(Preprint) AAS 23-146

REINFORCEMENT LEARNING FOR MULTI-SATELLITE AGILE EARTH OBSERVING SCHEDULING UNDER VARIOUS COMMUNICATION ASSUMPTIONS

Adam Herrmann[*], Mark Stephenson[†], and Hanspeter Schaub[‡]

This work explores the use of reinforcement learning (RL) for the multi-satellite agile Earth-observing (MSAEO) scheduling problem. In this work, a policy is trained in a single satellite environment on a fixed number of imaging targets where the status is updated as they are imaged and downlinked by the spacecraft. The policy is then deployed in a multi-satellite scenario where each spacecraft has its own list of imaging targets. Each spacecraft in the multi-satellite scenario has its own copy of the policy and makes decisions using a local observation of the state. The spacecraft communicate with one another to update their lists of targets. While the method applied to this problem generates sub-optimal policies in terms of global reward, the distributed nature of the architecture simplifies the training process and required training time. Furthermore, this method readily scales with changing numbers of satellites as no assumptions are made regarding the constellation design in training. This autonomous scheduling approach is evaluated and benchmarked for four cross-link communication assumptions, namely free communication, two line-of-sight communication methods, and no communication. A range of Walker-Delta constellations are explored to determine how the performance of the trained agents relates to both the communication method and constellation design. Experimental results demonstrate that the free communication assumption produces the best performance (i.e. fewer duplicate targets), and the no communication assumption produces the worst performance (more duplicate targets). The performance of the line-of-sight communication assumption depends heavily on the design of the constellation and how frequently the spacecraft can communicate with one another.

INTRODUCTION

In the multi-satellite agile Earth-observing (MSAEO) scheduling problem, a constellation of spacecraft attempt to maximize the weighted sum of imaging targets collected and downlinked while avoiding resource constraint violations. Each spacecraft maintains its own list of imaging targets, which may be shared between spacecraft. These lists of targets may be modified in multiple ways, either by the ground or another Earth-observing satellite outside of the constellation. An example of this problem is depicted in Figure 1, where a constellation of SmallSats are tasked with lists of imaging targets that are modified by both the ground station and a polar-orbiting satellite.

[*]PhD Candidate, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, Boulder, CO, 80309. AIAA Member.

[†]PhD Student, Ann and H.J. Smead Department of Aerospace Engineering Sciences, University of Colorado, Boulder, Boulder, CO, 80309. AIAA Member.

[‡]Professor and Department Chair, Schaden Leadership 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.

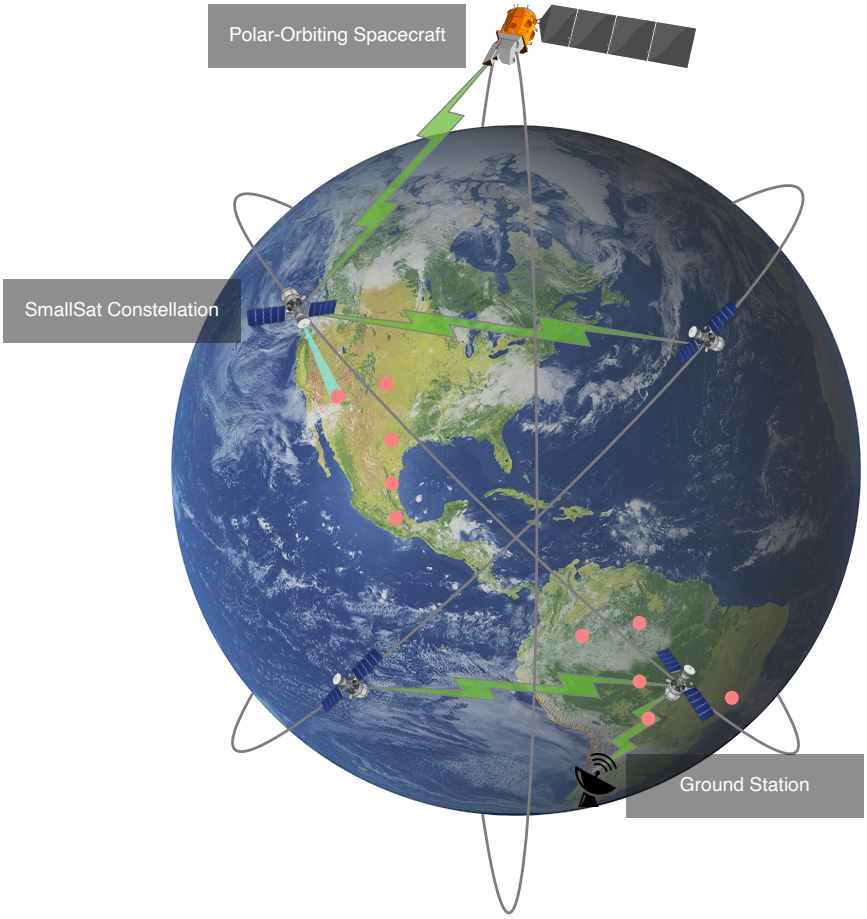


Figure 1: **Multi-Satellite Agile Earth-Observing Scheduling Problem.**

The traditional approach to EOS planning and scheduling formulates and solves an optimization problem over some planning horizon. The solution to this problem is sequenced into commands, which are uplinked to the spacecraft and executed open-loop. Many mixed-integer programming formulations of the multi-satellite EO scheduling problem are prevalent in the literature and address challenges specific to EO scheduling.[1–4] Both Planet[5] and Spire Global[6] take MIP-based optimization approaches for their constellations. One of the primary challenges in Earth-observing satellite scheduling that is difficult for optimization-based formulations and solutions to address is replanning in the event of opportunistic science events, missed ground contacts, mismodeling, or task execution taking longer or shorter than anticipated. Several authors have posed solutions to this problem. Chien et al. use iterative repair in real on-board planning systems to improve science return.[7–10] Valicka et al. formulate multi-stage stochastic MIP models for a constellation of satellites that addresses cloud coverage uncertainty, which could necessitate re-planning in deterministic planning systems.[11] The Scheduling Planning Routing Inter-satellite Network Tool (SPRINT) addresses these issues by utilizing a global planner for constellation-level management and local planners to handle unexpected opportunities and events.[12,13] In addition to optimization-based solutions, reinforcement learning has become popular for the multi-satellite agile EO scheduling problem. Reinforcement learning-based approaches are desirable because the planning is inherently closed-loop, so the decision-making agent is always acting based on the current state of the environment. Wei

et al. formulate a scheduling and timing problem for a constellation of Earth-observing satellites and apply actor-critic reinforcement learning, showing that it can outperform a genetic algorithm in terms of optimality and execution time.[14] The authors do not consider power or data constraints. Cui et al. apply double DQN for communication scheduling of a constellation of Earth-orbiting satellites.[15] The authors do consider power as a resource and demonstrate that their algorithm is superior to a genetic algorithm in terms of performance and computation time. Dalin et al. formulate a scheduling problem for multi-satellite tasking with both data and power constraints (although it's important to note that these are not replenishable resources) and apply the multi-agent deep deterministic policy gradient (MADDPG) algorithm to solve the problem.[16] The performance of the MADDPG algorithm is shown to be comparable to other solvers for this problem.

Reinforcement learning is an appealing approach for the MSAEO scheduling problem due to the ability of policy-based decision-making agents to rapidly plan in a closed-loop fashion, responding to the rapid changes in target requests. However, multi-agent reinforcement learning comes with many challenges, especially when decentralized deployment is desired. Multi-agent environments with limited to no communication between agents appear non-stationary to individual agents because other agents change the environment with their actions. The problem may be cast as a decentralized partially observable Markov decision process (Dec-POMDP) to account for the uncertainty in the environment due to other agents. Several multi-agent robotics problems using macro-actions have demonstrated the success of such an approach.[17–19] However, finding an optimal solution for a finite-horizon Dec-POMDP is NEXP-complete.[20] Determining the observations required for coordination amongst agents is also non-trivial. If free communication between agents and full observability of the environment is assumed, the Dec-POMDP can be reduced to a multi-agent Markov decision process (MMDP).[21,22] Finding an optimal solution for a finite horizon MDP is only P-complete,[23] but the joint action space is exponential in the number of agents. Furthermore, the assumption that each agent has full observability over all other agents is a dubious one. To avoid the computational requirements of solving a Dec-POMDP or MMDP, this work explores a problem formulation that treats the environment like it's an MDP for each individual spacecraft. The other spacecraft may change the environment, but this happens in a predictable manner if some assumptions are made about the behavior of the trained agents. The agents trained following the MDP assumption are deployed in a multi-satellite constellation, where each spacecraft uses its own local observations and policy to make decisions. Performance is benchmarked for various communication assumptions and various Walker-delta constellation designs to determine how the communication assumptions affect duplication of effort, global reward, and local reward. A past iteration of this work was presented at the 2022 Rocky Mountain Guidance, Navigation, and Controls Conference.[24] This work expands upon past work by investigating various communication assumptions. Furthermore, a slightly modified environment and updated agents are used for this work.

