite within the constellation, it may assume a specific role that biases it towards scanning or imaging.

Table of Contents

1. Introduction

While Earth-observing constellations have become increasingly prevalent, the operation of these systems has largely developed within a self-limiting framework: Given a list of *a priori* requests of varying values, operators create a global schedule that maximizes the output of the constellation subject to operational constraints [1]. However, the increasing availability of increasingly capable satellites (both with respect to sensing ability and onboard compute power) enables previously unavailable mission architectures [2]. In particular, emergent phenomena—such as natural disasters, human events, and scientifically interesting occurrences—cannot be responsively imaged in the existing paradigm, but distributed autonomy would allow constellations to adapt in order to exploit unpredictable and unanticipated events [3].

979-8-3315-7360-7/26/$31.00 ©2026 IEEE

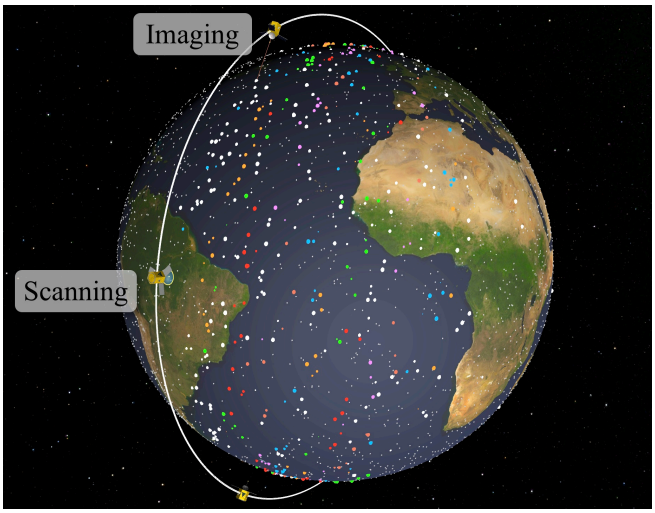


Figure 1: Visualization of LEO satellites in the scanning and imaging modes with unknown (gray), found (white), and imaged (colored) targets, configured in the $S = 6$ satellite ring constellation.

Such a system would extend existing tip-and-cue behavior, in which a leader "tips" following satellites about events of interest, to an automated, distributed, constellation-wide tip-and-cue architecture [4].

In the traditional agile Earth-observing satellite scheduling problem (AEOSSP), a list of requests with associated priorities must be satisfied by a satellite or constellation of satellites that can slew agilely (i.e., along and across track) to point at and image targets. Solution methods tend to follow the same general process [5, 6, 7]: First, a representation of feasible request sequences (usually, a graph) is generated from the request list, fulfillment constraints, and system parameters. Then, a discrete optimizer, such as a mixed-integer linear program (MILP) solver or iterative local search (ILS), is used to find a feasible sequence that maximizes the value fulfilled requests. Because such optimizers are computationally expensive (especially as the number of satellites, number of requests, and horizon increases), the previous steps are completed on the ground, and the plan is uploaded to the constellation at the next opportunity. Sometimes, satellites will be equipped with a method of repairing small segments of the plan onboard if they are interrupted [3].

Contrasting traditional methods with offline planning and open-loop execution, reinforcement learning (RL) offers a closed-loop, onboard approach to planning and scheduling. References [8], [9] and [10] apply RL to satellite schedul-

ing problems, demonstrating how mission objectives and resource constraints for Earth observation can be managed autonomously onboard a satellite. Reference [11] shows that deep reinforcement learning (DRL) yields autonomous policies that are capable of performing competitively with optimal schedulers for the AEOSSP. This prior work leverages the closed-loop properties of these results to plan for a responsive scheduling problem that cannot be solved by traditional methods.

Decentralized job allocation in constellations has been considered previously in order to avoid the computational expense of global planners. An early DRL algorithm is applied in reference [12] to combine individual and collaborative task planning for satellites. In reference [13], self-adaptive complex system theory is used for multi-satellite mission planning, with a focus on flexibility and robustness. References [14] and [15] formulate the allocation problem as a distributed constraint optimization problem (DCOP), allowing a variety of algorithms with different levels of communication requirement to be applied. In references [16] and [17], policies learned in a single-agent RL environment are induced to work together to deconflict requests. Unlike this paper, these prior works all still assume an *a priori* task list to be distributed.

The necessity of having responsive satellites has also been well-established in the literature. One of the earliest examples of using data collected onboard to update the mission plan in a closed-loop manner was on Earth Observing One (EO-1) [18]. Over time, EO-1 and other assets have been used as part of a larger sensor web in order to automate detection, tasking, and data acquisition [19]. More recently, autonomous responsive tasking has been demonstrated and flown in a single-satellite architecture in a method known as dynamic targeting [20, 21, 22]. This system uses a forward-looking sensor to identify obstructed or valuable regions of the upcoming ground track, then points the imaging instrument accordingly. Other work studies decentralized consensus-based algorithms for reactive replanning of Earth-observing satellites, finding that the efficacy of replanning for reobservation of new events is highly dependent on constellation geometry [23, 24].

This paper expands the Markov decision process (MDP) formulation of the standard AEOSSP given in reference [11] into a tip-and-cue problem with a constellation of agile low Earth orbit (LEO) satellites (Figure 1). Instead of being given an *a priori* request list, satellites are equipped with a wide field-of-view scanning instrument that identifies new imaging targets and a narrow field-of-view imaging instrument that can collect reward-yielding images of those targets. To perform this task well, agents must learn a decentralized policy that balances searching for new targets and exploiting known targets; ultimately, this must be a collaborative behavior between agent. In this work, policies are learned and tested across a variety of constellations. By examining the individual and collective behaviors of the satellites, it is apparent that agents learn to work together by diversifying roles within the constellation and increasing overall performance.

2. Problem Formulation

The search-and-image problem formulation (Figure 2) is a modification of the standard AEOSSP environment described in reference [11]. The primary changes are to the appearance of potential targets in the environment, how targets are identified by the satellites, and how information is shared within the constellation.

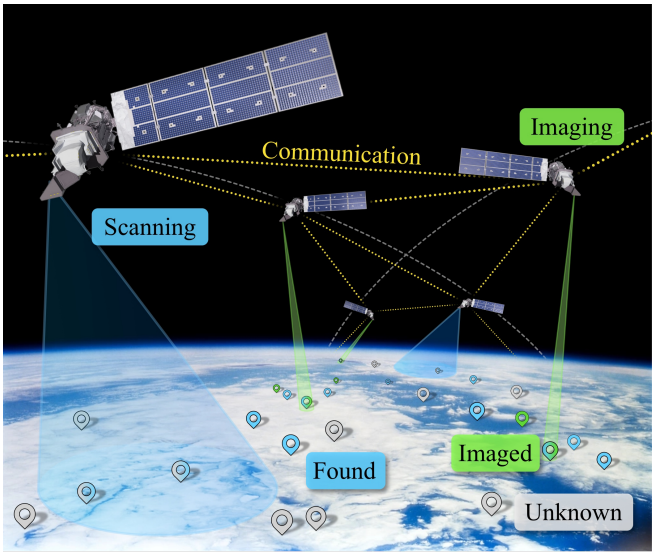


Figure 2: Constellation architecture for homogenous satellites equipped with scanning and imaging instruments.

