

IAC-26,C1,4,1,x117405

Autonomous Spacecraft Scheduling Yielding Scalable and Safe Earth and Space Object Imaging

Hanspeter Schaub^{a,*}

^a *Distinguished Professor, Glenn L. Murphy Endowed Chair, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, 431 UCB, Colorado Center for Astrodynamics Research, Boulder, CO, 80309. AAS Fellow, AIAA Fellow., hanspeter.schaub@colorado.edu*

* Corresponding Author

Abstract

Autonomous spacecraft scheduling is becoming increasingly important as Earth-observing missions transition from individually operated spacecraft to responsive, distributed constellations. Traditional ground-based scheduling methods can produce high-quality plans, but they are computationally expensive, open loop, and brittle to changing request sets, environmental uncertainty, and spacecraft subsystem degradation. Research over the last decade in the Autonomous Vehicle Systems (AVS) Laboratory at the University of Colorado Boulder has developed a progression of methods for autonomous spacecraft scheduling that begins with optimal scheduling benchmarks, advances to semi-Markov reinforcement-learning policies for onboard closed-loop scheduling, extends these policies to distributed constellation operations, and then improves their long-duration reliability through curriculum learning and shielded neural networks. This paper discusses that research program and synthesizes its major technical results. Particular emphasis is placed on the development of robust, fault-resilient, and shielded neural networks for onboard autonomous scheduling of agile Earth-observing and space-to-space observing satellites. The focus is on scalable autonomous scheduling architectures that can task either a single spacecraft or large distributed constellations. Shields provide an explicit safety layer between a neural-network policy and the spacecraft, preventing unsafe decisions such as battery depletion or reaction-wheel saturation. Recent results show that shielded policies can be deployed over episodes thirty times longer than their training episodes while maintaining safety and achieving higher cumulative reward through increased survivability. Newer shielded-training approaches, including action replacement, action masking, and interference penalties, further demonstrate that safety information can be incorporated into the learning process so that policies become less dependent on shield intervention.

Keywords: spacecraft autonomy, autonomous scheduling, agile Earth observing, autonomous space-to-space imaging, shielded neural networks

1. Introduction

The increasing demand for timely Earth-observation data, combined with the rapid growth of satellite constellations, is changing the way spacecraft missions must be operated. Traditional spacecraft operations rely heavily on ground-based planning systems. Requests are collected on the ground, visibility opportunities are predicted, spacecraft resource constraints are modeled, and a command sequence is generated and uploaded to the spacecraft. This paradigm has been successful for many missions, but it becomes increasingly difficult to scale as the number of spacecraft, imaging requests, ground contacts, and dynamic mission opportunities increases. Further, future missions are demanding an agile response capability to rapidly reschedule the flight tasks to account for changing environmental conditions or mission objectives that current offline scheduling tools cannot provide. Monitoring rapidly evolving natural disasters such as wildfires pro-

vides a representative motivating example.¹ Here the fire conditions can change rapidly and the satellite needs to quickly incorporate the latest information into its scheduling process while also maintaining safe satellite operation that does not deplete onboard resources.

Agile Earth-observing satellites complicate this planning problem further. Unlike traditional Earth-observing satellites that primarily slew across track, an agile Earth-observing satellite (AEOS) can slew both along and across track. This increased pointing freedom expands the feasible set of imaging opportunities, but it also creates a more difficult sequencing problem because both the order and timing of observations affect feasibility. The agile Earth-observing satellite scheduling problem (AEOSSP) is therefore highly combinatorial and is widely treated as an NP-hard scheduling problem, especially if 1000's of targets are being considered. Classical optimization approaches, including greedy methods, dynamic pro-

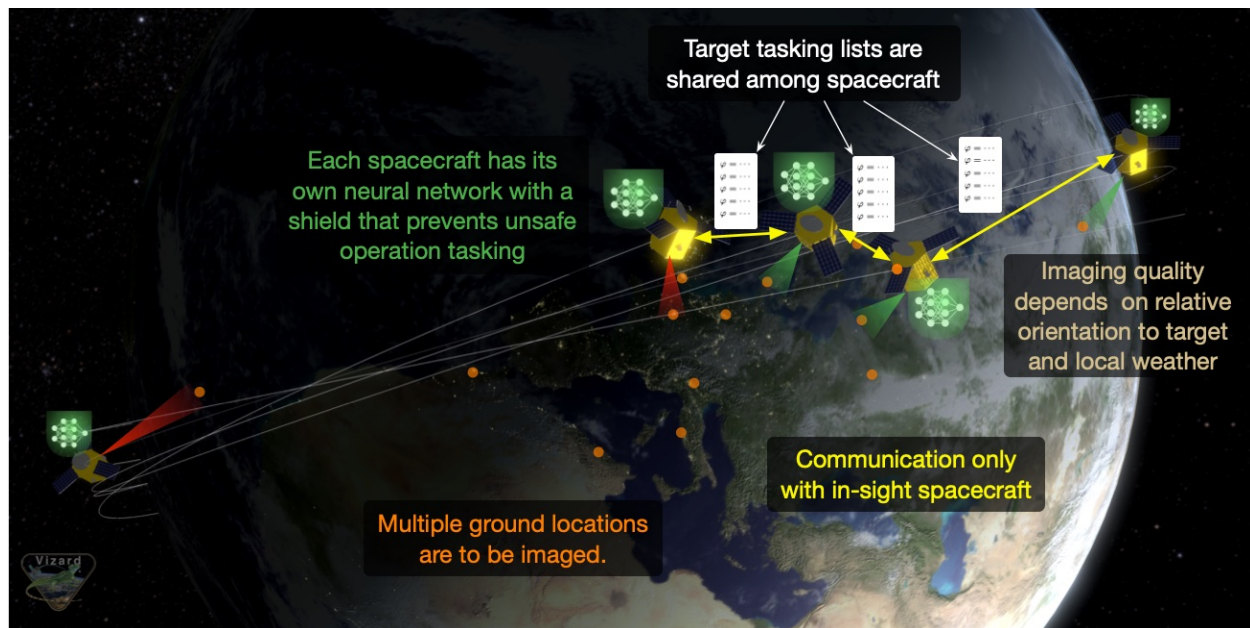


Fig. 1: Illustration of a cluster of satellites performing coordinated autonomous agile Earth-observation task scheduling.

gramming, local search, constraint programming, mixed-integer programming, and graph-based formulations, can produce high-quality schedules. However, they typically require an expensive ground-based planning step and produce open-loop plans.^{2,3}

Open-loop plans are vulnerable to changes in the operational environment. New high-priority imaging requests may arrive after the schedule is generated. Cloud cover may invalidate a planned observation.⁴ A downlink opportunity may be lost. Battery capacity, solar-array efficiency, and reaction-wheel behavior may differ from their modeled values or degrade over time. In these cases, an onboard scheduler that can react to the current spacecraft state is preferable to a static plan that must be repaired or recomputed on the ground.

Deep reinforcement learning offers one path toward onboard autonomous scheduling. In this approach, a neural-network policy is trained offline in simulation and then evaluated onboard in closed loop. Once trained, the policy evaluation is computationally inexpensive, allowing decisions to be made rapidly from the current spacecraft state. This makes deep reinforcement learning (DRL) attractive for responsive Earth observation, dynamic target lists, and multi-spacecraft coordination as illustrated in Fig. 1. The autonomous on-board scheduling work in this paper has been applied to an actual spacecraft mission in Ref. 5, illustrating flight-heritage of this proposed on-board scheduling approach and that the resulting neural networks are computationally suitable for flight processors.

Deploying neural networks in spacecraft operations raises safety and certification concerns. A policy trained

to maximize mission reward may discover aggressive strategies that perform well over short training horizons but eventually violate safety constraints during long-duration deployment. In Earth-observation scheduling, unsafe behavior can include depleting the battery, saturating reaction wheels, selecting imaging modes at inappropriate times, or failing to allocate time for recovery actions such as charging or momentum desaturation. These risks become more important when policies are deployed for episodes much longer than those used during training.

Research over the last decade in the Autonomous Vehicle Systems (AVS) Laboratory at the University of Colorado Boulder has addressed these challenges through a systematic sequence of developments.^{6–15} First, optimal^{3,12,16} and near-optimal scheduling methods^{8,17,18} were developed to provide performance benchmarks and realistic transition-time models. Second, the scheduling problem was reformulated as a semi-Markov decision process (sMDP)^{11,19} so that reinforcement-learning policies could make decisions over variable-duration task intervals. Third, multi-agent methods such as intent sharing were developed to scale autonomy to constellations.²⁰ Fourth, training environment enhancements such as curriculum learning and domain randomization were investigated to improve policy robustness over long deployments and degraded spacecraft conditions.¹³ Fault-resilience studies then evaluated how trained scheduling networks behave under reaction-wheel fault conditions and how fault-injected training, fault-aware observations, recurrent architectures, and fault-response logic can improve mission continuity.^{21,22} Finally, shielded neural networks were introduced to provide an explicit safety layer for au-

tonomous scheduling.¹⁵ A key enabler of this research has been the development of the Basilisk spacecraft simulation framework* and the accompanying BSK-RL reinforcement learning infrastructure†, which provide the high-fidelity simulation capability required to train and evaluate autonomous scheduling policies.