This paper first describes the single and multi-satellite agile EOS scheduling problems. The Markov decision process formulations, communication assumptions, and simulation architecture for each problem formulation are then presented. Four communication assumptions are presented: free communication, no communication, single-degree line-of-sight communication, and multi-degree line-of-sight communication. The Monte Carlo tree search and supervised learning-based training method is discussed, as well as the deployment of the trained decision-making agents in the multi-satellite environment. Finally, the results that explore how the various communication assumptions impact performance are presented. The paper then concludes with a summary of the findings and a

discussion of future work.

PROBLEM FORMULATION

Single Satellite Agile Earth-Observing Satellite Scheduling Problem

Overview In the single satellite agile Earth-observing satellite (SSAEO) scheduling problem, a spacecraft in low-Earth orbit attempts to maximize the weighted sum of targets collected and downlinked while avoiding data buffer, reaction wheel speed, and battery charge resource violations. Over the course of its three orbit planning horizon, the spacecraft has a set of 135 targets along its flight-path available, each with its own priority (1-3). This set of targets is referred to as $\mathbf{T}$. The set is ordered by spacecraft access time. The three orbit planning horizon is split into 45 decision-making intervals, each of which last for a total of six minutes.

Markov Decision Process Formulation The Markov decision process (MDP) formulation for the SSAEO scheduling problem is described in detail in Reference 25. A Markov decision process is a sequential decision making problem in which a decision-making agent selects an action, a_i, in some state, s_i, based on a policy, $\pi : \mathcal{S} \times \mathcal{A}$. The agent transitions to a new state, s_{i+1}, and receives a reward, r_i, based on the reward function of the MDP, $R : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{R}$. MDPs follow the Markov assumption, which states that the next state is conditionally dependent only on the current state and action:

$$T(s_{i+1}|s_i, a_i) = T(s_{i+1}|s_i, a_i, s_{i-1}, a_{i-1}, ..., s_0, a_0) \tag{1}$$

The state space, $\mathcal{S}$, must be constructed to maintain the Markov assumption. In the SSAEO scheduling problem, the state space is given as follows to adhere to this assumption as closely as possible:

- ECEF spacecraft position, ${}^{\mathcal{E}}\mathbf{r}$
- ECEF spacecraft velocity, ${}^{\mathcal{E}}\mathbf{v}$
- Image tuples for targets $c_j \in \mathbf{U}$
  - Target position in the spacecraft Hill frame, ${}^{\mathcal{H}}\mathbf{r}_j$
  - Priority, p_j
- L^2 norm of Modified Rodrigues Parameter (MRP) attitude error, $||\boldsymbol{\sigma}_{\mathcal{B}/\mathcal{R}}||$
- L^2 norm of angular attitude rate vector, $||{}^{\mathcal{B}}\boldsymbol{\omega}_{\mathcal{B}/\mathcal{N}}||$
- Reaction wheel speeds, $\boldsymbol{\Omega}$
- Battery charge, z
- Eclipse indicator, k
- Stored data in buffer, b
- Data transmitted, h
- Ground station access indicators

Information on the spacecraft geometry, attitude states and rates, imaging target states and priorities, and spacecraft resources are included in the state space. Eclipse indicators and ground station access indicators are also included in the state space.

The action space, $\mathcal{A}$ is given as follows:

- Charge
- Desaturate
- Downlink
- Image target $c_1 \in \mathbf{U}$

  $\vdots$

- Image target $c_j \in \mathbf{U}$

In the charging mode, the spacecraft points its solar panels at the sun to charge its batteries. In the desaturation mode, the spacecraft points its solar panels at the sun to maintain power and simultaneously maps reaction wheel momentum to thrust commands to remove momentum from the wheels. In the downlink mode, the spacecraft points its antenna in the nadir direction and downlinks data when a ground station is in view.

Finally, the last few modes deal with imaging. Because the spacecraft cannot image all targets in $\mathbf{T}$ at any given timestep, only the next few upcoming targets are included in the action space for imaging. The subset of upcoming targets, $\mathbf{U}$, is defined in Equation 2, where J is the number of targets in the state and action space. $\mathbf{D}$ is a subset of $\mathbf{T}$ that contains the targets that have been imaged by the spacecraft or passed by already.

$$\mathbf{U} = \{c_j \in (\mathbf{T} - \mathbf{D}) \mid \forall\, j \in [1, J]\} \tag{2}$$

Finally, a reward function is created that accounts for a.) the desire to avoid resource constraint violations and b.) the desire to image and downlink targets. The reward function is given in Equation 3.

$$R(s_i, a_i, s_{i+1}) = \begin{cases} -1 & \text{if failure} \\ \frac{1}{45}\sum_j^{|\mathbf{T}|} H(d_j) & \text{if } \neg\text{failure} \wedge a_i \text{ is downlink} \\ \frac{0.1}{45} H(w_j) & \text{if } \neg\text{failure} \wedge a_i \text{ is image } c_j \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The first condition checked for is the failure condition. If the spacecraft exceeds the maximum reaction wheel speeds, expends all charge in the battery, or overflows the data buffer then the failure condition is true and the agent receives -1 reward.

$$\text{failure} = (z = 0 \ \vee \ \text{any}(\hat{\mathbf{\Omega}} \geq 1) \ \vee \ b \geq 1) \tag{4}$$

The second condition checked for handles the downlink of targets. If the downlink mode occurs, the $H(d_j)$ function is computed for all targets using a downlink state, d_j, which represents whether or not target j has been downlinked. The total reward is summed and divided by 45, the total number of planning intervals, to ensure the upper limit for this component of the reward is equal to 1.

$$H(x_j) = (1/p_j) \text{ if } \neg x_{j_i} \ \wedge \ x_{j_{i+1}} \tag{5}$$

If the image target c_j mode is initiated, the $H(w_j)$ operator for target c_j is computed, returned, and scaled by $0.1/45$. The variable w_j represents if the target has been imaged or not. The addition of this small positive reward helps to make reward less sparse and ensures that the decision-making agent still has incentive to image after all downlink windows have been passed.

Multi-Satellite Agile Earth-Observing Satellite Scheduling Problem

Overview In the multi-satellite agile Earth-observing satellite (MSAEO) scheduling problem, more than one spacecraft attempt to maximize the number of images collected and downlinked while avoiding resource constraint violations. In this work, the decision-making agents on-board each spacecraft attempt to maximize local reward and do not coordinate with other spacecraft to maximize global reward. The spacecraft have access to a global set of targets, $\mathbf{M}$, and each spacecraft k has its own set of targets, $\mathbf{T}_k$. Spacecraft may share targets within $\mathbf{M}$. The satellite constellations are designed using the Walker-Delta notation. The N satellites are distributed evenly between P orbit planes. The orbital planes are distributed at $360/P$ deg intervals of the longitude of ascending node. Relative phasing may be prescribed in Walker-delta constellations, but is not in this work.

The Markov decision process formulation of the problem is largely unchanged, at least for individual decision-making agents. The complete state space is now given by $\mathcal{S} : \{s_i^0, \cdots, s_i^k\}$, but each decision-making agent maintains an observation over its own state, s_i^k. The state changes with the joint action space, $\mathcal{A} : \mathcal{A}^1 \times \cdots \times \mathcal{A}^k$. The transition function is therefore a function of the state space and joint actions, $s_{i+1}, r_i^0, \cdots, r_i^k \sim G(s_i, \mathbf{a}_i)$. Likewise, the joint reward function is a function of the state space and joint actions, $\mathbf{R}(s, \mathbf{a}) = (R^0(s, \mathbf{a}), \cdots, R^k(s, \mathbf{a}))$. For the individual decision-making agents, the reward function in Equation 3 is the same, but the $H(x_j)$ function now sweeps through the global target set $\mathbf{M}$ to determine if a target was imaged or downlinked already. If another spacecraft already imaged or downlinked the target, no new reward is returned.