Target Model—In the problem, each target is defined by a tuple of Earth-fixed location, priority, and appearance time $\tau_i = (\boldsymbol{r}_i, r_i, t_i)$. When a new target appears at time t_i, it is added to the set of all targets $\mathcal{T}$. New targets appear in the environment at the rate $\dot{\tau}$, which is randomized between 100 and 1000 targets per hour, with locations uniformly distributed over Earth's surface. When a satellite identifies a previously unknown target using its scanning instrument, the target is added to the set of known targets $\mathcal{K}$. Once a target is known, it may be imaged with a satellites imaging instrument; this adds the target to the set of imaged targets $\mathcal{I}$ and yields a reward equal to the target's priority $r_i \in [0, 1]$. Each of these sets is a subset of the previous: $\mathcal{I} \subset \mathcal{K} \subset \mathcal{T}$.

Instrument Models— Each instrument in the homogenous constellation has two instruments: a scanning instrument for identifying unknown targets and an imaging instrument for collecting images of known targets.

The imaging instrument works as described in reference [11]. It is a camera with a body-fixed boresight $\hat{\boldsymbol{c}}$ that is aimed using the agilely maneuvering satellite; the image is automatically collected once the attitude controller [25] has settled. This instrument can collect images of targets with an elevation angle $\phi > 58°$, which corresponds to a circular field-of-regard with a 500 km radius.

The scanning instrument is used when the satellite is in a nadir-pointing attitude to reveal unknown targets within the 500 km-radius field-of-regard. When activated, the instrument follows a sequence consisting of: 45 seconds of warm-up and slewing time; 90 seconds of active scanning time; and 45 seconds of image processing time, during which targets scanned with the instrument are identified.

Communication—Free communication is assumed within the constellation. Once every 50 seconds, satellites broadcast a list of newly scanned and imaged targets to update the constellation's knowledge of the environment. While this degree of constellation-wide communication is unrealistic for current systems, it is justified in two ways: advances in in-space

Table 1: Observation vector elements; request observations are given for next $N = 32$ upcoming unimaged targets in $\mathcal{K} \setminus \mathcal{I}$.

Quantity	Dim.	Description
${}^{\mathcal{H}}\boldsymbol{\omega}_{\mathcal{BE}}$	3	Body angular rate
${}^{\mathcal{H}}\hat{\boldsymbol{c}}$	3	Hill-frame instrument direction
${}^{\mathcal{E}}\boldsymbol{r}_{\mathcal{BE}}$	3	Earth-fixed position
${}^{\mathcal{E}}\boldsymbol{v}_{\mathcal{BE}}$	3	Earth-fixed velocity
$r_{n\in N}$	N	Target priorities
${}^{\mathcal{H}}\boldsymbol{r}_{n\in N}$	$3N$	Target positions
$\delta\theta_{n\in N}$	N	Target pointing errors
$t^o_{n\in N}, t^c_{n\in N}$	$2N$	Target opportunity windows

communication networks reasonably imply that bandwidth may become a less restricted resource, and prior research has demonstrated that in similar scenarios, local-only and global communication produce similar results since only activities from other physically proximate impact decision-making in a time-sensitive way [26].

Constellation Geometry—Two constellation geometries are considered in this work, both parameterized by the number of satellites S. The **ring** constellation consists of S satellites equally spaced in a single-plane $90° \times 800$ km orbit. The **string** constellation consists of S satellites separated by $10°$ true anomaly in a single-plane $60° \times 800$ km orbit.

Objective—The objective of the environment is straightforward: maximize the sum of priorities of imaged targets

$$\text{maximize} \sum_{\tau_i \in \mathcal{I}} r_i \tag{1}$$

subject to the required operational sequence of scanning then imaging and other environment dynamics.

MDP Formalization

In order to find a policy for this problem, it must be formulated as a decentralized partially-observable semi-Markov decision process (Dec-POsMDP). The properties of this formulation are discussed in section 3. More detail about the AEOSSP MDP that this is based on can be found in [11]. The elements of the MDP are defined as follows:

- **State Space:** The state space consists of all information about the environment and the dynamics simulation necessary to propagate the environment. Since the total state is very high dimensional and most of these states are irrelevant to decision-making, a hand-designed observation space is selected.
- **Observation Space:** The per-satellite observation is as given in Table 1. The observation consists of relevant information about the satellite, such as its position, velocity, attitude, and pointing direction, as well as information about the next $N = 32$ along-track targets in the known-but-not-imaged set $\mathcal{K} \setminus \mathcal{I}$. Target information includes Hill-frame $\mathcal{H}$ relative position and the angle $\delta\theta$ between the imaging instrument boresight $\hat{\boldsymbol{c}}$ and the target. The positions of the upcoming targets provide insight into whether the satellite should scan to find more targets or try to image known targets.
- **Action Space:** Each satellite has $N + 1 = 33$ actions available. Imaging actions $a_{\text{im},i}$ account for 32 of the actions. With this action, the satellite attempts to image the corresponding target in the observation space. The action is not guaranteed to be successful, as the target may go out of range before the controller has settled to point the instrument at the target. If the action is successful, the target is added to the imaged set $\mathcal{I}$. The satellite also has the scanning action a_{scan}. This action activates the previously described sequence of warm-up, scan, and cool-down, then adds any detected targets to the known set $\mathcal{K}$.
- **Reward Function:** The reward function reflects the optimization objective defined in Equation 1. With $\mathcal{I}_s$ being a satellite's imaged set before the step and $\mathcal{I}'_s$ after the set, the reward at a step for satellite s is

$$r_s(\mathcal{I}_s, \mathcal{I}'_s) = \sum_{\tau_i \in \mathcal{I}'_s \setminus \mathcal{I}_s} r_i \tag{2}$$

unless multiple satellites have imaged the same target at the same step, in which case the reward is distributed evenly among them.
- **Transition Model:** Transitions are given by a generative model (i.e., a simulator). At each step, the simulation is propagated until any satellite completes an action. When that action is done, a new action is selected for that satellite and the simulation continues. Since actions can take different amounts of time, this results in the satellites acting asynchronously. Episodes are executed for 15 orbits before the environment is reset and rerandomized.