The novelty of this paper is the synthesis of a decade of AVS Lab research into a unified technical architecture for autonomous spacecraft scheduling, showing how optimal planning, semi-Markov reinforcement learning, distributed intent sharing, robustness training, fault resilience, and shielded neural networks collectively enable scalable and safe onboard tasking. Rather than treating shielded neural networks as an isolated safety mechanism, this paper places them in the broader progression from optimal planning to robust, safe, distributed autonomy. This perspective highlights an important conclusion: onboard spacecraft autonomy requires the integration of four capabilities. It must be performant relative to optimal planning, robust to environmental and subsystem variations, safe with respect to mission-critical constraints, and scalable to multi-spacecraft architectures.

2. The Autonomous Scheduling Problem

2.1 Agile Earth-Observation Tasking

In the Earth-observation scheduling problem, as illustrated in Fig. 1, a spacecraft is tasked with selecting a sequence of operational modes that maximizes mission utility while respecting spacecraft constraints. A typical AEOS scheduling environment includes imaging requests distributed over the Earth, each with a value or priority. The spacecraft must slew to image targets when they are visible, manage stored data volume, downlink collected imagery, maintain sufficient battery charge, and keep reaction-wheel speeds within allowable limits.

The action set in scheduling environments commonly includes four high-level operational modes:

1. imaging, in which the spacecraft slews to and images an available target from a large set of targets;
2. charging, in which the spacecraft points its solar arrays at the sun to replenish battery energy;
3. downlinking, in which stored data are transmitted to a ground station; and
4. momentum desaturation, in which reaction-wheel momentum is managed through an appropriate control mode.

The objective is not simply to image as many weighted Earth-fixed targets as possible. The policy must choose when to sacrifice immediate imaging reward in order to

preserve future mission capability. Charging and desaturation actions may not directly produce science reward, but they are essential for long-duration operation.

2.2 From MDPs to Semi-Markov Decision Processes

A standard Markov decision process is defined by the tuple

$$\mathcal{M} = \langle S, A, T, R, \gamma \rangle, \quad (1)$$

where S is the state space, A is the action space, $T(s'|s, a)$ is the transition model, $R(s, a)$ is the reward function, and γ is the discount factor. At each decision step, the agent observes a state or observation, selects an action, receives a reward, and transitions to a new state.²³

For spacecraft scheduling, the standard fixed-step MDP abstraction is restrictive because operational modes have variable durations. Imaging one target may require a different slew time than imaging another. Downlink duration depends on visibility, data volume, and ground-station geometry. Charging and momentum-management actions may also be selected for variable periods. A sMDP generalizes the MDP by allowing each transition to have a duration τ . The transition model becomes¹¹

$$T(s', \tau | s, a), \quad (2)$$

and the value of a policy must account for both reward accumulation and elapsed time. This is a better representation of mission-like operations because the policy selects event-based tasks rather than artificial fixed-duration actions.

In the AVS lab scheduling work, the sMDP formulation is particularly important because it allows reinforcement-learning policies to make decisions at variable intervals while still managing resources.¹¹ This moves the learning problem closer to the way an onboard scheduler would actually operate. It also helps reinforcement-learning policies compete more directly with graph-based and mixed-integer linear programming (MILP) scheduling methods, which naturally reason over task intervals and transitions. In particular, this is true when the scheduler is exposed to a very high target density where only small maneuvers are required to transition between targets.

3. Simulation Infrastructure for Autonomous Spacecraft Scheduling

The autonomous scheduling approaches discussed in this paper rely on machine learning methods that require enormous quantities of training data. Unlike many terrestrial applications, spacecraft operations do not generate sufficiently large public datasets from which autonomous decision-making systems can be trained. Most missions involve only a small number of vehicles, operational data are often proprietary, and many of the scenarios of interest have never previously occurred. Consequently, sim-

* <https://avslab.github.io/basilisk>

† https://avslab.github.io/bsk_rl/

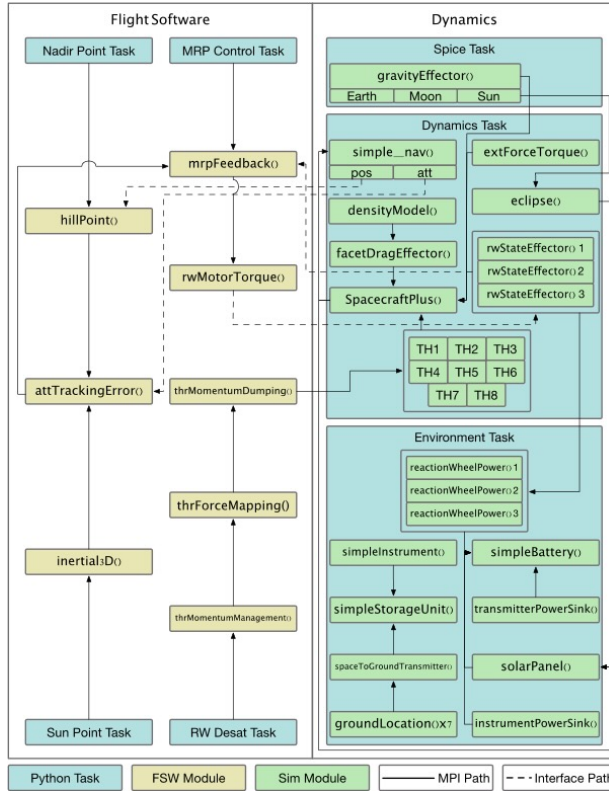


Fig. 2: Illustration of a Basilisk training simulation setup¹⁸

ulation becomes a fundamental enabling technology for spacecraft autonomy research.

This challenge is particularly important for reinforcement learning. Training campaigns routinely require millions of state-action interactions and may represent the equivalent of decades or centuries of simulated spacecraft operations. The resulting policies must learn not only mission tasking behavior but also the complex interactions between spacecraft dynamics, attitude control, power generation, communication opportunities, environmental conditions, and onboard resource management. Generating such data with sufficient fidelity and computational efficiency represents a significant research challenge in its own right.

To address this need, the AVS Lab has developed the open-source Basilisk astrodynamics simulation framework.^{24,25} Basilisk provides a modular spacecraft simulation environment capable of modeling translational and rotational dynamics,^{26–37} attitude determination and control systems, power systems, communication systems,³⁸ imaging payloads, reaction wheels, orbital environments, and multi-spacecraft interactions.³⁹ Figure 2 illustrates a representative Basilisk simulation architecture used for autonomous scheduling research. Unlike many traditional mission simulators, Basilisk was designed from the outset to support large-scale Monte Carlo analysis, autonomy research, and machine-learning applications while maintaining flight-relevant modeling fidelity.

Building upon Basilisk, the AVS Lab developed the BSK-RL framework⁴⁰ to provide a direct interface between high-fidelity spacecraft simulation and modern reinforcement-learning algorithms. BSK-RL supplies standardized spacecraft tasking environments, observation spaces, reward formulations, and evaluation tools while remaining compatible with widely used machine-learning frameworks such as RLlib. This architecture allows researchers to rapidly develop and evaluate new autonomous scheduling approaches without sacrificing simulation fidelity.

The combination of Basilisk and BSK-RL has served as the common research foundation for the autonomous scheduling developments summarized throughout this paper. The same infrastructure has supported studies in optimal scheduling, semi-Markov reinforcement learning, distributed constellation operations, strip imaging, space-domain-awareness tasking, curriculum learning, fault resilience, and shielded neural networks. By providing both computational efficiency and high-fidelity modeling, these tools have enabled the systematic investigation of scalable and safe spacecraft autonomy across a broad range of mission applications.

The availability of this simulation infrastructure enabled the development and rigorous evaluation of increasingly sophisticated autonomous scheduling architectures, beginning with optimal scheduling formulations and progressing toward reinforcement-learning-based on-board autonomy.