The spacecraft may have the ability to communicate with one another to update whether or not targets in $\mathbf{T}_k$ have already been imaged or downlinked. The spacecraft do not update their target lists based on contact with ground stations. Only inter-satellite communication is considered. Several communication assumptions are explored in this work, including no communication, free communication, and line-of-sight communication. During the communication step at the end of each decision interval, the spacecraft that have communicated with one another during the previous interval of simulation loop through the lists of targets available from other spacecraft and mark which targets have already been imaged or downlinked, which in turn prevents them from being added to $\mathbf{U}^k$ and selected for imaging in the future. The four cases are as follows:

Free Communication Free communication models a scenario where additional communication infrastructure supplements the imaging satellites, such as a communications constellation. In the free communication case, every satellite is able to share imaged target lists with every other satellite at the communication step of every decision interval; that is, if $c_i \in \mathbf{T}_j$ is marked as imaged by

spacecraft j, c_i is also marked as imaged by any other spacecraft k for which $c_i \in \mathbf{T}_k$. This behavior is illustrated in Figure 2d. Note that even with communication, satellites do not actively coordinate their next actions, so two satellites can still image the same target if they make the decision to do so at the same interval.

Single Degree Line-of-Sight Communication Single degree line-of-sight models a constellation with limited inter-satellite communication bandwidth. Line-of-sight connectivity is defined as a straight-line connection between two satellites unoccluded by Earth plus a 100km layer of atmosphere. Each satellite updates its list against that of direct line-of-sight neighbors. Information is only able to travel one degree through the network of satellites in a single communication step: If satellites $i \leftrightarrow j \leftrightarrow k$ have line-of-sign connections but not between $i \not\leftrightarrow k$, k will receive updates from j's list but not i's list.

Multi-Degree Line-of-Sight Communication As opposed to the single degree model, multi-degree line-of-sight assumes satellites have high bandwidths and fast communication speeds. If at the communication step there is any connection in the satellite network between two satellites, they will share imaged satellite information. For example, if satellites $i \leftrightarrow j \leftrightarrow k$ but not between $i \not\leftrightarrow k$, k will receive updates from i's list via j.

No Communication The no communication case models the satellites as independent agents without inter-satellite communication capabilities. Satellites never share information about imaged targets with each other; if $c_i \in \mathbf{T}_j$ is marked as imaged by satellite j, it does not affect any other satellite k's $\mathbf{T}_k$.

Simulation Architecture

Both the single and multi-satellite agile EOS scheduling problems are simulated using the Basilisk* astrodynamics software architecture, a high-fidelity simulation framework for astrodynamics problems.[26] Each of the simulations are wrapped within a Gym environment, which is a standard interface for reinforcement learning problems that allows decision-making agents to pass action to the simulation and receive observations and rewards in return, which is depicted in Figure 3.

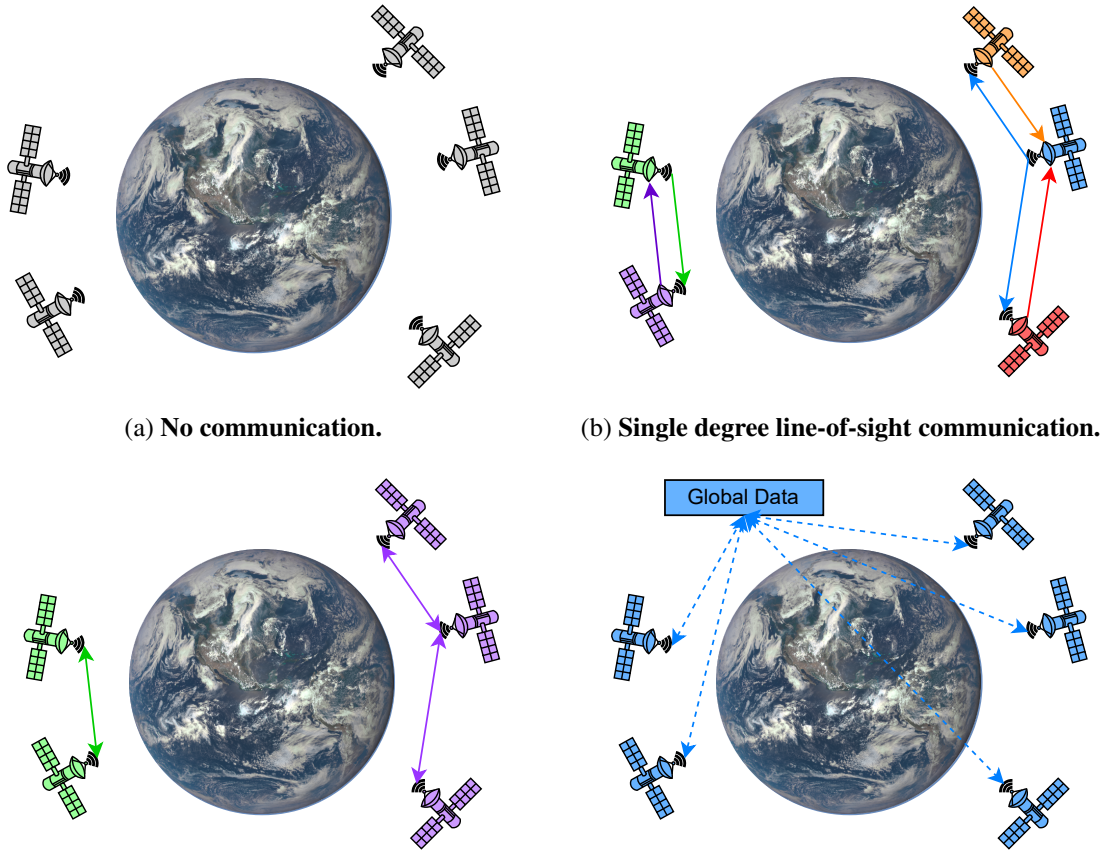
The Basilisk simulation for both the single and multi-satellite agile EOS scheduling problem contains an attitude control system with reaction wheels and thrusters, a power system with batteries, solar panels, and power sinks, and an on-board data storage system that includes a transmitter, instruments, and a data buffer. Ground stations on the surface of the Earth are also simulated to ensure the transmitter only downlinks when a ground station is in view. The simulation architectures are described in detail in References 24 and 27. Furthermore, the source code for each of the simulation architectures may be found on the develop branch of the basilisk-gym-interface library† under the names `multiTgtEarthEnvironment` and `multiSatMultiTgtEarthEnvironment`.

METHODS

MCTS-Train

The agents are trained using the MCTS-Train architecture, a training pipeline inspired by AlphaZero that is described in detail in References 25 and 27. A diagram of the MCTS-Train pipeline is

*`https://hanspeterschaub.info/basilisk`
†`https://bitbucket.org/avslab/basilisk-gym-interface`



(a) **No communication.**

(b) **Single degree line-of-sight communication.**

(c) **Multi degree line-of-sight communication.**

(d) **Free communication.**

Figure 2: **Communication Methods.**

provided in Figure 4. In summary, MCTS-Train utilizes Monte Carlo tree search, an online search-based algorithm commonly used for RL problems, to generate solutions over the planning horizon as well as an estimate of the state-action value function, $\hat{Q}(s, a)$, in the form of thousands of data points. The state-action value estimates are regressed over using various neural networks, each with a unique combination of hyperparameters, to generate a neural network approximation of the state-action value function, $Q_\theta(s, a)$. The trained state-action value functions are then deployed in the environment for validation using the following policy:

$$\pi(s) = \arg\max_a Q_\theta(s, a) \tag{6}$$

At the core of the MCTS-Train pipeline is Monte Carlo tree search (MCTS). At each step through the environment, MCTS runs a number of simulations in the environment to determine the next best action to take. During the simulation step, MCTS selects the action that maximizes the current estimate of the state-action value function (represented in tabular form at this point, based on the simulated states) and the exploration term. If MCTS reaches a state it has never visited before, it initializes $\hat{Q}$ and executes a rollout policy, a heuristic policy that avoids resource constraint violations and downlinks or images in the nominal states.