Implementation—The MDP is implemented using BSK-RL[2], a package for defining high-speed, high-fidelity RL environments for spacecraft tasking [27]. The environment uses Basilisk [28] for spacecraft dynamics and flight software modelling, and it provides an interface to RL libraries using the standard PettingZoo [29] and Gymnasium [30] APIs.

The fidelity of the simulation environment allows the results to be applicable to a flight-like system; no major simplifications are made, and any additional effects that one may wish to account for can be added to the simulation. In this work, satellites are modelled using a multibody physics simulation. Flight-proven flight software actuates the reaction wheels and controls the instruments. Satellites are affected by J_2 effects and atmospheric drag. Solar system data is loaded using SPICE [31]. System parameters are the defaults given in BSK-RL, other than those specified in reference [11].

3. Reinforcement Learning

The objective of RL is to find the policy $a = \pi(s)$, or mapping from states (or observations) to actions, that maximizes the expected γ-discounted sum of future rewards, which is known as the value:

$$V^\pi_{\text{MDP}}(s) = r + \gamma \mathbb{E}\left[V^\pi_{\text{MDP}}(s') \middle| T(s'|s,a), \pi(a|s)\right] \tag{3}$$

$$= r_0 + \gamma r_1 + \gamma^2 r_2 + \cdots = \sum_{i=0}^{\infty} \gamma^i r_i \tag{4}$$

The learning agent does not directly know information about the environment; rather, it must explore the environment and gain experience to maximize the performance of the policy [32].

Modifications to the MDP

This problem requires various modifications to the standard MDP formulation, yielding a decentralized partially-

[2] https://avslab.github.io/bsk_rl/

observable semi-Markov decision process (Dec-POsMDP). The rationale for these modifications are described below.

Semi-MDPs—Semi-Markov decision processes (sMDPs) are used to describe MDPs in which steps have a non-constant time differential associated with them [33, 34]. While many problems can be represented with fixed timesteps (e.g. turn-based games, discretized continuous control, etc.), scheduling problems often include tasks with different time costs associated with them. In this problem, each imaging action takes a variable amount of time, which is also different from the duration of the scanning action; thus, using the sMDP framework is advantageous. To encode this opportunity cost in the value function, the discount factor γ becomes a discount rate that is exponentiated by the amount of time elapsed rather than the number of steps:

$$V_{\text{sMDP}}(s_0) = \gamma^{\Delta t_0} r_0^{(\gamma)} + \gamma^{\Delta t_0 + \Delta t_1} r_1^{(\gamma)} + \dots \tag{5}$$

$$= \sum_{t=0}^{\infty} \gamma^{\sum_{i=0}^{t} \Delta t_i} r_t^{(\gamma)} \tag{6}$$

where the γ-discounted step reward is

$$r^{(\gamma)} = \int_0^{\Delta t} \gamma^t \rho(t) \mathrm{d}t \tag{7}$$

where $\rho(t)$ is a reward density function over the course of a step; in this problem, all reward densities are given as a Dirac-δ function at the end of the step. This modification is also made within any value-like computations within training algorithms, such as generalized advantage estimation [35].

Decentralization—Decentralization and partial observability are common properties in multiagent RL environments [36]. In these cases, each agent receives an individual observation of the environment and independently acts accordingly. For this problem, each agent makes decisions independently based on their local view of the environment. However, all experience is used to train the same policy, of which copies are executed independently on each satellite for a scalable solution.

Asynchronicity— Multi-agent sMDPs with decentralized decision-making lead to asynchronicity: Because different actions of different durations are taken by different agents, decisions are not made at the same time. Reference [37] demonstrates that learning is possible under these conditions in similar scheduling problems.

Training Pipeline

In this work, proximal policy optimization (PPO) is used for training policies [38]. PPO is a widely-used DRL algorithm that has been demonstrated to perform well across domains such as games and robotics. It has previously been demonstrated on other Earth-observation scheduling problems [11]. The RLlib implementation of PPO is used [39], modified to handle asynchronicity and semi-Markov intervals[3].

A hyperparameter search was performed to find a strong training configuration, yielding a learning rate of 3×10^{-5}, discount rate of 0.995 (with time units of s), and batch size of 100 samples per thread. All other parameters use the RLlib defaults. Policies are represented by a 2×2048 node multilayer perceptron (MLP). Each policy is trained for 48 hours on 32 threads; this corresponds to 3.3M to 3.7M environment interactions and 8.8 to 13.8 satellite-years on-orbit. Four policies are trained: String-3 and String-6 are trained on a string constellation with $S = 3$ and 6 respectively, and Ring-6 and Ring-12 are likewise trained on a ring constellation. In each episode, the target appearance rate $\dot{\tau}$ is randomly sampled between 100 and 1000 targets per hour. Each episode is propagated for 15 orbits of decision-making.

[3]Examples of these modifications are given in avslab.github.io/bsk_rl/examples/time_discounted_gae.html and avslab.github.io/bsk_rl/examples/async_multiagent_training.html.

Table 2: Total reward ($\mu \pm \sigma$) relative to Ring-6 policy across all benchmarks [%].

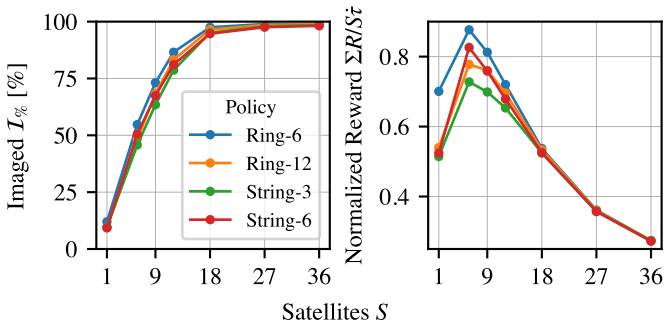
	Benchmark Constellation		
Policy	**Ring**	**String**	**All**
Ring-6	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
Ring-12	92.1 ± 9.7	86.3 ± 13.4	88.8 ± 12.3
String-3	91.1 ± 7.7	94.0 ± 6.7	92.7 ± 7.3
String-6	93.5 ± 8.1	97.8 ± 6.0	95.9 ± 7.3

4. Policy Performance

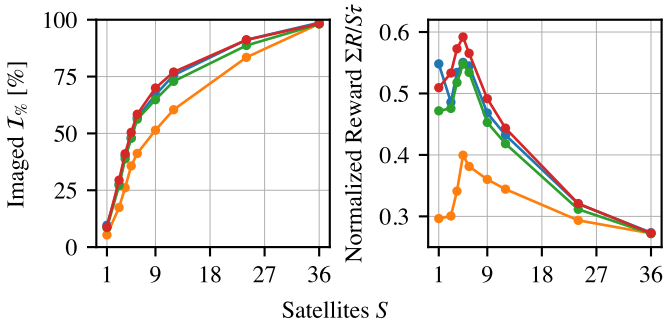
To evaluate the policies, each is benchmarked across different constellations and target rates for 15 orbits. The ring and string constellation are both tested for S between 1 and 36 satellites; other than a difference in inclination (90° versus 60°), the two constellations are the same for $S = 36$. The target rate $\dot{\tau}$ is varied between 100 and 1000 targets per hour, as the agent was exposed to in training. The size of the policy network makes its practical impact on storage and compute negligible, with inference taking 10 to 20 ms at each decision.