4. Optimal Scheduling Foundations

Before autonomous schedulers could be evaluated, strong reference solutions were required. This motivated the development of optimization-based scheduling methods capable of producing near-optimal schedules against which learning-based approaches could be compared. The AVS Lab has therefore developed optimization-based formulations of the AEOSSP that provide reference solutions and reveal the structure of the scheduling problem.³

4.1 Graph-Based Scheduling

Many agile-satellite scheduling approaches represent the problem as a directed graph. Vertices correspond to possible imaging opportunities, and edges correspond to feasible transitions between opportunities. An edge exists when the spacecraft can complete one observation, slew to the next target, and satisfy all relevant timing constraints. The objective is to select a feasible path through this graph that maximizes the cumulative value of fulfilled requests.

The graph formulation is attractive because it exposes the combinatorial structure of the problem. However, it also creates computational challenges. If many requests are available and each request has many possible fulfillment times, the number of vertices and edges can grow

rapidly. Sparse graph construction and carefully designed MILP formulations are therefore necessary to keep the problem tractable.

4.2 Learned Maneuver Dynamics

A central challenge in graph-based scheduling is estimating the transition time between imaging opportunities. Simplified slew-rate models can be computationally convenient but may misrepresent the true spacecraft maneuver time. This mismatch can lead to schedules that appear feasible in the planner but perform poorly when executed in a high-fidelity spacecraft simulation.

To address this problem, Stephenson and Schaub developed learned maneuver-transition models trained on simulated attitude-maneuver data.³ A neural network predicts dynamically accurate, time-dependent slew durations between observation opportunities. These learned transition times are then incorporated into sparse graph and MILP formulations. This approach preserves the strengths of optimization-based scheduling while reducing the gap between the planning model and flight-like simulation.

The resulting optimal scheduling framework plays two roles in the broader AVS autonomy program. First, it provides strong performance benchmarks for reinforcement-learning schedulers. Second, it demonstrates that machine learning can be used not only as a policy generator but also as a surrogate model inside traditional optimization pipelines.

5. Reinforcement Learning for Onboard Scheduling

5.1 Motivation for Closed-Loop Policies

The transition from optimization-based planning to reinforcement-learning-based scheduling represents a fundamental shift in how spacecraft tasking is performed. Rather than generating a complete schedule before execution, the scheduler continuously makes decisions onboard using the current spacecraft state.

Optimization methods can produce high-quality schedules, but they are fundamentally limited by computation time and the fragility of open-loop execution. If a new target appears, a planned observation fails, or spacecraft resources evolve differently than expected, the schedule may need to be repaired or regenerated. This motivates closed-loop policies that select actions onboard using the current state.

Reinforcement learning directly trains such policies. The policy $\pi_\theta(a|o)$ maps an observation o to a distribution over actions using parameters θ . During training, the policy interacts with a simulation environment and updates its parameters to maximize expected cumulative reward,

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[\sum_{k=0}^{\infty} \gamma^k r_k \right]. \quad (3)$$

In the AVS lab autonomous scheduling studies, proximal policy optimization is commonly used because it provides stable policy-gradient updates while remaining practical for complex simulation environments.^{10,11}

5.2 Semi-Markov Reinforcement Learning Results

Reference 11 shows that policies trained over sMDP formulations can perform competitively against optimal global schedules in a flight-like evaluation environment. This result is important because earlier fixed-duration RL formulations can underperform when task durations vary significantly. By giving the policy timing information and allowing variable decision intervals, the sMDP formulation enables the agent to reason about the opportunity cost of each task.

The learned policies offer several practical advantages. They are computationally inexpensive at deployment because the onboard scheduler only evaluates a neural network. They operate in closed loop, so they can respond to the current battery state, reaction-wheel speeds, stored data, target availability, and other local conditions. They can also implicitly learn resource-management strategies that are difficult to encode exactly in simplified optimization models.

This work established a key premise for the subsequent shielded-neural-network research: reinforcement learning can produce useful onboard schedulers, but additional mechanisms are needed to ensure robust and safe long-duration operation.

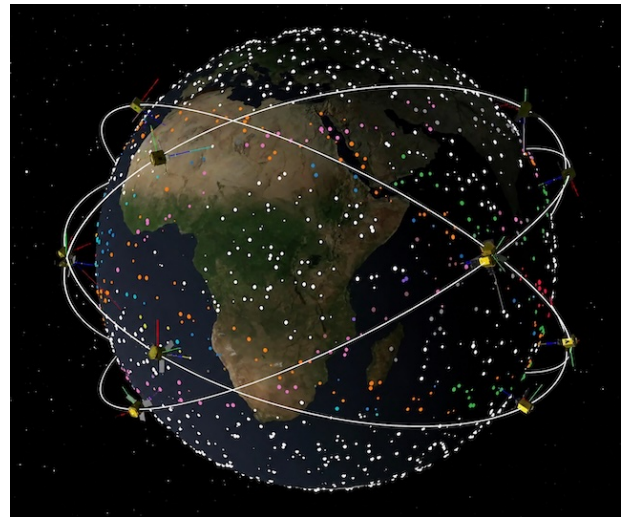


Fig. 3: Illustration of a constellation of satellites collaborating in agile Earth observations. White markers are un-imaged locations, colored markers have been imaged.

6. Distributed Constellation Scheduling

Once reinforcement-learning schedulers demonstrated competitive performance for single-spacecraft Earth observation, the next challenge was extending the frame-

work to increasingly complex mission classes. These extensions included distributed constellations, strip-imaging missions, space-domain-awareness applications, and autonomous spacecraft inspection.

6.1 From Single Spacecraft to Constellations

The value of onboard autonomous scheduling increases in multi-spacecraft missions. A constellation can respond more flexibly than a single spacecraft, but it also introduces the possibility of duplicated effort, conflicting assignments, and inefficient use of resources. Centralized constellation scheduling can be effective but becomes increasingly expensive as the number of spacecraft and requests grows.

A scalable alternative is to deploy decentralized policies on each satellite. In this architecture, each spacecraft runs its own local policy using its own observation vector. The policy may be trained in a single-agent setting and then deployed across multiple satellites. This avoids the need to train a unique policy for every constellation size or geometry. Figure 3 illustrates a Basilisk simulation of a constellation sharing the Earth observation tasks. The simulation data is visualized using the Vizard tool.⁴¹ Each spacecraft is responsible for its own onboard resource management and safe operation. The image tasking effort, however, should be shared and coordinated across all spacecraft in the constellation.

6.2 Intent Sharing for Emergent Collaboration

A key AVS lab contribution to distributed scheduling is the intent-sharing architecture.^{12,17,20} Each spacecraft communicates the target or task it intends to complete next. Other spacecraft include this shared intent information in their local observations and can avoid selecting the same request. This induces cooperation without requiring a centralized optimizer.

Intent sharing is especially useful because satellites often make decisions asynchronously. In a variable-duration scheduling problem, one satellite may complete an imaging action while another is still downlinking or charging. A practical distributed scheduler must therefore handle asynchronous decision intervals and stale information. Reference 12 shows that intent sharing can produce effective collaboration under these conditions and can approach globally optimal request allocation over a variety of constellation configurations and request distributions. A video illustrating BSK-RL trained policies using intent sharing is available online.* This methodology yields a very scalable autonomous scheduling solution as visualized in this video.[†]

* <https://www.youtube.com/watch?v=LCN0TiNJ1i4&t=2s>

† https://www.youtube.com/watch?v=Ia_1dXj_pP4&t=56s

The distributed scheduling work expands the relevance of shielded autonomy. In a constellation, a single failed spacecraft can reduce mission capability, and repeated unsafe local decisions can propagate into global performance losses. Robustness and safety therefore become constellation-level concerns, not merely single-spacecraft design features.

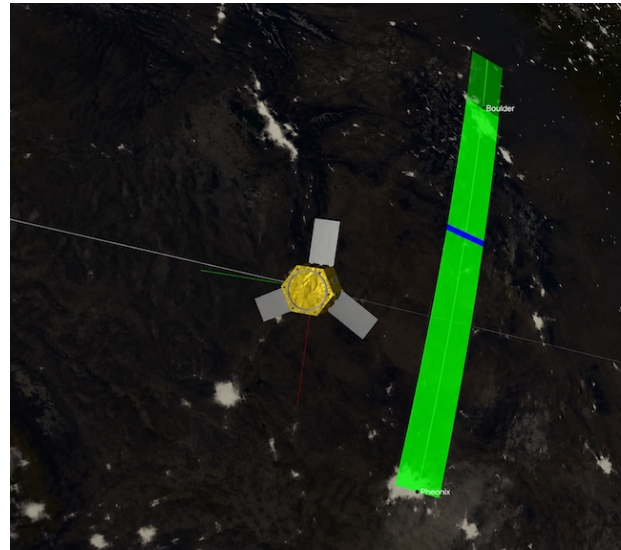


Fig. 4: Illustration of autonomous strip imaging on Earth.