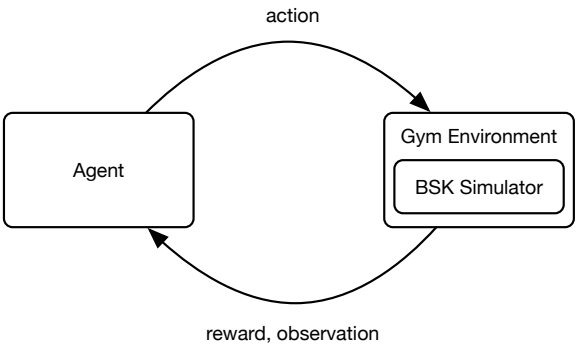


Figure 3: **Gym Environment Interface**

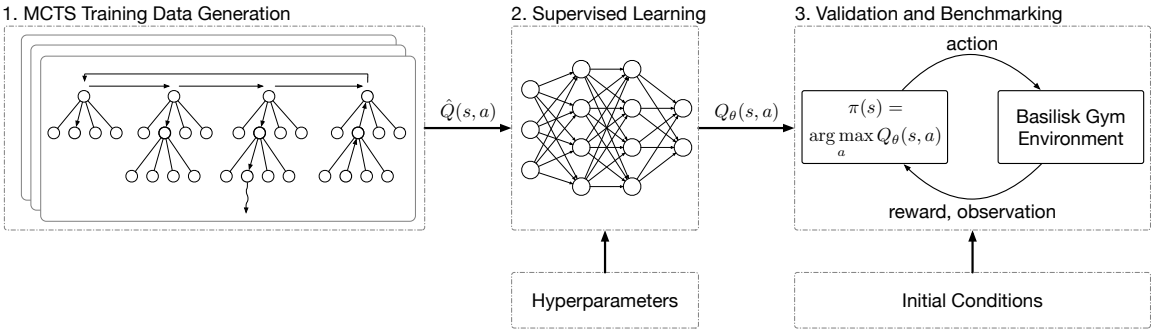


Figure 4: **MCTS-Train Pipeline.**

To construct the rollout policy, a low dimensional safety MDP is first constructed. The state space of this safety MDP is $S_{\text{safe}} : \{\text{tumbling},\ \text{saturated},\ \text{low power},\ \text{buffer overflow}\}$. The values take a value of 0 or 1 based on whether or not the state exceeds a safety limit. The safety limits are tuned such that the resource constraint violations in the reward function do not occur if the safe action is taken. Safe actions include desaturation, charging, or downlink. A safety policy, referred to as the "shield," is constructed based on the value of the safety MDP. This policy is provided in Table 1 and was originally developed in Reference 27.

If the spacecraft is in a nominal state, meaning an imaging action is returned by the safety shield, then either imaging or downlink can be performed, depending on the availability of the ground stations. If a ground station is available, then the downlink mode is initiated. If there is no ground station available, then the nearest target is selected for imaging.

Deployment

After training in the single satellite EOS environment, the trained agents are deployed in the multi-satellite scenario. Each spacecraft has a copy of the trained neural network and takes actions according to the policy in Equation 6, using its observation of its own state s_i^k. The safety MDP constructed for use within MCTS is also used in deployment in the multi-satellite scenario. The output of the network is compared to the policy constructed using the safety MDP. If the network requests that an unsafe action is taken, like attempting to image when a data buffer overflow is imminent, the shield action is taken instead. If the safety MDP is in a nominal state (i.e. an imaging mode is returned), any action requested by the trained agent is permissible. A diagram of the

Table 1: Shield Policy[27]

Tumbling	Saturated	Low Power	Buffer Limit	Action
1	1	1	1	Charge
1	1	1	0	Charge
1	1	0	1	Desaturate
1	1	0	0	Desaturate
1	0	1	1	Charge
1	0	1	0	Charge
1	0	0	1	Downlink
1	0	0	0	Image
0	1	1	1	Desaturate
0	1	1	0	Desaturate
0	1	0	1	Desaturate
0	1	0	0	Desaturate
0	0	1	1	Charge
0	0	1	0	Charge
0	0	0	1	Downlink
0	0	0	0	Image

shielded agent is provided in Figure 5.

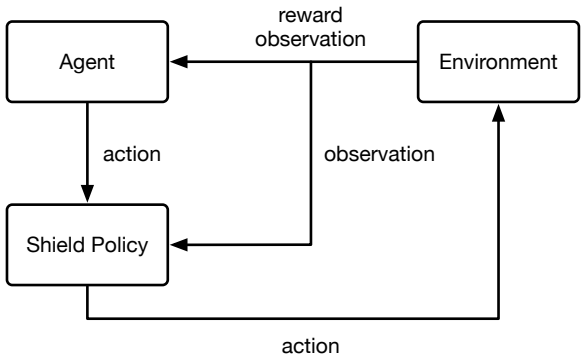


Figure 5: **Shielded Agent Deployment.**

RESULTS

To evaluate the performance of the policy in multi-agent constellations across different communication assumptions, two classes of cases are studied: single plane constellations, in which the density of satellites in a single plane is varied to evaluate the impact of intra-plane communication; and multi-plane constellations, in which the number of planes is varied to evaluate the impact of inter-plane communication. For each case, the performance is evaluated over a range of target densities.

Single Plane

For each single plane case, a Walker-delta constellation is constructed using a 500km circular orbit at a 45° inclination with N satellites equally distributed along the orbit.

Analytical predictions can be made about communication behavior in single plane constellations. There exists some number N^* below which no satellites have line-of-sight connections with their

neighbors, and above which all satellites have line-of-sight with their neighbors. Considering occlusion by the Earth plus 100km of atmosphere,

$$N^* = \frac{\pi}{\cos^{-1}\left(\frac{R_E+100\text{km}}{R_E+500\text{km}}\right)} = 9.2 \text{ satellites} \tag{7}$$

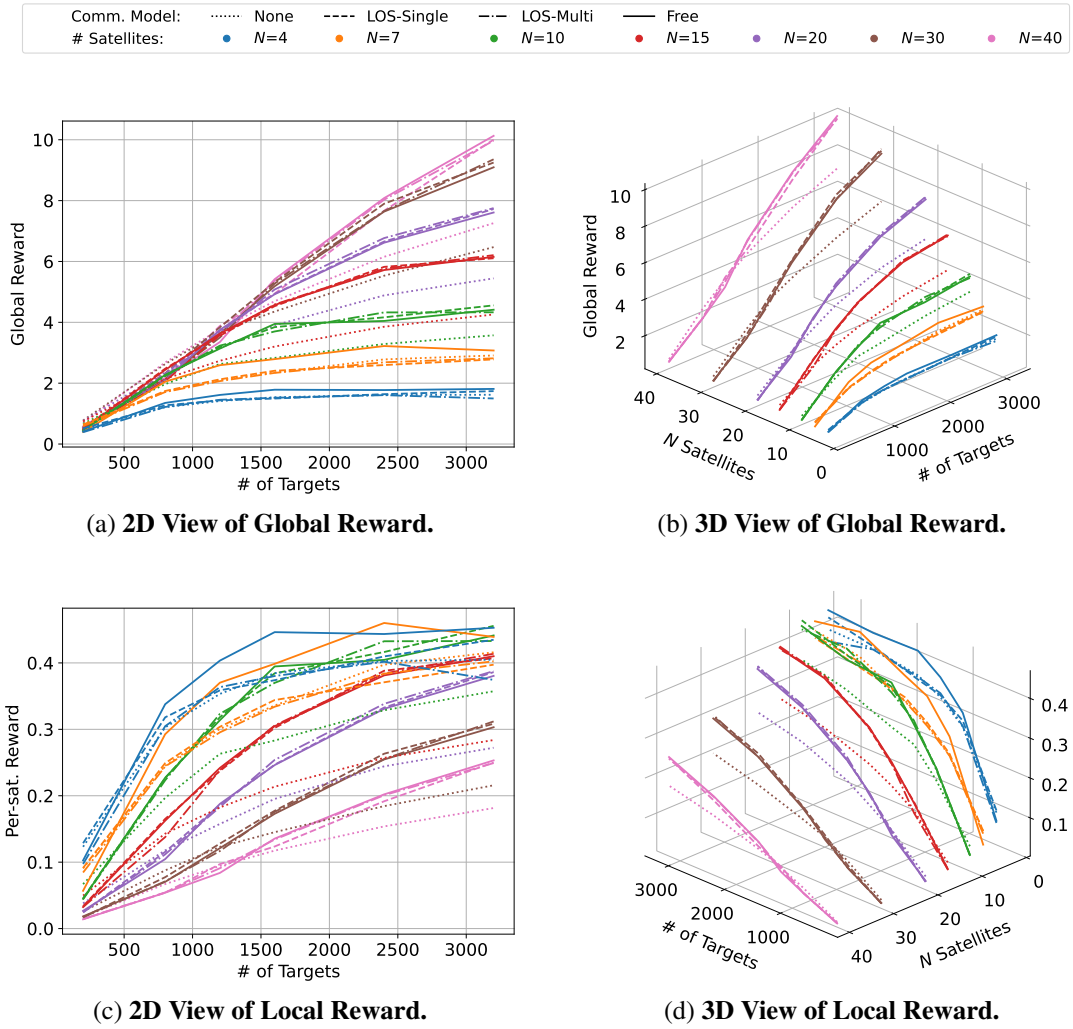
Thus, for $N < N^*$ the single- and multi-degree line-of-sight communications cases is functionally the same as the no communication case. For $N > N^*$ the multi-degree line-of-sight communication case acts the same as the free communication case; single-degree line-of-sight does not collapse to the free communication case because information propagation is not instantaneous throughout the network.