A few key metrics are examined to understand the behavior and performance of the policies under different conditions. The known percentage $\mathcal{K}_{\%}$ and imaged percentage $\mathcal{I}_{\%}$ reflect the fraction of targets with at least one opportunity that are respectively scanned and imaged by the constellation; this metric is used so that the performance is not diluted by targets that no satellite has access to over the course of the simulation purely due to geometry. Imaged per known $\mathcal{I}/\mathcal{K}$ gives the fraction of scanned targets that are eventually imaged. The scanning mode percent is the fraction of time spent taking the scanning action (as opposed to an imaging action) on average by all satellites in the constellation. The total reward ΣR is the sum of the priorities of imaged targets, i.e., the undiscounted sum of rewards. The normalized reward $\Sigma R / S\dot{\tau}$ is the previous metric normalized by the target appearance rate and the number of satellites.

The four policies, Ring-6, Ring-12, String-3, and String-6, are compared in cross-sections of the benchmark in Figure 3. The Ring-6 policy performs best across all satellite counts when deployed in the ring constellation (Figure 3a) at $\dot{\tau} = 500$, while for the target appearance rate $\dot{\tau} = 1000$ String-6 policy is generally best when deployed in the string constellation (Figure 3b). This is expected based on the hypothesis that policies perform best when the deployment environment is more similar to the training environment. However, Table 2 shows that on average the Ring-6 constellation is best, even if other policies slightly outperform it in certain cases. The Ring-6 policy is used throughout the remainder of the paper due to its strong ability to generalize.



(a) Ring benchmark with $\dot{\tau} = 500$ targets per hour.



(b) String benchmark with $\dot{\tau} = 1000$ targets per hour.

Figure 3: Policy performance comparison on a subset of benchmark cases.

The full benchmark of the Ring-6 policy on the ring and string constellations are given in Figure 4a and Figure 5, respectively. One of the most important trends is in the $\mathcal{I}/\mathcal{K}$ ratio: as long as $S > 1$, at least a majority of scanned targets are imaged, with fractions $> 75\%$ in both constellations if $S \geq 6$. Unsurprisingly, increasing the number of satellites "saturates" the environment: with enough satellites, nearly all possible targets are scanned and imaged. As the number of satellites increases, the fraction of time spend in the scanning mode also increases in order to avoid exhausting the known target list.

5. Evidence for Collaboration

A critical goal of this work is demonstrating that the agents have learned to collaborate. There are two questions that can be investigated to demonstrate this:

1. Does increasing the number of agents scale performance at a rate greater than one?
2. Do agent behaviors diversify when working with other agents?

Both of these questions can be answered affirmatively by examining the behavior of the agents across various cases.

Multi-Satellite Performance Gain

The first question to test for collaboration—whether increasing the number of satellites disproportionately improves performance—can be addressed by comparing the behavior of the policy and constellation with and without communication. In the no communication cases, the satellite known $\mathcal{K}_s$ and imaged $\mathcal{I}_s$ sets are maintained individually, without ever updating based on other satellites' knowledge.

Communication Benchmark Comparison—The communication case (Figure 4a) and no communication case (Figure 4b) are compared across all metrics. The no communication case is uniformly worse than the standard case with communication. The percent of known targets is slightly depressed, and the percent of imaged targets is considerably lower. This is despite the fact that the no communication case spends as much or more time than the communication case in the scanning mode.

Image Counts—It remains a possibility that the no communication case primary does poorly relative to the standard environment due to many images being duplicates of those already taken by another satellite. Previous work has shown that simply introducing a request deduplication mechanism between independent imaging agents improves performance, but this does not imply learned collaboration [17].

To examine this, Figure 6 compares the ratios of unique and total (unique + duplicated) images between the communication and no communication cases. As expected, communication dominates no communication for the number of unique images in all cases. Also unsurprisingly, when there are a large number of satellites and relatively few targets, the no communication case dominates for total images since each target can be reimaged by many satellites (which communication prevents). The evidence for collaboration lies in the low satellite count, high target count region of the total images plots: In these cases, communication leads to more total images (without any duplication) than the no communication case (which could be duplicating images). This implies that the satellites are working together in a way that improves overall performance beyond what communication-based image deduplication induces.