6.3 Autonomous Strip Imaging Scheduling

The initial autonomous scheduling research in the AVS Lab focused on point-target (i.e. spot) imaging tasks, where the scheduler selects individual imaging opportunities from a discrete set of requests. Recent work by Cheval and Schaub extended this framework to support strip-imaging operations using super-agile Earth-observing satellites.⁴² Unlike point imaging, strip imaging requires the spacecraft to continuously track a moving reference point along an extended ground path while simultaneously satisfying attitude, field-of-view, and image acquisition rate constraints. This capability is particularly important for monitoring elongated features such as coastlines, borders, rivers, wildfire fronts, and flood regions.

To support strip imaging, a dedicated guidance and control architecture was developed and integrated into the Basilisk simulation environment. The scheduling problem was formulated as a partially observable semi-Markov decision process, allowing the policy to reason about both task selection and the timing associated with imaging opportunities. A custom reinforcement-learning environment was implemented within BSK-RL, enabling the training and evaluation of onboard autonomous strip-imaging policies as illustrated in Figure 4.*

* <https://www.youtube.com/watch?v=x5iylxf6k6A>

Subsequent work introduced a hybrid-action reinforcement-learning architecture in which the scheduler simultaneously selects the next strip-imaging task and estimates the required transition time before image acquisition.⁴³ This Hybrid Observation-task Planning Oriented Proximal Policy Optimization (HOP-PPO) framework combines a dual-actor architecture with transformer-based observation encoding to better capture relationships among candidate imaging tasks. Results demonstrated significant improvements over heuristic approaches and conventional hybrid-action reinforcement-learning methods, illustrating that autonomous scheduling can be extended beyond discrete imaging requests to more complex continuous observation tasks.

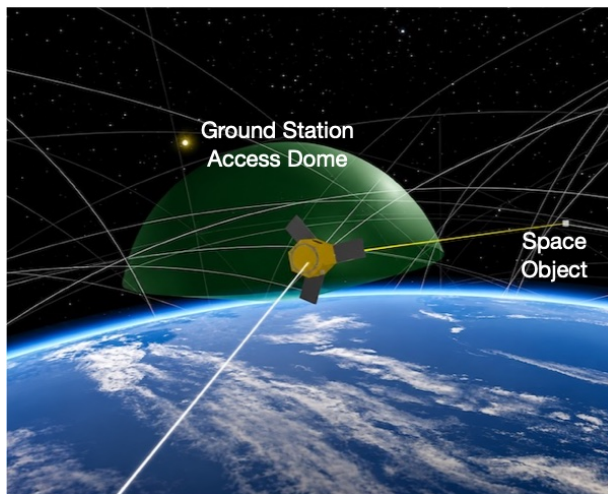


Fig. 5: Illustration of a satellite imaging other resident space objects and transmitting the data to a ground station.

6.4 Autonomous Space-to-Space Imaging for Space Domain Awareness

The autonomous scheduling framework has also been extended beyond Earth observation to support space-based space surveillance and space domain awareness applications. Unlike Earth-observation missions, where targets are fixed on the Earth's surface, space-based surveillance requires scheduling observations of resident space objects (RSOs) that are themselves moving through a wide range of orbital regimes. Imaging opportunities are therefore highly dynamic and depend on rapidly changing relative geometry, illumination conditions, line-of-sight constraints, and onboard resource limitations.

Initial work by Huterer Prats, Schaub, and Wheeler formulated the space-to-space imaging problem as a partially observable Markov decision process and trained reinforcement-learning policies to autonomously select imaging targets while simultaneously managing spacecraft resources such as battery charge, onboard stor-

age, reaction-wheel momentum, and downlink opportunities.⁴⁴ Using the Basilisk and BSK-RL framework, the resulting policies demonstrate the ability to generalize across different orbital configurations, exploit eclipse periods for communication activities, and outperform heuristic scheduling approaches while maintaining spacecraft safety constraints. A Basilisk simulation visualization of a satellite doing space-to-space imaging is shown in Figure 5.[†] A notable aspect of this work is the explicit modeling of onboard data storage for the images are taken, as well as the need to transmit the image data to a ground station. The autonomous scheduler is able to balance the mission image taking goals with the requirements of scheduling downlink times. This work demonstrates that reinforcement learning can support responsive onboard scheduling for space-based surveillance missions while accounting for realistic spacecraft operational constraints.

More recent work expanded the problem from LEO-to-LEO imaging to a unified multi-regime surveillance environment in which a single LEO-based imaging platform schedules observations of targets distributed across LEO, MEO, and GEO orbital regimes.³⁸ In addition to autonomous target selection, the framework introduces explicit tradeoffs between image acquisition and data delivery through a multi-objective reward formulation. Results show that policies trained in LEO-only environments transferred effectively to mixed-regime target catalogs without retraining while maintaining stable imaging performance and improving image-delivery metrics. These studies demonstrate that reinforcement-learning-based autonomy can support future space-domain-awareness missions operating in increasingly congested orbital environments while balancing collection, resource management, and timely data delivery.

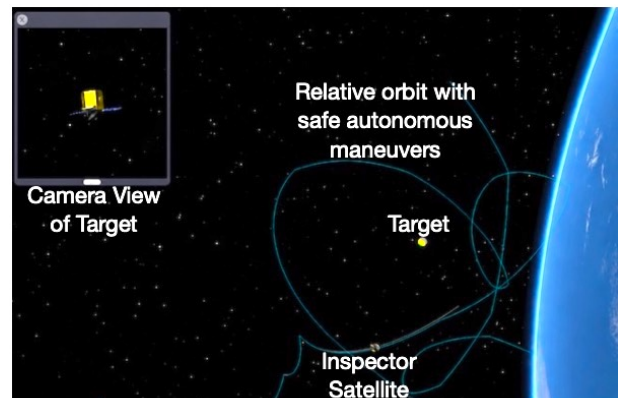


Fig. 6: Illustration of an inspector satellite making safe autonomous maneuvers about a target object to image it from all sides.

[†] <https://www.youtube.com/watch?v=Aas0z43uS9M>

6.5 Safe Autonomous Spacecraft Circumnavigation and Inspection

The same safety-autonomy principles discussed in prior sections extend to other spacecraft tasking problems. The AVS Lab research on autonomous resident space object inspection provides an example.^{45,46} In this problem, one or more servicer spacecraft must maneuver around a target object, inspect illuminated surfaces, avoid collision with the object and other inspection satellites, and satisfy fuel and operational constraints. This work differs from AEOS in that the action space is not a discrete set of flight modes, but rather a continuous action space that needs to estimate safe and suitable Δv burns and coast durations. Figure 6 illustrates a single satellite performing autonomous maneuvers about a target vehicle to position itself to inspect illuminated components. A video illustration is available online.*

Reinforcement learning is attractive for this problem because it can produce closed-loop inspection policies in complex dynamical environments. However, unshielded RL is again problematic because collision avoidance is safety critical. The AVS inspection work therefore develops projection-based shields that modify proposed impulsive maneuvers when they would violate keep-out constraints over the next few orbits.⁴⁵ Unlike discrete scheduling shields for battery and reaction-wheel management, these inspection shields operate in continuous relative-motion space and use analytical dynamics and optimal projection to find the nearest minimally modified safe actions.

This extension shows that shielded autonomy is not limited to a single scheduling application. The broader pattern is reusable: a learning-based policy proposes high-value actions, while a model-based safety mechanism enforces mission-critical constraints. For resource scheduling, the shield may be a safety MDP or rule-based supervisor. For proximity operations, it may be an optimization problem over relative orbital dynamics. In both cases, the shield provides a safety layer that makes learning-based autonomy more credible for spacecraft deployment.

7. Robustness for Long-Duration Operations

7.1 The Training–Deployment Horizon Gap

A major challenge in reinforcement-learning-based spacecraft autonomy is the difference between training and deployment horizons. Training episodes are often short because high-fidelity simulation is computationally expensive. However, spacecraft must operate for many orbits, days, months, or years. A policy that performs well over a three-orbit training episode may still fail over a ninety-orbit or longer deployment.

This horizon gap is particularly important for resource management. An aggressive imaging policy can appear successful in a short episode because it collects many rewards before resource depletion occurs. Over longer episodes, the same policy may eventually drain the battery, saturate reaction wheels, or enter unrecoverable states. Long-duration testing is therefore essential for evaluating whether an autonomous scheduler has learned sustainable operational behavior.