To test this hypothesis, 16 simulations are run for each combination of N satellites $\in \{4, 7, 10, 15, 20, 30, 40\}$ and global targets $\mathbf{M} \in \{200, 800, 1200, 1600, 3200\}$. The global targets are randomly generated and distributed amongst the spacecraft based on the same access model used in training. Experimental results are provided in Figures 6, 7, and 8.

Reward Figures 9a and 9b shows both 2D and 3D views of the global reward of all satellites. The global reward is defined as the sum of the local reward of each spacecraft. The global reward is limited by two factors: target density and maximum per-satellite reward. At low target densities and a high number of satellites, the global reward saturates because there are not enough targets for all satellites to maximize their individual performance. This is reflected in the per-satellite reward curves in Figures 9c and 9d where satellites in the larger constellations underperform with respect to their small-constellation counterparts. The reduced reward is due to competition for a limited number of targets: either satellites have no available targets due to other satellites having imaged them, or they are imaging the same target as another satellite in which case only the first satellite to downlink the image is rewarded. This duplication behavior is further examined in the next section. The second limiting factor is a maximum single-satellite reward achieved by the policy. When targets are plentiful compared to the number of satellites, this maximum of ~ 0.45/satellite is approached, limiting the global reward to $0.45N$. Note that the target density utilized in training is 135 total targets for each satellite, all of which are long the orbital path of the spacecraft. This corresponds to around 1000 global targets randomly distributed on the surface of the Earth.

The behavior of different communication models is visible in the reward plots. As predicted, line-of-sight cases with fewer than N^* satellites per plane ($N = 4, 7$ cases) perform the same as the no communication case, which is poor relative to the free communication case because the no communication model will reimage already imaged targets even when other non-imaged targets are available. Once N^* is exceeded ($N \geq 10$ cases) multi-degree line-of-sight performs the same as the free communication model, as predicted. Notably, the single degree line-of-sight model also matches the performance of free communication. While this was not guaranteed to be the case, the result is not surprising: Satellites receive updates to their target lists immediately from adjacent satellites, which are most likely to be considering similar targets for imaging. While far satellites receive information with a delay due to propagation through the network, the latency is shorter than the time it takes for them to orbit to the region where the information is relevant.

The relationship between unique targets imaged (Figures 10a and 10b), unique targets downlinked (Figures 10c and 10d), and global reward is also considered. Unique imaged and downlinked are roughly proportional by a factor of ~ 0.75. This behavior exists because the last downlink window for a satellite can occur well before the end of the simulation. Any images taken after that point are not downlinked within the simulation duration and thus receive relatively little reward (Eq. 3).



(a) **2D View of Global Reward.**

(b) **3D View of Global Reward.**

(c) **2D View of Local Reward.**

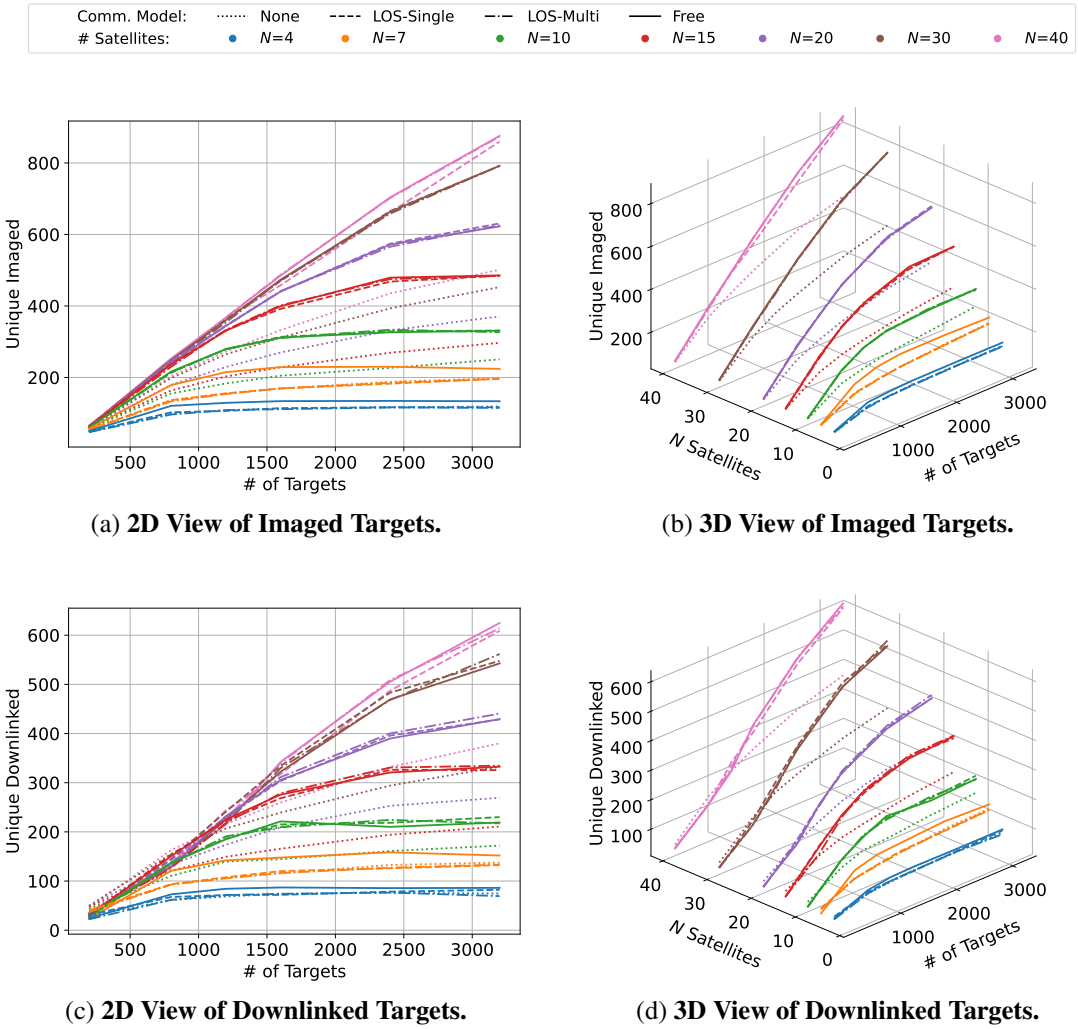
(d) **3D View of Local Reward.**

Figure 6: **Global and Local Reward for the Single Plane Experiment.**

However, since the priority of targets is distributed evenly, the resulting global reward is likewise proportional to the unique images and downlinks.