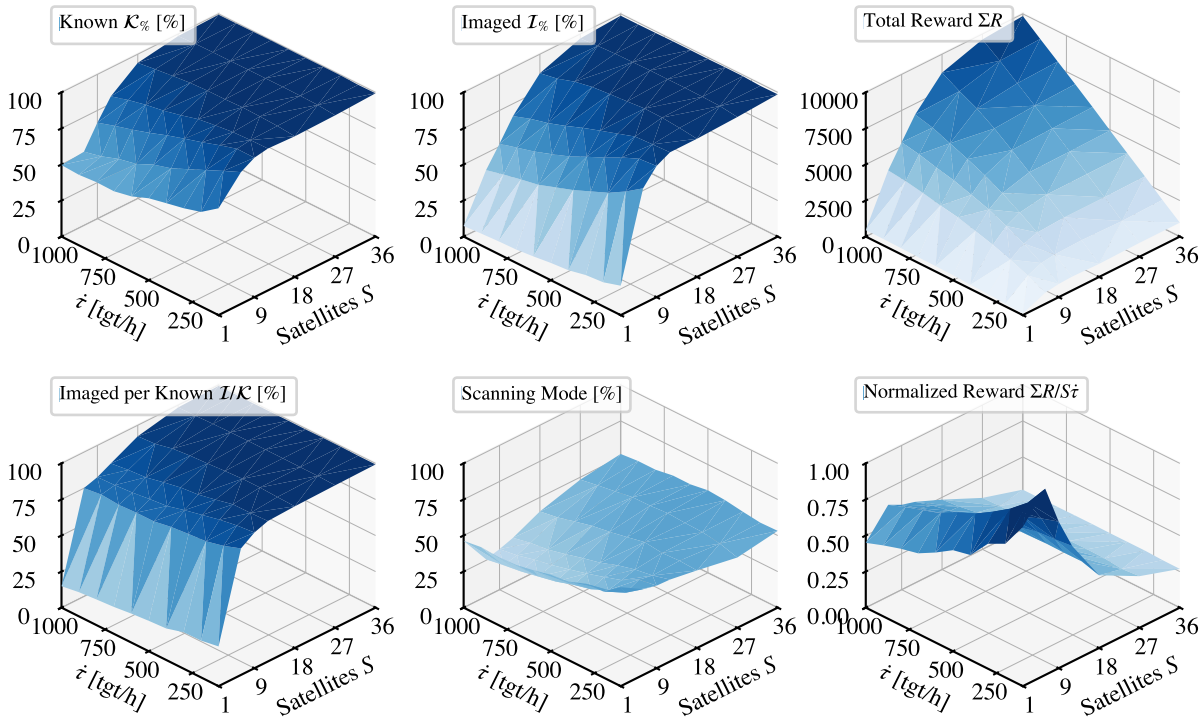
Individual Behaviors

Collaboration is also implied by implicit role assignment or diversification of behaviors within the constellation. If a satellite acts differently due to the presence and actions of another satellite, and that positively impacts the constellation-wide performance, collaboration is occurring.

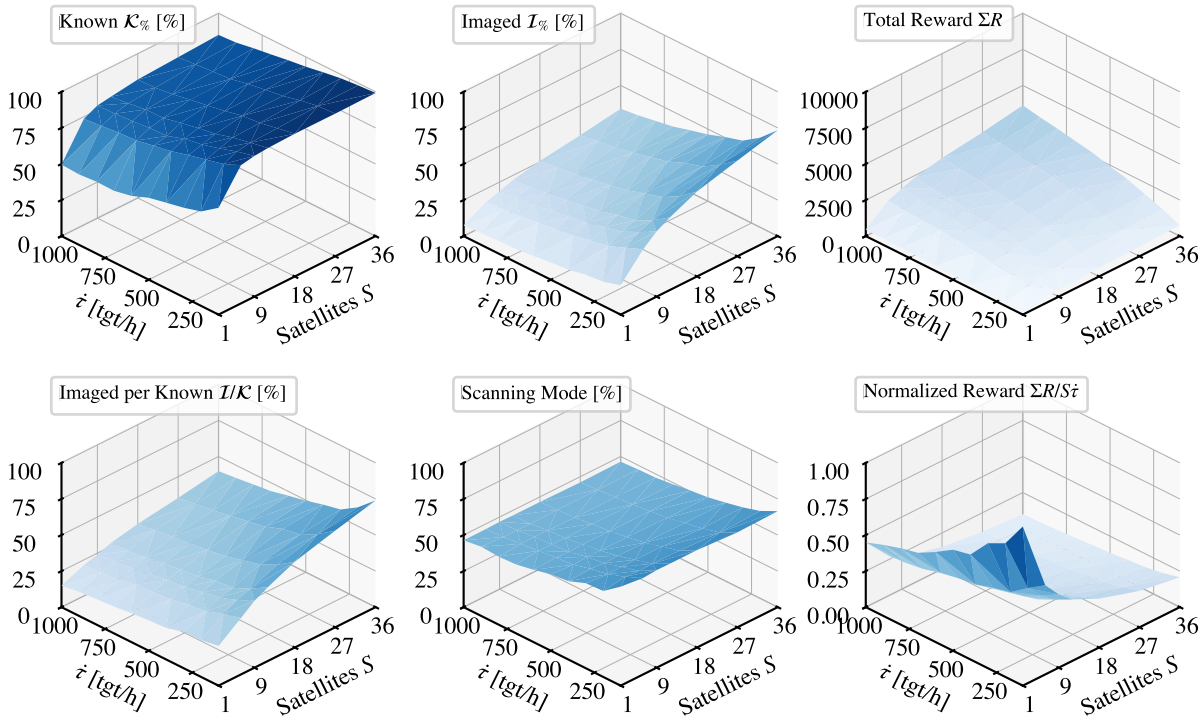
Discontinuity for Single Satellites—The first evidence that satellite behavior changes when working together is apparent in the policy benchmarks (Figure 4a and Figure 5). The $S = 1$ cases are discontinuous relative to the multisatellite behavior in many of the metrics. Both the types of actions being taken and the outcomes of those actions are considerably different when there are not other satellites impacting the environment.

Per-Agent Scanning Frequency—In the string constellation, a satellite's position in the string is predicted to influence the behaviors exhibited by that satellite. Figure 7 tests this hypothesis by plotting the amount of time spent in the scanning mode versus the satellite's position in the string. The leader scans 40% to 68% of the time, since known targets are sparse in the upcoming ground track. The fourth to sixth satellite spends the smallest fraction of time (13% to 47%) in the scanning mode, since the leading satellites have found a dense list of targets to be scanned. As one looks farther back in the string, scanning fractions increase since most possible targets have been scanned and imaged, so these followers can only find newly appearing targets.

Differing scanning behavior is also expected among agents



(a) In the standard environment with communication.



(b) Without communication.

Figure 4: Ring constellation performance of the Ring-6 policy with varying target rate $\dot{\tau}$ and satellite count S.