7.2 Enhanced Training Environments

A natural question following the development of closed-loop schedulers is whether policies trained over a few orbital periods remain effective during long-duration mission operations. This training–deployment gap motivated the investigation of curriculum learning and domain randomization. Reference 13 investigates curriculum learning and domain randomization to improve robustness of autonomous scheduling policies. The central idea is to expose the agent during training to progressively more difficult or more diverse environments. Instead of training only in a nominal spacecraft model, the policy is trained with degraded battery capacity, increased external disturbance torques, reduced subsystem capability, or randomized environmental parameters. The policy is intentionally exposed to operating conditions significantly more challenging than those expected during nominal mission operations. In particular, including no-win mission situations created a more resilient network.

This process encourages the policy to learn slightly more conservative and generalizable resource-management strategies. It also helps address the problem of heterogeneous constellations. Even if spacecraft begin as identical vehicles, their battery capacity, solar-array efficiency, disturbance environment, and actuator health can diverge over time. Training a single policy that remains effective across such broad variations is more practical than maintaining separate policies for every spacecraft and degradation state.

The robustness studies show that enhanced training environments can improve survivability and mission performance when policies are deployed for episodes much longer than those used during training. They also reveal that training-environment design is a first-order driver of policy reliability. Robustness is not simply a property of the neural-network architecture; it is shaped by the distribution of states, failures, and degradations encountered during training. However, even the most robust network training cannot guarantee that an unsafe action is never taken. Curriculum learning does not negate the need for shielded neural networks during deployment. Curriculum learning ensures near-optimal scheduling that does not continuously depend on the shield to keep the satellite operating safely.

* <https://www.youtube.com/watch?v=eQEoTOYADKc&t=31s>

7.3 Fault Resilience and Adaptive Autonomy

While curriculum learning and shielding improve robustness to environmental variations and off-nominal operating conditions, spacecraft must also remain operational in the presence of hardware faults. Recent work in Reference 22 extends the autonomous scheduling framework to investigate the resilience of reinforcement-learning-based schedulers under reaction-wheel faults. Several representative fault modes were considered, including complete reaction-wheel failures, power limitations, increased friction, and encoder measurement faults. This work demonstrates the resilience of the neural networks in providing autonomous scheduling, and at what hardware degradation level the scheduling begins to fail to operate safely or effectively.

These findings highlight an important distinction between robustness and fault tolerance. Robust policies can generalize across a range of environmental and operational uncertainties, but severe hardware failures may still require dedicated fault-detection, isolation, and recovery mechanisms. Consequently, future autonomous spacecraft architectures will likely combine reinforcement-learning-based decision making with traditional spacecraft health-management systems to achieve both adaptability and fault resilience.

8. Shielded Neural Networks for Safe Scheduling

8.1 Motivation for Shielding

Curriculum learning improves robustness but does not provide formal safety guarantees. A policy may become less likely to select unsafe actions, but there is still no deterministic mechanism preventing a neural network from violating constraints. For safety-critical spacecraft operations, this limitation motivates the addition of a shield.

A shield is a supervisory system placed between the policy and the spacecraft. The policy proposes an action $a = \pi(o)$ from observation o . The shield evaluates whether that action is safe from the current state. If the action is safe, it is passed to the spacecraft. If the action is unsafe, the shield replaces it with an action that satisfies the safety specification. The deployed action is written as

$$a_{\text{exec}} = \mathbb{S}(s, a_{\text{policy}}), \quad (4)$$

where $\mathbb{S}$ is the shield operator.

For autonomous Earth-observation scheduling, safety constraints commonly include

$$E(t) > E_{\min}, \quad (5)$$

$$|\Omega_i(t)| < \Omega_{\max}, \quad i = 1, \dots, N_w, \quad (6)$$

where $E(t)$ is battery energy and Ω_i are reaction-wheel speeds. Depending on the formulation, angular-rate constraints, storage constraints, and other subsystem limits may also be included.

8.2 Handmade and Formal Shields

Recent AVS Lab work compares several types of shields.¹⁴ A handmade shield uses expert knowledge to identify unsafe situations and select recovery actions. For example, if the battery state is low and a proposed action would further reduce power, the handmade shield may force the spacecraft to charge. Such shields can be simple and effective, and they may interfere less often than more conservative formal methods. However, because they are based on hand-designed logic, they may miss edge cases.

Formal shields are constructed from abstract safety models.⁴⁷ A safety Markov decision process is built using safety-relevant states, such as battery level and reaction-wheel speeds. Transition probabilities are estimated by sampling the high-fidelity simulator. Dynamic programming is then used to identify safe actions or to compute the probability that an action will preserve safety. The Q-Optimal shield is one example. It allows only actions whose safety value exceeds a specified threshold, and if no proposed action is acceptable, it selects the action most likely to preserve safety.

This comparison reveals a common trade-off. Handmade shields can be less conservative and therefore may collect more reward or interfere less often, but they do not necessarily provide mathematical safety guarantees. Formal shields can be more conservative, but they provide stronger assurance that the spacecraft remains within the specified safe set.

8.3 Long-Term Performance of Shielded Policies

The performance-evaluation studies of shielded neural networks demonstrate the practical value of shielding in long-duration autonomous scheduling.¹⁴ Policies trained over short episodes were deployed in episodes thirty times longer than training. In these tests, shielded agents often showed lower instantaneous reward rates because the shield occasionally forced recovery actions instead of allowing aggressive imaging. However, shielded agents survived longer, and their cumulative reward was higher because they continued operating after unshielded agents entered terminal states.

This result is central to the AVS Lab shielded-autonomy narrative. A policy that maximizes short-term imaging reward is not necessarily the policy that maximizes mission return. Over long horizons, survival is a prerequisite for productivity. Shielding changes the optimization problem from short-term reward maximization to sustained mission utility.

The studies also show that shield effectiveness depends on the underlying policy. Policies trained with some awareness of safety require fewer shield interventions than policies trained in environments without meaningful safety constraints. Thus, again, shielding should not be viewed as a replacement for good policy training. The

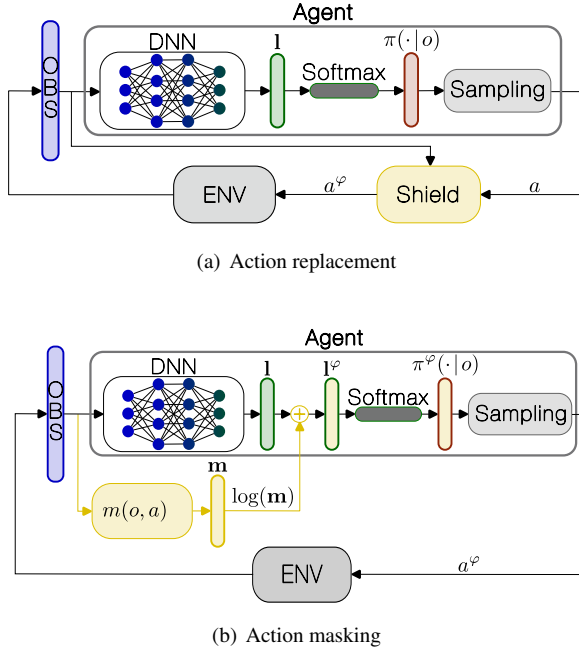


Fig. 7: Illustration of Shielding Methodologies.

best results arise when the policy and shield are designed together.

9. Learning Safe Behavior During Training

9.1 Deployment-Only Shielding versus Shielded Training

A shield can be used only during deployment, or it can be integrated into the training loop. Deployment-only shielding has the advantage of simplicity: the policy is trained normally and the shield is added afterward to enforce safety. However, if the policy frequently proposes unsafe actions, the shield may intervene often. High shield interference can reduce performance and indicates that the neural network has not internalized the safety structure of the problem.

Shielded training attempts to address this issue by exposing the policy to the shield during learning. The objective is not merely to prevent unsafe actions during training, but to teach the agent to avoid them. Recent AVS Lab work compares action replacement, action masking, and interference-penalty approaches in the AEOS scheduling context in Reference 15.

9.2 Action Replacement

In action replacement, the policy selects an action, but the shield replaces it before execution if it is unsafe.⁸ The environment therefore remains safe during training. However, the policy update can still be associated with the originally selected action. This allows the learning algorithm to observe that the proposed action caused shield intervention.

An interference penalty can be added whenever the

shield overrides the policy:¹⁵

$$r'_k = r_k - \lambda I(a_k \neq a_{\text{exec},k}), \quad (7)$$

where λ is the interference penalty and $I(\cdot)$ is an indicator function. A small penalty may be ignored by the agent, allowing it to rely on the shield. A very large penalty can produce overly conservative behavior or prevent effective learning. Moderate penalties provide a more useful learning signal because they penalize unsafe proposals without dominating the task reward.