Target Duplication Target duplication statistics provide more insight into the behavior of each communication method. Figure 8 gives the number of unique targets imaged over the total number of images taken. A value of one implies that no target was imaged more than once, while values approaching zero imply a high degree of image duplication. Two broad trends exist across all constellations. The uniqueness of images increases as the density of targets increases and uniqueness decreases as the number of satellites decreases. These trends are intuitive from probability. Ignoring any intelligent behavior, fewer agents randomly selecting from more targets are less likely to select the same target. The no communication cases exhibit these trends particularly well, as there is no relationship between the targets imaged by two different satellites. Note that the decision-making agents are not trained to coordinate and avoid duplicating efforts.

As seen in the reward curves, line-of-sight communication models exactly or nearly collapse to



(a) **2D View of Imaged Targets.**

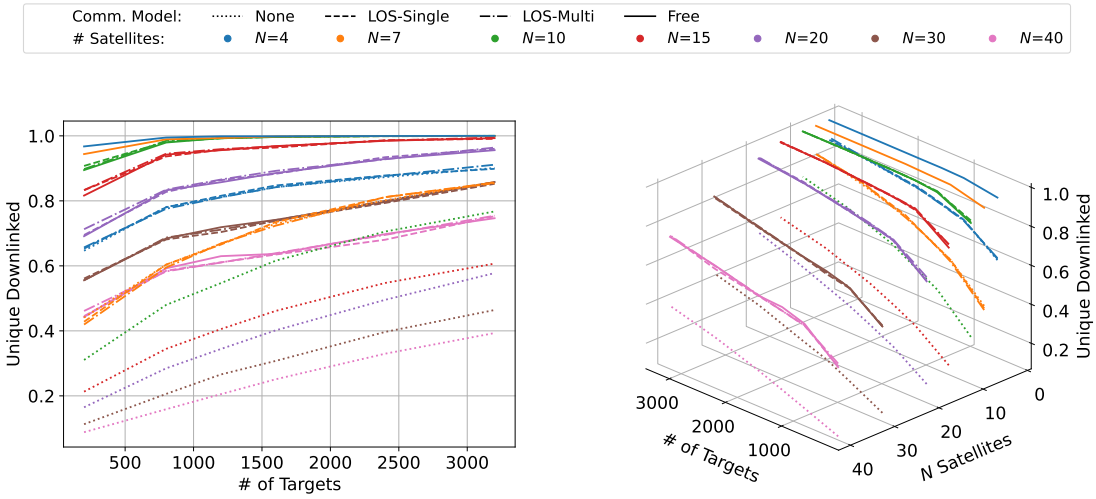
(b) **3D View of Imaged Targets.**

(c) **2D View of Downlinked Targets.**

(d) **3D View of Downlinked Targets.**

Figure 7: **Global Numbers of Unique Imaged and Downlinked Targets for the Singe Plane Experiment.**

the no communication ($N < N^*$) or free communication ($N > N^*$) models. The no communication cases display considerable image duplication because they have no mechanism to avoid it. However, the (pseudo-)free communication cases do not guarantee no target duplication. As previously mentioned, there is no protection from two satellites selecting the same target at the same decision interval, assuming they both have the same target in observation. With denser constellations and sparser targets, this competitive behavior occurs more often, resulting in free communication cases with poor duplication percentages. The best-case limiting behavior occurs in sparse constellations ($N \leq 10$) with free communication and a sufficient target density ($\geq$800). In these cases, all images taken are unique. Because six minute decision-making intervals are used, each spacecraft covers roughly 24 degrees along its orbit in a single decision-making interval. As the spacecraft increase in numbers, the amount of overlap in target coverage in a single decision-making interval increases.



(a) **2D View.** (b) **3D View.**

Figure 8: **Percent of Imaged Targets that are Unique for the Singe Plane Experiment.**

Multiple Planes

For each multiple plane case, a Walker Delta constellation is constructed with P equally-phased planes, each with 4 satellites in a 500km circular orbit at a $45°$ inclination. Note that the number of satellites per plane is below N^*, so any communication is due to inter-plane line-of-sight connections. P is odd for this experiment to avoid constellations with pairs of retrograde and prograde planes, which obfuscates the interpretation of the experimental results.

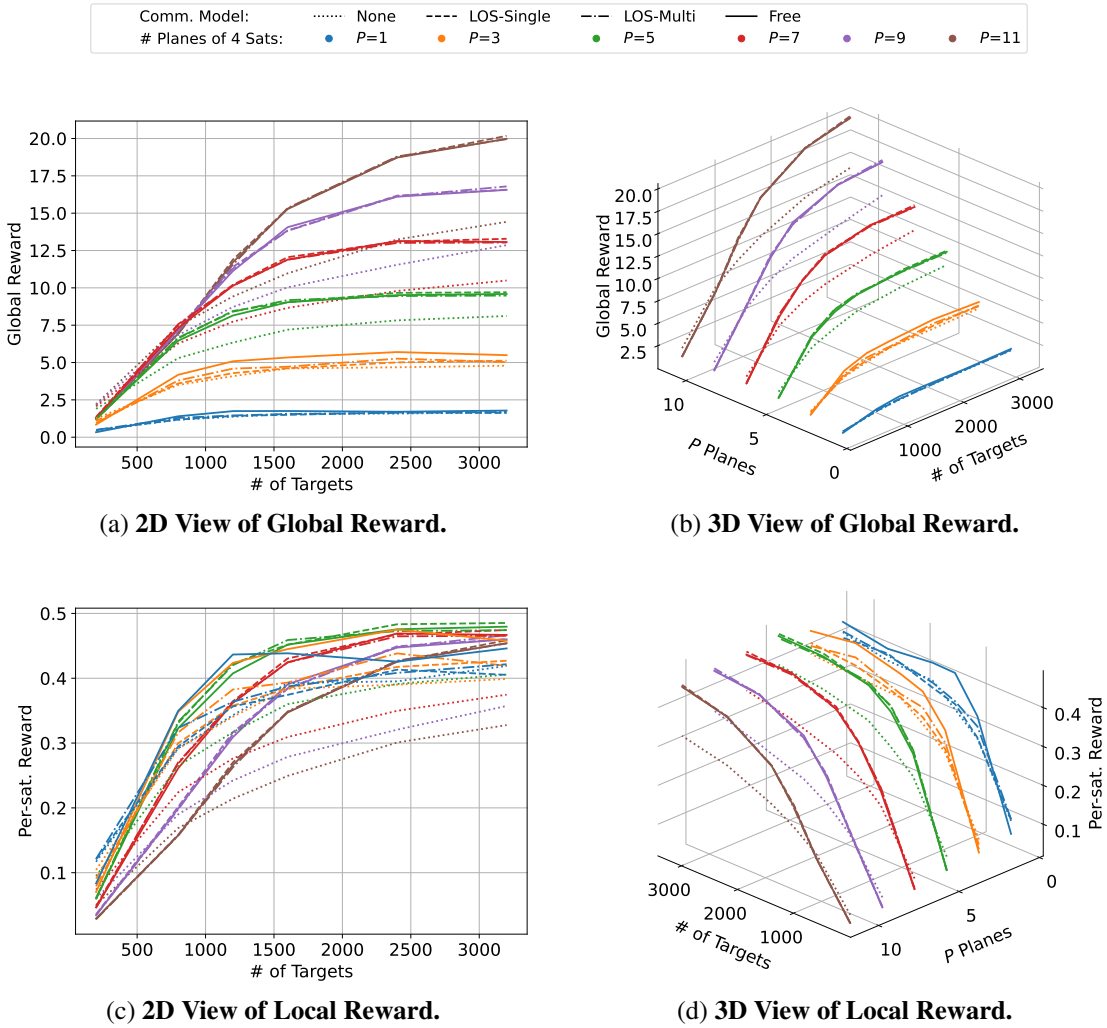
As with the single-plane cases, larger constellations are expected to produce more line-of-sight connections, causing the line-of-sight communication model to perform closer to the best-case free communication model. With a small number of planes, line-of-sight connections will only occur at high latitudes where the planes intersect. There exists some number of planes P^* where two satellites in adjacent planes can communicate with each other at the equator, where the planes are farthest apart:

$$P^* = \frac{\pi}{2\cos^{-1}\left(\frac{R_E+100\text{km}}{R_E+500\text{km}}\right)} = \frac{1}{2}N^* = 4.6 \text{ planes} \tag{8}$$

Unlike the single plane experiments, P^* does not provide a guarantee of particular behavior but does imply that constellations with $P > P^*$ will display very good line-of-sight communication.