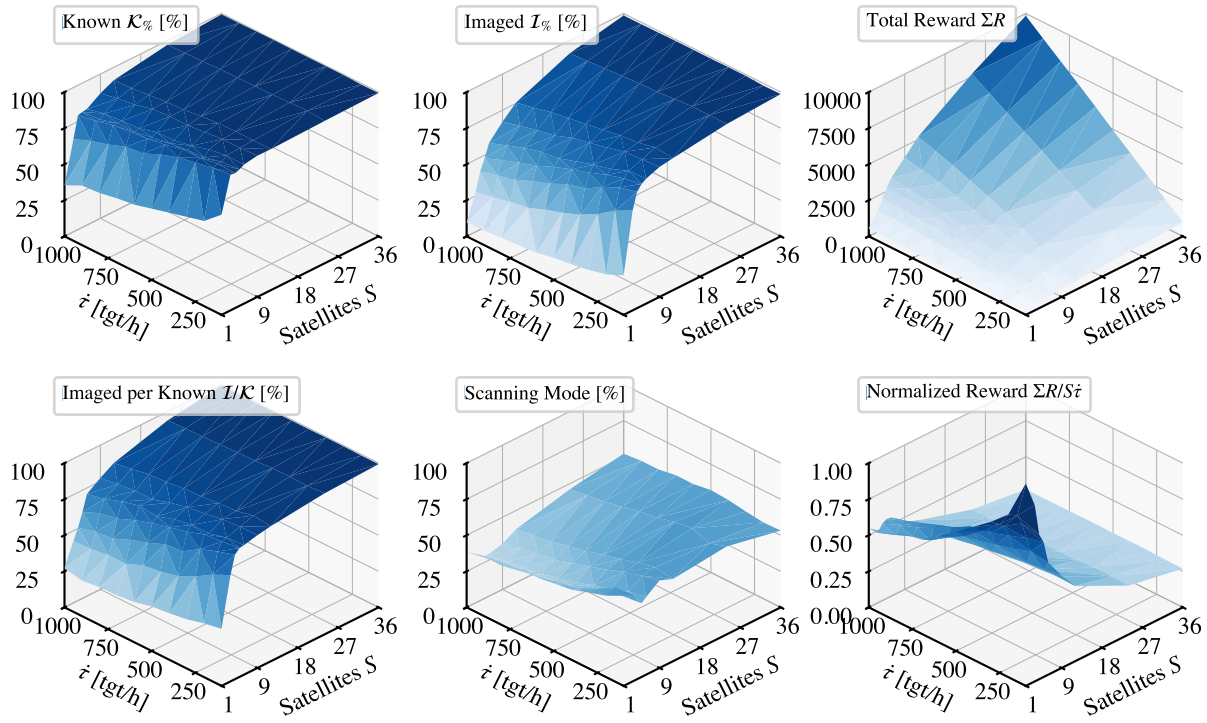


Figure 5: String constellation performance of the Ring-6 policy with varying target rate $\dot{\tau}$ and satellite count S.

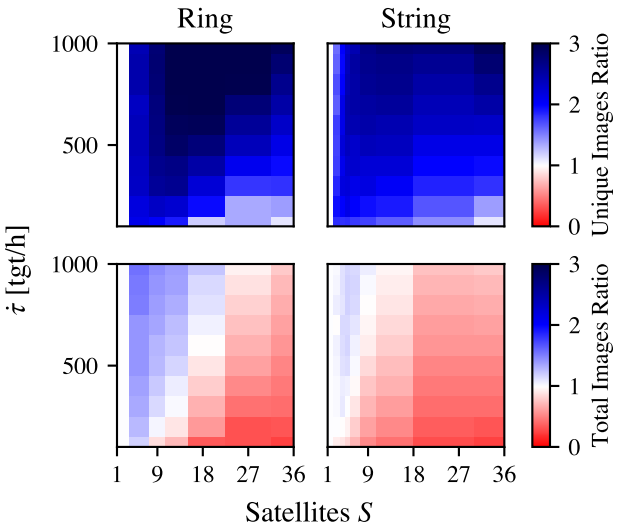


Figure 6: Ratio of unique and total (unique + duplicated) images between constellations with and without communication.

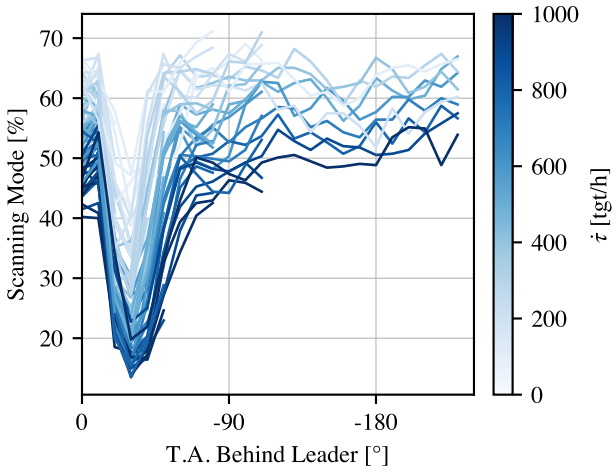


Figure 7: Scanning time fraction of each satellite in various string constellations.

in the ring constellation, but due to the constellation's symmetry, a bias towards scanning versus imaging is expected to change over time. Figure 8 shows the orbit-by-orbit scanning percentage of each agent in an $S = 6$ string constellation. While initially the behaviors are homogenous for the first two orbits, diversification of behaviors is first evident for orbits two through five, where the even-indexed satellites take a scanning-heavy role, while odd-indexed satellites tend towards imaging. This behavior inverts—though less prominently—during orbits six through eight. In essence, the constellation has dynamically created tip-and-cue pairs, as needed. Through the end of the episode, scanning is less frequent throughout the constellation since many targets are available to image that were scanned but not imaged during earlier opportunities.

Scan-Image Delays—The final evidence for collaboration is that the satellite that scans a given target is typically not the satellite that images it. Figure 9 shows a histogram of the scan-image delay for all imaged targets in a ring constellation with $S = 6$ and $\dot{\tau} = 1000$ targets per hour. On the left

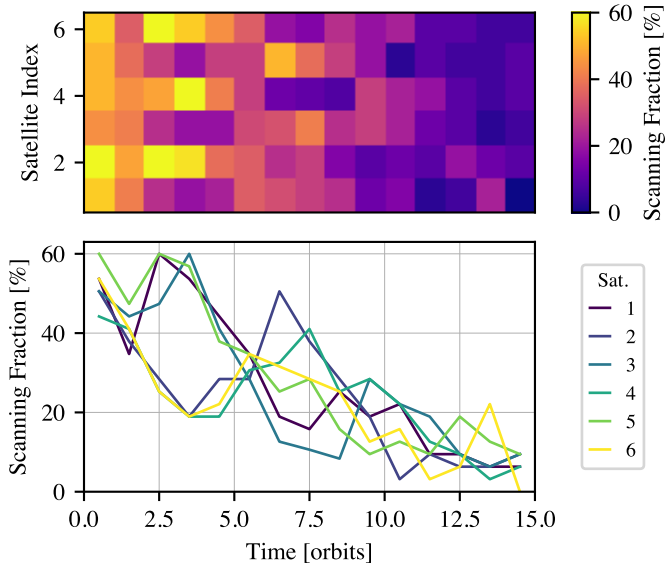


Figure 8: Scanning time fraction of each satellite in a ring constellation. Target rate $\dot{\tau} = 1000$ tgt/h.

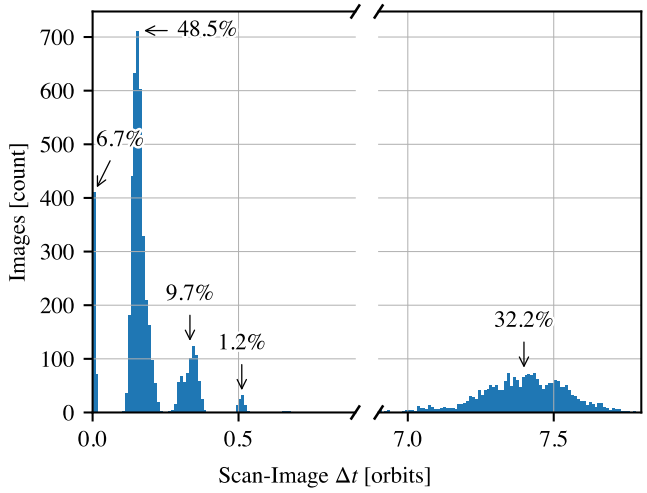


Figure 9: Histogram of delay between scanning and imaging in the ring constellation with $S = 6$ satellites.

side of the plot, each grouping is a different follower satellite imaging the scanned target. As such, only 6.7% of targets are imaged by the same satellite that scanned it, while 59.4% are imaged by one of the next three following satellites (at which point Earth's rotation has moved the target out of the ground track). The remainder of images are taken a half-day later when the satellites have another opportunity to image them; at this point, the specific agent that images a target is irrelevant. Satellites are working together because the majority of images are collected with minimal delay by a later satellite.