Recent results show that action replacement with moderate interference penalties can significantly improve survivability when policies are later tested without shields, and it can reduce shield interference when shields are used during deployment.¹⁵ This indicates that the agent learns part of the safety structure instead of merely being corrected by an external filter.

9.3 Action Masking

In action masking, unsafe actions are removed from the policy's action distribution before sampling. If the neural network produces logits z_i for actions a_i , the mask sets unsafe logits to a very negative value,

$$\tilde{z}_i = \begin{cases} z_i, & a_i \in A_{\text{safe}}(s), \\ -\infty, & a_i \notin A_{\text{safe}}(s), \end{cases} \quad (8)$$

so that the resulting softmax assigns zero probability to unsafe actions. This guarantees safe action sampling during training, but it can also hide unsafe actions from the policy. Because the agent never selects masked actions, it may not learn why those actions were unsafe.

The comparison between masking and replacement is therefore nuanced.¹⁵ Masking can produce good performance because the agent always acts within the shielded action set. Replacement with interference penalties may produce better internal safety learning because unsafe proposals generate explicit feedback. The most appropriate method may depend on the mission, the quality of the shield, and the desired balance between policy autonomy and shield dependence.

10. Synthesis: Optimality, Robustness, Safety, and Scalability

The AVS Lab autonomous scheduling body of research can be understood as a progression through five interdependent goals.

10.1 Optimality

Optimization-based scheduling and learned-dynamics MILP formulations provide strong baselines. They answer the question: how good can a schedule be if the problem is solved with an accurate model and sufficient computation? These baselines are essential because they allow reinforcement-learning policies to be evaluated quantitatively rather than qualitatively.

10.2 Closed-Loop Autonomy

Semi-Markov reinforcement-learning policies address the limitations of open-loop scheduling. They make onboard decisions at variable task intervals and respond to the current spacecraft state. This enables rapid replanning and dynamic resource management without solving a large optimization problem onboard.

10.3 Robustness

Curriculum learning and domain randomization address the gap between training and deployment. They expose policies to degraded and diverse conditions so that learned strategies generalize beyond nominal short-horizon simulations. Fault-resilience studies extend this idea by exposing policies to explicit actuator degradations and by evaluating when learned adaptation should be combined with conventional fault-response actions, such as shutting down a failed reaction wheel. Robustness is especially important for long-duration missions and heterogeneous constellations.

10.4 Safety

Shielded neural networks add an explicit safety layer. They prevent neural-network policies from executing actions that violate critical spacecraft constraints. Shielded training then goes further by teaching policies to avoid unsafe actions, reducing their dependence on shield intervention.

10.5 Scalability

Distributed constellation methods such as intent sharing allow autonomous scheduling to scale beyond a single spacecraft. Limited communication enables emergent cooperation without centralized optimization. As constellations grow, this scalability becomes essential for practical mission operations.

Together, these developments suggest a mature architecture for autonomous spacecraft scheduling. An onboard scheduler should be trained in a high-fidelity environment, benchmarked against optimal planning methods, exposed to degraded conditions, protected by a shield during deployment, and designed to coordinate with other spacecraft through limited communication.

11. Future Research Directions

Several open problems remain before shielded autonomous scheduling can be routinely deployed in flight systems.

11.1 Information-Aware Autonomous Scheduling

Most current autonomous scheduling formulations assume that target locations are known with sufficient accuracy for successful observation. This assumption breaks down for many future applications involving dynamic

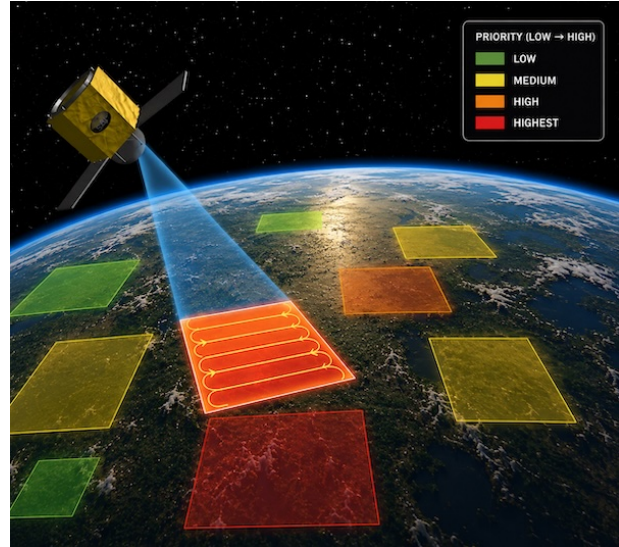


Fig. 8: Autonomous search of uncertain target regions with different priorities

and poorly characterized targets. An important future research direction is the incorporation of target-state uncertainty directly into the scheduling process.⁴⁸ Most current formulations assume that target locations are known with sufficient accuracy to permit successful observation. However, many applications, including maritime surveillance, airborne tracking, and space-domain-awareness missions, involve targets whose state uncertainty evolves over time and grows in the absence of observations.

In such scenarios, the scheduling objective extends beyond maximizing observation value. The spacecraft must also manage uncertainty across the target catalog to prevent poorly observed targets from becoming difficult or impossible to reacquire. Consequently, future schedulers should balance the collection of high-value observations with the need to revisit uncertain targets and maintain accurate state estimates as shown in Figure 8.

One promising approach is to augment the scheduling framework with uncertainty metrics derived from onboard estimation algorithms, such as covariance measures, entropy metrics, or information gain estimates. The reward function can then account for both observation value and uncertainty reduction, transforming the problem into an active sensing and information-gathering task. Such an approach is particularly relevant for future space-domain-awareness missions, where sensing resources should be allocated not only to maximize observations but also to maintain acceptable uncertainty levels across the resident space object catalog.

11.2 Foundation Models and Transformer-Based Spacecraft Autonomy

Recent advances in machine learning suggest that transformer architectures and foundation-model concepts may provide a promising path toward the next generation of

autonomous spacecraft scheduling systems. Most current reinforcement-learning schedulers rely on fixed observation structures in which a predefined number of candidate targets, spacecraft states, and resource variables are presented to the policy. While effective for many applications, these architectures become increasingly difficult to scale as the number of targets, spacecraft, sensors, and operational constraints grows.

Transformer architectures offer an attractive alternative because they naturally process variable-length input sets and can learn relationships between objects through attention mechanisms. This capability is already beginning to appear in autonomous spacecraft scheduling research. For example, the HOP-PPO strip-imaging framework employs transformer-based observation encoding to capture relationships among candidate imaging opportunities and improve decision making in large target sets. Similar architectures may enable future schedulers to reason over thousands of potential targets, distributed constellations, communication networks, and heterogeneous sensing assets simultaneously.

Beyond scheduling, foundation-model concepts offer the possibility of learning generalized representations of spacecraft operations from large collections of simulated missions. Rather than training a new policy for each mission type, a foundation model could learn common spacecraft operational behaviors and then be adapted to specific applications through fine tuning. Such an approach could reduce training costs, improve transfer learning between missions, and provide greater robustness when deployed in previously unseen environments.

An especially promising direction is the integration of transformer architectures with information-aware scheduling and state-estimation uncertainty. Attention mechanisms provide a natural framework for prioritizing observations based not only on target value but also on uncertainty growth, expected information gain, and mission objectives. This would allow future autonomous schedulers to reason simultaneously about task execution, resource management, target tracking, and uncertainty reduction within a unified decision-making framework.

Although significant challenges remain in verification, training efficiency, and flight qualification, transformer-based architectures and foundation models have the potential to substantially expand the scope of onboard spacecraft autonomy and enable scheduling systems capable of operating across diverse mission classes with minimal retraining.

11.3 Integrated Ground- and Space-Based Sensor Tasking

Another promising direction is the extension of autonomous scheduling and communication architectures beyond individual spacecraft and constellations to dis-



Fig. 9: Illustration of autonomous space object tracking using both space and ground assets.

tributed sensing networks composed of both space- and ground-based assets as shown in Figure 9. Current intent-sharing approaches focus on coordination among spacecraft, but similar concepts could be applied to heterogeneous sensor networks in which space-based imagers, ground telescopes, radar systems, and other sensing assets collaborate to maintain situational awareness.

This capability is particularly relevant for future space-domain-awareness missions. Rather than treating spacecraft and ground sensors as independent systems, future architectures could autonomously coordinate sensing assignments across the entire network. Such coordination would enable sensors to share target-state estimates, uncertainty metrics, and planned observations while dynamically allocating sensing resources where they provide the greatest reduction in uncertainty.