16 simulations are run for each combination of P planes of satellites $\in \{1, 3, 5, 7, 9, 11\}$ and global targets $\mathbf{M} \in \{200, 800, 1200, 1600, 3200\}$; results are given in Figures 9, 10, and 11.

Reward Similar to the single plane experiments, the multi-plane global reward plots (Figure 9a and 9b) show that the target density and maximum per-satellite reward limit the global reward. However, target density is less limiting than in the single plane cases; more orbital planes means that a greater percent of the global targets fall along a ground track, decreasing the amount of competition for the same targets. Thus for the same number of satellites, the multi-plane cases outperform the single plane cases until unless the maximum per-satellite reward limit has been met and they perform equally. This behavior is reflected in the per-satellite reward plots (Figure 9c



(a) **2D View of Global Reward.**

(b) **3D View of Global Reward.**

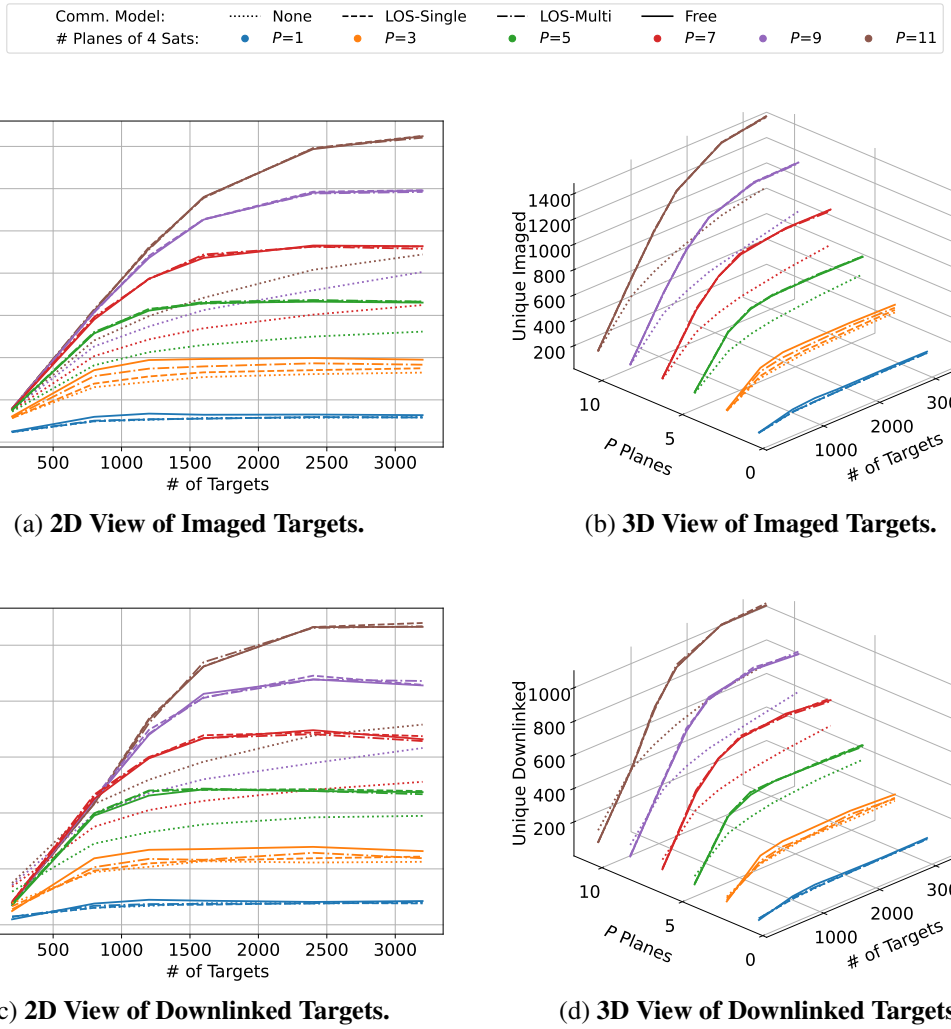
(c) **2D View of Local Reward.**

(d) **3D View of Local Reward.**

Figure 9: **Global and Local Reward for the Multiple Plane Experiment.**

and 9d), in which the per-satellite maximum of ~ 0.45 is more quickly approached with respect to number of global targets across all constellations and communication models. The best-case model, free communication, outperforms the worst-case model, no communication, in reward by a similar factor of $\sim 1.3\times$ in the single plane and multi-plane cases alike. Similarly, multi-plane unique images and downlinks (Figure 10) are proportional with each other and with global reward, as in the single plane cases.

The two line-of-sight communication models follow predicted trends. For constellations with $P > P^*$ ($P \geq 5$), the amount of inter-plane communication was sufficient for both the single and multi-degree line-of-sight models to perform equal to the free communication case. This result is stronger than the P^* analysis implied, since $P > P^*$ indicates that two satellites at the same latitude on adjacent planes can always communicate, but does not make any guarantees that satellites are frequently adjacent. Since satellites in adjacent planes orbit retrograde and at higher latitudes all of the planes are closer together, it is unsurprising that the constellation maintains a well connected



(a) **2D View of Imaged Targets.**

(b) **3D View of Imaged Targets.**

(c) **2D View of Downlinked Targets.**

(d) **3D View of Downlinked Targets.**

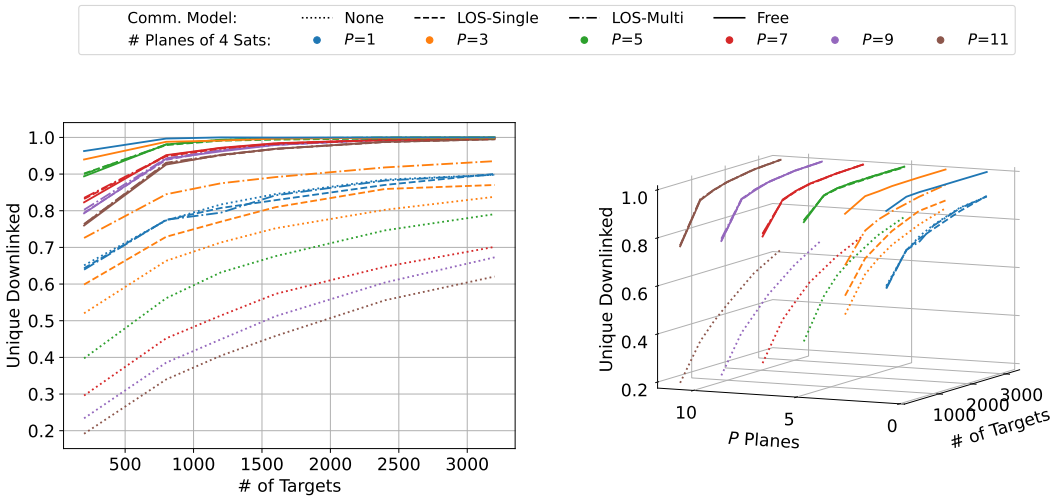
Figure 10: **Global Numbers of Unique Imaged and Downlinked Targets for the Multiple Plane Experiment.**

network.

For the single nontrivial case where $P < P^*$, $P = 3$, the two line-of-sight models do not collapse to either the free or no communication models, which is most easily seen in Figures 10a, 11a, and 11b. This is the only set of parameters in these experiments that produce this result. Multi-degree line-of-sight outperforms single degree which implies that the constellation sometime forms chains of line-of-sight connections that more effectively distribute information.

Target Duplication General trends in target duplication in Figure 11 are again similar to the single plane cases: more targets and fewer satellites leads to less duplication. Just as the multi-plane experiments yielded superior reward compared to single plane experiments, multi-plane cases with (pseudo-)free communication have fewer duplicated images than a similar-sized single plane constellation: more of the global targets are available to a multi-plane constellation, so statistically fewer duplicates will occur.

The nature of target duplication in the line-of-sight $P = 3$ cases is of note. Since the line-of-sight models do not match the performance of free communication, there are frequent occurrences where satellites pass over a region imaged by another satellite without having received information from that satellite first. These duplicates can be attributed to a lack of information and not competition in a single step since line-of-sight communication suffers from the latter to the same degree as free communication, and with $P = 3$ free communication produces very few duplicates.