6. Conclusions

This work presents a scalable solution to constellation-wide tip-and-cue scheduling, in which satellites must first scan regions to reveal new targets before imaging those targets. Formulating the problem as a Markov decision process (MDP) and solving it using deep reinforcement learning (DRL) is identified as a promising method for task, as it allows for per-satellite, closed-loop scheduling behaviors to be learned. This is necessary, as traditional preplanning methods are unable to account for target locations that are dynamically revealed. Benchmarks of the policy across constellation configurations and target appearance rates yield desirable performance; this shows that the method generalizes and scales well, and is not tied to a specific constellation architecture.

A significant finding of this work is that satellites appear to actively collaborate within the environment. As satellites are added to a constellation, the performance of the satellites working together grows faster than the number of satellites. Individually, the policy yields diverse behaviors among the agents: Depending on the location of a satellite within the constellation, it may assume a specific role that biases it towards scanning or imaging. As a result, it is clear that the policy found with reinforcement learning (RL) learns collaborative behaviors for the tip-and-cue constellation environment.

Future work will include implementing and comparing the policy's performance to a heuristic method as well as studying the performance of the system under more flight-like communication conditions and request distributions.

Acknowledgements

This work is supported by NASA Space Technology Graduate Research Opportunity grant 80NSSC23K1182.

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).

References

[1] X. Wang, G. Wu, L. Xing, and W. Pedrycz, "Agile Earth Observation Satellite Scheduling Over 20 Years: Formulations, Methods, and Future Directions," *IEEE Systems Journal*, vol. 15, no. 3, pp. 3881–3892, Sep. 2021.

[2] D. Selva and D. Krejci, "A survey and assessment of the capabilities of Cubesats for Earth observation," *Acta Astronautica*, vol. 74, pp. 50–68, May 2012.

[3] G. Picard, C. Caron, J.-L. Farges, J. Guerra, C. Pralet, and S. Roussel, "Autonomous Agents and Multiagent Systems Challenges in Earth Observation Satellite Constellations," in *International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021)*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, May 2021, pp. 39–44.

[4] S. Dockstader and P. Labs, "Future Trends In New Space: Automated Tip & Cue," 2021.

[5] S. Augenstein, "Optimal Scheduling of Earth-Imaging Satellites with Human Collaboration via Directed Acyclic Graphs," *The Intersection of Robust Intelligence and Trust in Autonomous Systems: Papers from the AAAI Spring Symposium*, pp. 11–16, 2014.

[6] S. Nag, A. S. Li, and J. H. Merrick, "Scheduling algo-

rithms for rapid imaging using agile Cubesat constellations," *Advances in Space Research*, vol. 61, no. 3, pp. 891–913, Feb. 2018.

[7] M. A. Stephenson and H. Schaub, "Optimal Agile Satellite Target Scheduling with Learned Dynamics," *Journal of Spacecraft and Rockets*, pp. 1–12, Oct. 2024.

[8] A. Harris, T. Teil, and H. Schaub, "Spacecraft Decision-Making Autonomy Using Deep Reinforcement Learning," in *AAS Spaceflight Mechanics Meeting*, Maui, Hawaii, Jan. 2019.

[9] A. Hadj-Salah, R. Verdier, C. Caron, M. Picard, and M. Capelle, "Schedule Earth Observation satellites with Deep Reinforcement Learning," Nov. 2019.

[10] D. Eddy and M. Kochenderfer, "Markov Decision Processes For Multi-Objective Satellite Task Planning," in *2020 IEEE Aerospace Conference*. Big Sky, MT, USA: IEEE, Mar. 2020, pp. 1–12.

[11] M. A. Stephenson, L. Q. Mantovani, and H. Schaub, "Learning Policies for Autonomous Earth-Observing Satellite Scheduling over Semi-MDPs," *Journal of Aerospace Information Systems*, vol. 22, no. 9, pp. 789–799, 2025.

[12] C. Wang, J. Li, N. Jing, J. Wang, and H. Chen, "A Distributed Cooperative Dynamic Task Planning Algorithm for Multiple Satellites Based on Multi-agent Hybrid Learning," *Chinese Journal of Aeronautics*, vol. 24, no. 4, pp. 493–505, Aug. 2011.

[13] J. Bonnet, M.-P. Gleizes, E. Kaddoum, S. Rainjonneau, and G. Flandin, "Multi-satellite Mission Planning Using a Self-Adaptive Multi-agent System," in *2015 IEEE 9th International Conference on Self-Adaptive and Self-Organizing Systems*, Sep. 2015, pp. 11–20.

[14] S. Parjan and S. A. Chien, "Decentralized Observation Allocation for a Large-Scale Constellation," *Journal of Aerospace Information Systems*, vol. 20, no. 8, pp. 447–461, Aug. 2023.

[15] I. Zilberstein, A. Rao, M. Salis, and S. Chien, "Decentralized, Decomposition-Based Observation Scheduling for a Large-Scale Satellite Constellation," *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 34, pp. 716–724, May 2024.

[16] M. Stephenson, L. Mantovani, and H. Schaub, "Intent Sharing for Emergent Collaboration in Autonomous Earth Observing Constellations," in *AAS GN&C Conference*, Breckenridge, CO, Februrary 2, 2024.

[17] M. Stephenson, L. Mantovani, A. Cheval, and H. Schaub, "Quantifying the Optimality of a Distributed Rl-Based Autonomous Earth-Observing Constellation," in *AAS GN&C Conference*, Breckenridge, CO, Feb. 2025.

[18] S. Chien, R. Sherwood, D. Tran, B. Cichy, G. Rabideau, R. Castano, A. Davis, D. Mandl, S. Frye, B. Trout, S. Shulman, and D. Boyer, "Using Autonomy Flight Software to Improve Science Return on Earth Observing One," *Journal of Aerospace Computing, Information, and Communication*, vol. 2, no. 4, pp. 196–216, Apr. 2005.