A key research challenge is the development of integrated scheduling frameworks that jointly optimize target search, target reacquisition, and target characterization. In these scenarios, observations would be selected not only to maximize imaging opportunities, but also to prevent uncertainty from growing beyond acceptable levels. Ground-based and space-based sensors possess complementary viewing geometries, fields of regard, and measurement characteristics, making them well suited for cooperative information-gathering strategies. The resulting architectures would extend current autonomous scheduling methods from spacecraft tasking problems toward distributed information-management systems capable of maintaining persistent awareness of large resident space object catalogs and other dynamic target populations.

11.4 Formal Verification and Certification

Formal shields provide safety guarantees within an abstracted model, but the relationship between the abstraction and the full spacecraft system must be carefully vali-

dated. Future work should quantify the conservatism and failure modes introduced by abstraction, discretization, and simulator sampling. Certification pathways will likely require evidence that the shield remains valid under model uncertainty and implementation constraints.

11.5 Uncertainty-Aware Shields

Most shields evaluate safety using estimated state variables and simplified transition models. Real spacecraft have uncertain state estimates, uncertain disturbance environments, and uncertain actuator behavior that can change with time. If the shields were trained on how a battery recharges when it is new, then years into the mission the battery charging performance could be degraded and the trained shields no longer can provide the formal safety guarantees. Future shields should incorporate probabilistic margins or robust optimization such that safety guarantees remain meaningful under uncertain time-varying spacecraft performance.

11.6 Adaptive and Degradation-Aware Policies

Spacecraft properties change over time. Battery capacity decreases, actuator performance changes, sensors degrade, and thermal behavior evolves. Reaction-wheel fault studies show that explicit fault information can improve scheduling performance when the fault is mild or partially recoverable, while severe failures may be better handled by discrete reconfiguration before the scheduler continues operation. A promising future direction is to combine shielded RL with onboard fault detection, online system identification, and adaptive policy conditioning so that the scheduler can respond to changing spacecraft capability while remaining safe.

11.7 Constellation-Level Safety

Most scheduling shields enforce single-spacecraft resource constraints. In constellations, safety and mission performance may also depend on network-level properties such as communication load, duplicated observations, inter-satellite interference, collision risk, and coordinated downlink behavior. Extending shielding concepts to distributed multi-agent systems is therefore an important next step.

11.8 Integration with Flight Software

Finally, shielded scheduling policies must be integrated into flight software architectures. This requires attention to timing, memory, deterministic execution, fault management, command authority, and operator override. High-fidelity environments such as Basilisk and modular RL interfaces such as BSK-RL provide important stepping stones, but flight implementation will require additional engineering work.^{49–53}

12. Conclusions

This paper summarized a decade of research in the Autonomous Vehicle Systems Laboratory toward scalable and safe autonomous spacecraft scheduling. The work progressed from optimization-based scheduling methods with learned maneuver dynamics, to semi-Markov reinforcement-learning schedulers capable of closed-loop onboard decision making, to distributed constellation operations through intent sharing. Subsequent developments improved long-duration performance through curriculum learning, fault-resilience techniques, and shielded neural networks that provide explicit protection against critical spacecraft constraint violations. More recently, these methods have been extended beyond traditional Earth observation to include strip-imaging missions, space-domain-awareness applications, and autonomous spacecraft inspection, demonstrating the flexibility of reinforcement-learning-based autonomy across a broad range of mission classes.

A central conclusion emerging from this research is that future spacecraft autonomy should not be viewed as a choice between learning-based decision making and model-based safety mechanisms. Instead, the most effective architectures combine the strengths of both approaches. Reinforcement learning provides adaptive, mission-driven decision making, while shields and other model-based methods enforce safety-critical constraints. When combined with robust training methodologies and scalable multi-agent coordination, these capabilities provide a practical foundation for future autonomous spacecraft operations.

Acknowledgments

Many thanks to the many graduate students who have contributed to this body of work. Special thanks to Andrew Harris, Adam Herrmann, Islam Nazmy, Mark Stephenson, Lorenzo Quevedo Mantovani, Yumeka Nagano, Daniel Huterer Prats, and Anais Cheval.

REFERENCES

- [1] Ryan S. Schaefer and Paul T. Grogan. Collaborative constellation analysis framework for wildfire observing missions. In *2024 IEEE Aerospace Conference*, pages 1–11, Big Sky, MT, USA, March 2024. IEEE.
- [2] Duncan Eddy and Mykel J. Kochenderfer. A maximum independent set method for scheduling earth observing satellite constellations. *Journal of Spacecraft and Rockets*, 58(5):1416–1429, Sept.–Oct. 2021.
- [3] Mark Stephenson and Hanspeter Schaub. Optimal agile satellite target scheduling with learned dynamics. *Journal of Spacecraft and Rockets*, 62(3):793–804, May–June 2025.
- [4] Lorenzo Quevedo Mantovani, Yumeka Nagano, and Hanspeter Schaub. Reinforcement learning for satellite

- autonomy under different cloud coverage probability observations. In *AAS Astrodynamics Specialist Conference*, Broomfield, CO, Aug. 11–15 2024. Paper No. AAS 24-208.
- [5] Cannon Whitney and Joseph Melville. Toward trusted on-board ai: Advancing small satellite operations using reinforcement learning. In *39th Annual Small Satellite Conference*, 2025.
 - [6] Andrew Harris and Hanspeter Schaub. Spacecraft command and control with safety guarantees using shielded deep reinforcement learning. In *AIAA SciTech*, Orlando, Florida, Jan. 6–10 2020.
 - [7] Islam Nazmy, Andrew Harris, Morteza Lahijanian, and Hanspeter Schaub. Shielded deep reinforcement learning for multi-sensor spacecraft imaging. In *American Control Conference*, Atlanta, Georgia, June 8–10 2022.
 - [8] Andrew Harris, Trace Valade, Thibaud Teil, and Hanspeter Schaub. Generation of spacecraft operations procedures using deep reinforcement learning. *Journal of Spacecraft and Rockets*, 59(2):611–626, March – April 2022.
 - [9] Andrew Harris and Kedar Naik. Autonomous command and control for earth-observing satellites using deep reinforcement learning. In *2023 IEEE Aerospace Conference*, pages 1–12, 2023.
 - [10] Adam Herrmann and Hanspeter Schaub. Reinforcement learning for the agile earth-observing satellite scheduling problem. *IEEE Transactions on Aerospace and Electronic Systems*, 59(5):5235–5247, Oct. 2023.
 - [11] Mark Stephenson, Lorenzo Quevedo Mantovani, and Hanspeter Schaub. Learning policies for autonomous earth-observing satellite scheduling over semi-markov decision processes. *Journal of Aerospace Information Systems*, 22(9):741–803, Sept. 2025.
 - [12] Mark Stephenson, Lorenzo Quevedo Mantovani, Anaïs Cheval, and Hanspeter Schaub. Quantifying the optimality of a distributed RL-based autonomous earth-observing constellation. In *AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, Jan. 31 – Feb. 5 2025. Paper No. AAS-25-033.
 - [13] Lorenzo Quevedo Mantovani and Hanspeter Schaub. Improving robustness of autonomous earth-observing spacecraft using training environment enhancements. *Journal of Aerospace Information Systems*, 23(3):293–304, 2026.
 - [14] Lorenzo Quevedo Mantovani and Hanspeter Schaub. Performance evaluation of shielded neural networks for autonomous agile earth observing satellites in long term scenarios. In *International Workshop on Planning & Scheduling for Space*, Toulouse, France, April 28–30 2025.
 - [15] Lorenzo Quevedo Mantovani and Hanspeter Schaub. Shielding against the unsafe: Evaluation of shielded deep reinforcement learning training approaches for autonomous spacecraft. In *AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, Jan. 31 – Feb. 4 2026.
 - [16] Mark Stephenson and Hanspeter Schaub. Optimal target sequencing in the agile earth-observing satellite scheduling problem using learned dynamics. In *AAS/AIAA Astrodynamics Specialist Conference*, Big Sky, MT, Aug. 13–17 2023. Paper AAS 23–108.
 - [17] Adam Herrmann, Mark Stephenson, and Hanspeter Schaub. Single-agent reinforcement learning for scalable earth-observing satellite constellation operations. *Journal of Spacecraft and Rockets*, 61(1):114–132, Jan. – Feb. 2024.
 - [18] Adam Herrmann and Hanspeter Schaub. Monte carlo tree search methods for the earth-observing satellite scheduling problem. *Journal of Aerospace Information Systems*, 19(1):70–82, January 2022.
 - [19] Mark Stephenson and Hanspeter Schaub. Scalable autonomous decentralized constellation tasking on asynchronous semi-markov decision processes. In *International Workshop on Satellite and Constellations Formation Flying*, Kaohsiung, Taiwan, Dec. 2–4 2024.
 - [20] Mark Stephenson, Lorenzo Quevedo Mantovani, and Hanspeter Schaub. Intent sharing for emergent collaboration in autonomous earth observing constellations. In *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 2–7 2024. Paper No. AAS 24-192.
 - [21] Yumeka Nagano and Hanspeter Schaub. Fault resilience of reinforcement-based satellite autonomous task scheduling. In *AAS Spaceflight Mechanics Meeting*, Kauai, Hawaii, Jan. 19–23 2025. Paper No. AAS 25-159.
 - [22] Yumeka Nagano and Hanspeter Schaub. Enhancing fault resilience in RL-based satellite autonomous task scheduling. In *AAS/AIAA Astrodynamics Specialist Conference*, Boston, MA, August 10–14 2025. Paper No. AAS 25-637.
 - [23] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. MIT Press, Cambridge, MA, 2 edition, 2018.
 - [24] Scott Carnahan, Scott Piggott, and Hanspeter Schaub. A new messaging system for basilisk. In *AAS Guidance and Control Conference*, Breckenridge, CO, Jan. 30 – Feb. 5 2020. AAS 20-134.
 - [25] Patrick W. Kenneally, Scott Piggott, and Hanspeter Schaub. Basilisk: A flexible, scalable and modular astrodynamics simulation framework. *Journal of Aerospace Information Systems*, 17(9):496–507, Sept. 2020.
 - [26] John Alcorn, Hanspeter Schaub, Scott Piggott, and Daniel Kubitschek. Simulating attitude actuation options using the basilisk astrodynamics software architecture. In *67th International Astronautical Congress*, Guadalajara, Mexico, Sept. 26–30 2016.
 - [27] John Alcorn, Cody Allard, and Hanspeter Schaub. Fully coupled reaction wheel static and dynamic imbalance for spacecraft jitter modeling. *AIAA Journal of Guidance, Control, and Dynamics*, 41(6):1380–1388, 2018.
 - [28] Paolo Panicucci, Cody Allard, and Hanspeter Schaub. Spacecraft dynamics employing a general multi-tank and