(a) **2D View.** (b) **3D View.**

Figure 11: **Percent of Imaged Targets that are Unique for the Multiple Plane Experiment.**

CONCLUSIONS

This work explores the deployment of independently trained decision-making agents in a Walker-Delta satellite constellation under various communication assumptions for Earth-observing satellite scheduling. The decision-making agents are trained in a single satellite environment, but deployed as a part of a multi-satellite environment where each agent retains its own copy of the policy. The agents communicate with one another to update their lists of targets. The performance of the agents is benchmarked for various communication assumptions that include free communication, single degree line-of-sight communication, multi-degree line-of-sight communication, and no communication. In all cases, the free communication assumption achieves the highest global and local reward and least amount of duplication of efforts. Likewise, the no communication assumption achieves the lowest global and local reward and most amount of target duplication. In a single-plane constellation, the two line-of-sight assumptions match the performance of the no communication assumption until some critical number of satellites, $N^* = 9.2$, is reached. After that point, the line-of-sight assumptions match the performance of the free communication case. This intuitively makes sense because for values of N less than N^* the satellites never communicate. In the multi-plane case, a similar phenomenon is demonstrated based on the ability of satellites in different planes to communicate with one another at the equator. The critical number of planes is shown analytically and experimentally to be $P^* = 4.6$.

Future work will determine an upper bound on performance if the agents coordinate with one another to maximize global reward, not just their own local reward. This will motivate the exploration

of problem formulations where agents coordinate with one another multiple orbits ahead. Furthermore, future work will explore alternative problem formulations with variable timestep imaging modes. In theory, the imaging modes only need to last as long as it takes to capture an image, at which point a new imaging target can be targeted.

ACKNOWLEDGEMENT

This work is partially supported by a NASA Space Technology Graduate Research Opportunity (NSTGRO) grant, 80NSSC20K1162. This work is also partially supported by the Air Force Research Lab grant FA9453-22-2-0050.

REFERENCES

[1] D.-H. Cho, J.-H. Kim, H.-L. Choi, and J. Ahn, "Optimization-Based Scheduling Method for Agile Earth-Observing Satellite Constellation," Journal of Aerospace Information Systems, Vol. 15, No. 11, 2018, pp. 611–626, 10.2514/1.I010620.

[2] J. Kim, J. Ahn, H.-L. Choi, and D.-H. Cho, "Task Scheduling of Agile Satellites with Transition Time and Stereoscopic Imaging Constraints," Journal of Aerospace Information Systems, Vol. 17, No. 6, 2020, pp. 285–293.

[3] X. Chen, G. Reinelt, G. Dai, and A. Spitz, "A Mixed Integer Linear Programming Model for Multi-Satellite Scheduling," European Journal of Operational Research, Vol. 275, No. 2, 2019, pp. 694–707, https://doi.org/10.1016/j.ejor.2018.11.058.

[4] J. Kim and J. Ahn, "Integrated Framework for Task Scheduling and Attitude Control of Multiple Agile Satellites," Journal of Aerospace Information Systems, Vol. 18, No. 8, 2021, pp. 539–552.

[5] V. Shah, V. Vittaldev, L. Stepan, and C. Foster, "Scheduling the World's Largest Earth-observing Fleet of Medium-resolution Imaging Satellites," International Workshop on Planning and Scheduling for Space, 2019.

[6] J. Cappaert, F. Foston, P. S. Heras, B. King, N. Pascucci, J. Reilly, C. Brown, J. Pitzo, and M. Tallhamm, "Constellation Modelling, Performance Prediction and Operations Management for the Spire Constellation," SmallSat Conference, 2021.

[7] 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, 2005, pp. 196–216, 10.2514/1.12923.

[8] S. Chien, D. Tran, G. Rabideau, S. Schaffer, D. Mandl, and S. Frye, "Timeline-based Space Operations Scheduling with External Constraints," Twentieth International Conference on Automated Planning and Scheduling, 2010.

[9] S. A. Chien, A. G. Davies, J. Doubleday, D. Q. Tran, D. Mclaren, W. Chi, and A. Maillard, "Automated Volcano Monitoring Using Multiple Space and Ground Sensors," Journal of Aerospace Information Systems, Vol. 17, No. 4, 2020, pp. 214–228, 10.2514/1.I010798.

[10] 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, 2019, pp. 107–119, 10.2514/1.I010672.

[11] C. G. Valicka, D. Garcia, A. Staid, J.-P. Watson, G. Hackebeil, S. Rathinam, and L. Ntaimo, "Mixed-Integer Programming Models for Optimal Constellation Scheduling Given Cloud Cover Uncertainty," European Journal of Operational Research, Vol. 275, No. 2, 2019, pp. 431–445.

[12] A. K. Kennedy, Planning and Scheduling for Earth-Observing Small Satellite Constellations. PhD thesis, Massachusetts Institute of Technology, 2018.

[13] M. Dahl, J. Chew, and K. Cahoy, "Optimization of SmallSat Constellations and Low Cost Hardware to Utilize Onboard Planning," ASCEND 2021, p. 4172, 2021.

[14] L. Wei, Y. Chen, M. Chen, and Y. Chen, "Deep Reinforcement Learning and Parameter Transfer Based Approach for the Multi-Objective Agile Earth Observation Satellite Scheduling Problem," Applied Soft Computing, Vol. 110, 2021, p. 107607.

[15] K. Cui, J. Song, L. Zhang, Y. Tao, W. Liu, and D. Shi, "Event-Triggered Deep Reinforcement Learning for Dynamic Task Scheduling in Multi-Satellite Resource Allocation," IEEE Transactions on Aerospace and Electronic Systems, 2022.

[16] L. Dalin, W. Haijiao, Y. Zhen, G. Yanfeng, and S. Shi, "An Online Distributed Satellite Cooperative Observation Scheduling Algorithm Based on Multiagent Deep Reinforcement Learning," IEEE Geoscience and Remote Sensing Letters, Vol. 18, No. 11, 2020, pp. 1901–1905.

[17] C. Amato, G. Konidaris, G. Cruz, C. A. Maynor, J. P. How, and L. P. Kaelbling, "Planning for Decentralized Control of Multiple Robots under Uncertainty," 2015 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2015, pp. 1241–1248.

[18] S. Omidshafiei, A.-A. Agha-Mohammadi, C. Amato, S.-Y. Liu, J. P. How, and J. Vian, "Decentralized Control of Multi-Robot Partially Observable Markov Decision Processes using Belief Space Macro-actions," The International Journal of Robotics Research, Vol. 36, No. 2, 2017, pp. 231–258.

[19] L. Matignon, L. Jeanpierre, and A.-I. Mouaddib, "Coordinated Multi-Robot Exploration under Communication Constraints using Decentralized Markov Decision Processes," Twenty-sixth AAAI conference on artificial intelligence, 2012.

[20] F. A. Oliehoek and C. Amato, A Concise Introduction to Decentralized POMDPs. Springer, 2016.

[21] M. J. Kochenderfer, T. A. Wheeler, and K. H. Wray, Algorithms for Decision Making. MIT Press, 2022.

[22] C. Boutilier, "Sequential Optimality and Coordination in Multiagent Systems," IJCAI, Vol. 99, 1999, pp. 478–485.

[23] C. H. Papadimitriou and J. N. Tsitsiklis, "The Complexity of Markov Decision Processes," Mathematics of Operations Research, Vol. 12, No. 3, 1987, pp. 441–450.

[24] A. Herrmann and H. Schaub, "Reinforcement Learning for the Multi-Satellite Earth-Observing Scheduling Problem," AAS Guidance and Controls Conference, Breckenridge, CO, Feb. 3-9 2022.

[25] A. Herrmann and H. Schaub, "Autonomous On-Board Planning for Earth-Orbiting Spacecraft," IEEE Aerospace Conference, Big Sky, MT, March 5-12 2022.

[26] P. W. Kenneally, S. Piggott, and H. Schaub, "Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework," Journal of Aerospace Information Systems, Vol. 17, Sept. 2020, pp. 496–507.

[27] A. P. Herrmann and H. Schaub, "Monte Carlo Tree Search Methods for the Earth-Observing Satellite Scheduling Problem," Journal of Aerospace Information Systems, 2021, pp. 1–13, 10.2514/1.I010992.