[19] S. Chien, D. Mclaren, J. Doubleday, D. Tran, V. Tanpipat, and R. Chitradon, "Using Taskable Remote Sensing in a Sensor Web for Thailand Flood Monitoring," *Journal of Aerospace Information Systems*, vol. 16, no. 3, pp. 107–119, Mar. 2019.

[20] A. Candela, J. Swope, and S. A. Chien, "Dynamic Targeting to Improve Earth Science Missions," *Journal of Aerospace Information Systems*, vol. 20, no. 11, pp. 679–689, Nov. 2023.

[21] A. Kangaslahti, A. Candela, J. Swope, Q. Yue, and S. Chien, "Dynamic Targeting of Satellite Observations Incorporating Slewing Costs and Complex Observation Utility *," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. Yokohama, Japan: IEEE, May 2024, pp. 4876–4882.

[22] S. Chien, I. Zilberstein, A. Candela, D. Rijlaarsdam, T. Hendrix, A. Dunne, O. Aragon, and J. P. Miquel, "Flight of Dynamic Targeting on CogniSAT-6," International Symposium on Artificial Intelligence, Robotics, and Automation for Space, May 2025.

[23] B. Gorr, A. A. Jaramillo, H. Gao, D. Selva, A. Mehta, Y. Sun, V. Ravindra, C. H. David, and G. H. Allen, "Decentralized Satellite Constellation Replanning for Event Observation," *Journal of Spacecraft and Rockets*, pp. 1–19, Jan. 2025.

[24] A. A. Jaramillo, B. Gorr, H. Gao, D. Selva, A. Mehta, Y. Sun, V. Ravindra, C. H. David, and G. H. Allen, "Decentralized Consensus-based Algorithms for Satellite Observation Reactive Planning with Complex Dependencies," in *AIAA SciTech Forum*. Orlando, FL: AIAA, Jan. 2025.

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

[26] A. Herrmann, M. Stephenson, and H. Schaub, "Reinforcement Learning For Multi-Satellite Agile Earth Observing Scheduling Under Various Communication Assumptions," in *AAS Rocky Mountain GN&C Conference*, Breckenridge, CO, Feb. 2–8, 2023.

[27] M. A. Stephenson and H. Schaub, "BSK-RL: Modular, High-Fidelity Reinforcement Learning Environments for Spacecraft Tasking," in *75th International Astronautical Congress*. Milan, Italy: IAF, Oct. 2024.

[28] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework," *Journal of Aerospace Information Systems*, vol. 17, no. 9, pp. 496–507, Sep. 2020.

[29] J. K. Terry, B. Black, N. Grammel, M. Jayakumar, A. Hari, R. Sullivan, L. Santos, R. Perez, C. Horsch, C. Dieffendahl, N. L. Williams, Y. Lokesh, and P. Ravi, "PettingZoo: Gym for Multi-Agent Reinforcement Learning," Oct. 2021.

[30] M. Towers, J. K. Terry, A. Kwiatkowski, J. U. Balis, G. de Cola, T. Deleu, M. Goulão, A. Kallinteris, A. KG, M. Krimmel, R. Perez-Vicente, A. Pierré, S. Schulhoff, J. J. Tai, A. J. S. Tan, and O. G. Younis, "Gymnasium," Oct. 2023.

[31] C. H. Acton, "Ancillary data services of NASA's Navigation and Ancillary Information Facility," *Planetary and Space Science*, vol. 44, no. 1, pp. 65–70, Jan. 1996.

[32] R. S. Sutton and A. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Adaptive Computation and Machine Learning. Cambridge, Massachusetts London, England: The MIT Press, 2018.

[33] S. Bradtke and M. Duff, "Reinforcement Learning Methods for Continuous-Time Markov Decision Problems," in *Advances in Neural Information Processing Systems*, vol. 7. MIT Press, 1994.

[34] R. S. Sutton, D. Precup, and S. Singh, "Between MDPs and semi-MDPs: A framework for temporal abstraction in reinforcement learning," *Artificial Intelligence*, vol. 112, no. 1-2, pp. 181–211, Aug. 1999.

[35] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, "High-Dimensional Continuous Control Using Generalized Advantage Estimation," in *International Conference on Learning Representations*, San Juan, Puerto Rico, Oct. 2018.

[36] A. Oroojlooy and D. Hajinezhad, "A Review of Cooperative Multi-Agent Deep Reinforcement Learning," Apr. 2021.

[37] K. Menda, Y.-C. Chen, J. Grana, J. W. Bono, B. D. Tracey, M. J. Kochenderfer, and D. Wolpert, "Deep Reinforcement Learning for Event-Driven Multi-Agent Decision Processes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 4, pp. 1259–1268, Apr. 2019.

[38] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," Aug. 2017.

[39] E. Liang, R. Liaw, P. Moritz, R. Nishihara, R. Fox, K. Goldberg, J. E. Gonzalez, M. I. Jordan, and I. Stoica, "RLlib: Abstractions for Distributed Reinforcement Learning," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, Jun. 2018, pp. 3053–3062.

BIOGRAPHY

Mark Stephenson received his B.S. in Mechanical Engineering from USC and M.S. in Aerospace Engineering from the University of Colorado, Boulder. He is currently a Ph.D. Candidate and NASA Space Technology Graduate Research Opportunity Fellow in the Autonomous Vehicle Systems Lab at the University of Colorado, Boulder, advised by Dr. Hanspeter Schaub. His research focuses on the application of deep reinforcement learning to spacecraft tasking and operations, including Earth observation scheduling and relative-motion-based inspection of space objects. Previously, he led the University of Southern California Rocket Propulsion Lab (USCRPL) and interned at Momentus Space and SpaceX.

Hanspeter Schaub is a distinguished professor and chair of the University of Colorado aerospace engineering sciences department. He holds the Schaden leadership chair. He has over 30 years of research experience, of which 4 years are at Sandia National Laboratories. His research interests are in astrodynamics, relative motion dynamics, charged spacecraft motion as well as spacecraft autonomy. This has led to about 228 journal and 371 conference publications, as well as a 4th edition textbook on analytical mechanics of space systems. Dr. Schaub has been the ADCS lead in the CICERO mission, the ADCS algorithm lead on a Mars mission and supporting ADCS for a new asteroid mission. In 2023 he won the Hazel Barnes Prize, the top award granted to faculty at the University of Colorado. He has been awarded the H. Joseph Smead Faculty Fellowship, the Provost's Faculty Achievement Award, the faculty assembly award for excellence in teaching, as well as the Outstanding Faculty Advisor Award. He is a fellow of AIAA and AAS, and has won the AIAA/ASEE Atwood Educator award, AIAA Mechanics and Control of Flight award, as well as the Collegiate Educator of the Year for the AIAA Rocky Mountain section. In 2025 he became a member of the National Academy of Engineering.