- multi-thruster mass depletion formulation. *Journal of Astronautical Sciences*, 65(4):423–447, 2018.
- [29] João Vaz Carneiro and Hanspeter Schaub. Spacecraft dynamics with the backsubstitution method: Survey and capabilities. In *AAS Spaceflight Mechanics Meeting*, Kauai, Hawaii, Jan. 19–23 2025. Paper No. AAS 25-232.
 - [30] João Vaz Carneiro and Hanspeter Schaub. Scalable architecture for rapid setup and execution of multi-satellite simulations. *Advances in Space Research*, 73(11):5416–5425, June 2024.
 - [31] Adam Herrmann, João Vaz Carneiro, and Hanspeter Schaub. Reinforcement learning for the multi-satellite earth-observing scheduling problem. In *AAS Guidance and Control Conference*, Breckenridge, CO, Feb. 3–9 2022.
 - [32] João Vaz Carneiro, Morteza Lahijanian, Kendra Lang, Konstantin Borozdin, and Hanspeter Schaub. Machine learning for reaction wheel fault detection using simulated telemetry data. In *AIAA SciTech*, San Diego, CA, Jan. 3–7 2022.
 - [33] João Vaz Carneiro, Cody Allard, and Hanspeter Schaub. General dynamics for single- and dual-axis rotating rigid spacecraft components. *Journal of Spacecraft and Rockets*, 61(4):1099–1113, July–August 2024.
 - [34] Leah Kiner, Hanspeter Schaub, and Cody Allard. Spacecraft backsubstitution dynamics with general multibody prescribed subcomponents. *Journal of Aerospace Information Systems*, 22(8):703–715, August 2025.
 - [35] Riccardo Calaon, Cody Allard, and Hanspeter Schaub. Momentum management of a spacecraft equipped with a dual-gimbaled electric thruster and rotating solar arrays. *The Journal of the Astronautical Sciences*, 73(47), 2026.
 - [36] Andrew Morell and Hanspeter Schaub. Expanded backsubstitution dynamics modeling for branched force and torque based spacecraft components. In *AAS Spaceflight Mechanics Meeting*, Kauai, Hawaii, Jan. 19–23 2025. Paper No. AAS 25-185.
 - [37] Andrew Morell and Hanspeter Schaub. Back-substitution based spacecraft dynamics modeling with selective configuration space branching. In *AAS Astrodynamics Specialist Conference*, Broomfield, CO, Aug. 11–15 2024. Paper No. AAS 24-361.
 - [38] Daniel Huterer Prats and Hanspeter Schaub. Autonomous space-based imaging: Reinforcement learning scheduling with imaging–downlink tradeoffs across orbital regimes. In *AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, Jan. 31 – Feb. 4 2026. Paper No. AAS 26-052.
 - [39] João Vaz Carneiro and Hanspeter Schaub. Novel architecture for numerical multi-satellite simulations. In *International Workshop on Satellite Constellations and Formation Flying*, Milano, Italy, June 7–10 2022.
 - [40] Mark Stephenson and Hanspeter Schaub. Bsk-rl: Modular, high-fidelity reinforcement learning environments for spacecraft tasking. In *International Astronautical Congress*, Milano, Italy, Oct. 14–18 2024.
 - [41] Jennifer Wood, Mar Cols Margenet, Patrick W. Kenneally, Hanspeter Schaub, and Scott Piggott. Flexible basilisk astrodynamics visualization software using the unity rendering engine. In *AAS Guidance and Control Conference*, pages 621–629, Breckenridge, CO, Feb. 1–7 2018. Paper AAS 18-093.
 - [42] Anaïs Cheval and Hanspeter Schaub. Autonomous strip imaging task scheduling in super-agile satellites using reinforcement learning. In *AAS/AIAA Astrodynamics Specialist Conference*, Boston, MA, August 10–14 2025. Paper No. AAS 25-726.
 - [43] Anaïs Cheval and Hanspeter Schaub. Reinforcement learning with hybrid action representation for autonomous strip imaging task scheduling in high-resolution super-agile satellites. In *AIAA SciTech*, Orlando, FL, Jan. 12–16 2026.
 - [44] Daniel Huterer Prats, Hanspeter Schaub, and Chris Wheeler. Reinforcement learning for space-to-space surveillance: Autonomous scheduling for resident space object imaging. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, Hawaii, September 16–19 2025.
 - [45] Mark Stephenson and Hanspeter Schaub. Safe, autonomous multi-agent inspection of space objects leveraging relative orbit dynamics. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, Hawaii, September 16–19 2025.
 - [46] Mark Stephenson and Hanspeter Schaub. Comparing probabilistic roadmaps and reinforcement learning for relative motion inspection of space objects. In *AAS Guidance, Navigation and Control Conference*, Breckenridge, CO, Jan. 31 – Feb. 4 2026.
 - [47] Ryan Reed, Hanspeter Schaub, and Morteza Lahijanian. Shielded deep reinforcement learning for complex spacecraft specifications. In *2024 American Control Conference (ACC)*, pages 4972–4979, Toronto, ON, Canada, July 2024. IEEE.
 - [48] Yumeka Nagano and Hanspeter Schaub. Autonomous task scheduling for earth-observing satellites tracking moving targets with low maneuverability. In *AIAA SciTech*, Orlando, FL, Jan. 12–16 2026.
 - [49] Mar Cols Margenet and Hanspeter Schaub. Sequentially distributed attitude guidance software across a spacecraft constellation. In *International Workshop on Satellite Constellations and Formation Flying*, University of Strathclyde, Glasgow, Scotland, July 17–19 2019. Paper No. IWSCFF 19-52.
 - [50] Mar Cols Margenet, Hanspeter Schaub, and Scott Piggott. Modular platform for hardware-in-the-loop testing of autonomous flight algorithms. In *International Symposium on Space Flight Dynamics*, Matsuyama-Ehime, Japan, June 3–9 2017.
 - [51] Mar Cols Margenet, Hanspeter Schaub, and Scott Piggott. Flight software development, migration, and testing in desktop and embedded environments. *Journal of Aerospace Information Systems*, 18(4):157–174, 2021.

- [52] Mar Cols Margenet, Hanspeter Schaub, and Scott Piggott. End-to-end flight algorithm development and testing: Transitioning from desktop prototyping to embedded testing. In *12th Annual Workshop on Spacecraft Flight Software*, Huntsville, AL, Dec. 9–12 2019.
- [53] Mar Cols Margenet, Hanspeter Schaub, and Scott Piggott. Avionics hardware modeling and embedded flight software testing in an emulated flat-sat. In *AAS Guidance and Control Conference*, Breckenridge, CO, Jan. 30 – Feb. 5 2020. AAS 20-103.