Cislunar Space Situational Awareness Sensor Tasking using Deep Reinforcement Learning Agents

Peng Mun Siew[*], **Daniel Jang**[†], **Thomas G. Roberts**[‡], **Richard Linares**[§]
Massachusetts Institute of Technology
Justin Fletcher[¶]
United States Space Force Space Systems Command

ABSTRACT

Cislunar space is gaining popularity with numerous missions being planned for the near future. However, operating in the cislunar space domain poses additional risk to satellites due to the lack of space situational awareness in the regime. Without maintaining a proper catalog of resident space objects in cislunar space, active space assets are susceptible to catastrophic collisions with untracked resident space objects (RSOs). The cislunar orbital regime is unique in that propagation of orbits is not easily predicted nor learned due to the complex three-body dynamics. In this work, we explored the usage of a deep reinforcement learning agent to optimally task a narrow field of view ground-based optical telescope for cislunar space situational awareness. The performance of our trained agent is compared to two random policies; a policy that randomly select a direction to observe and a policy that randomly select a RSO within the field of regard to observe.

1. INTRODUCTION

Previously published research demonstrates the promise of using deep reinforcement learning (DRL) to train ground- and space-based narrow field of view (FOV) sensors to efficiently observe space objects in the near-Earth space environment, including those in low Earth orbit (LEO) and the geosynchronous (GEO) orbital regime [1, 2]. For these orbital regimes, modeled sensors informed by DRL-trained agents have been shown to outperform those informed by myopic policies, observing higher numbers of cumulative space objects during various observation periods while reporting lower average covariances across all objects in the environment [3–5]. This work continues to depend on the proximal policy optimization algorithm and population-based training (PBT) for DRL, but adapts the space situational awareness (SSA) environment formulation described in [5] to develop an agent for observing objects in cislunar space from the Earth's surface, with a particular focus on those in residence in Earth-Moon L1 halo orbits, Earth-Moon L2 halo orbits, distant retrograde orbits (DRO), and particular Earth-Moon resonance orbits. This work differs from a previous study by Fedeler, et al. [6] dedicated to sensor tasking for cislunar space object observation, which used a Monte Carlo Tree Search algorithm for pointing direction decision making and evaluated both single and multiple non-Earth-based observers, including those in L1, L2, and L4 halo orbits and Earth and Moon orbits. When evaluating observability metrics for cislunar domain awareness, some studies include Earth-based observer locations, including Fowler, et al. [7], while others only evaluate observers that are themselves in one of several cislunar orbits [8].

Although cislunar orbits offer attractive operational benefits (described in greater detail in section 3) they present unique challenges for ground-based observation. Objects in cislunar space are much farther away than those in LEO or GEO—they are often observed from the Earth's surface at distances more than ten times greater than GEO altitude [7]—making them appear smaller and dimmer in the sky. In addition, many of the cislunar orbits discussed as part of this work have large orbital periods compared to the study's observation period, making objects appear to move slowly when observed from the Earth's surface and requiring a greater number of observations for orbit determination.

[*]Postdoctoral Associate, Department of Aeronautics and Astronautics. E-mail: siewpm@mit.edu
[†]Ph.D. Candidate, Department of Aeronautics and Astronautics. E-mail: djang@mit.edu
[‡]Ph.D. Candidate, Department of Aeronautics and Astronautics. E-mail: thomasgr@mit.edu
[§]Associate Professor, Department of Aeronautics and Astronautics. E-mail: linaresr@mit.edu
[¶]Program Support to SMC/SPG. E-mail: justinfletcher@odysseyconsult.com

Perhaps most critically, objects in cislunar orbit often appear relatively close to the Moon's position in the night sky; objects that are too close to the illuminated Moon cannot be observed due to the bright sunlight reflected off of the lunar surface [9]. Lastly, due to the gravitational force of the Moon acting on space objects, the orbital dynamics of cislunar space are complex and are not easily predicted or learned by the trained agent.

Despite these challenges, the DRL-trained agent continues to perform well, achieving low final mean uncertainties (mean trace covariance) for all RSOs and visiting a higher number of unique RSOs compared to random policies over the two-hour observation window.

2. CIRCULAR RESTRICTED THREE BODY PROBLEM

Satellites in Earth's orbit face a variety of perturbative forces that add complexity to their orbital dynamics and cause deviations from the simple two-body Keplerian model for satellite motion: atmospheric drag; perturbations from the Earth's geopotential coefficients; the gravitational forces from the Moon, Sun, and other celestial bodies; solid Earth tides; and solar radiation pressure, among others. For satellites at low altitudes, atmospheric drag is often the highest-magnitude perturbative force. At higher altitudes, in which many Earth-orbiting satellites operate, perturbations due to the J_2 geopotential coefficient—which describes the oblateness of the Earth's shape—are the highest-magnitude perturbative force. But for objects orbiting the Earth at altitudes higher than the GEO altitude, including those being observed as part of this study, it is the Moon's gravitational force that poses the highest-magnitude perturbative force [10].

Since objects in cislunar space are constantly under the influence of both the Earth and Moon's gravitational forces—which is not the case for objects that merely pass through this region as part of an interplanetary trajectory—their motion should not be modeled with two-body dynamics perturbed by external forces, but rather a simplified three-body framework [11]. Since the mass of satellites in cislunar space are negligible compared to the mass of the Earth and Moon, and the Earth and Moon move in approximately circular orbits about the Earth-Moon center of mass (the barycenter), the problem of modeling the satellites' orbital dynamics can be approximated by the *circular restricted three-body problem* (CR3BP).

The CR3BP problem is modeled under the assumption that the Earth, Moon, and a cislunar-orbiting satellite are point masses [12]. In the Earth-Moon system's *synodic coordinate frame*, the Earth-Moon barycenter lies at the origin, the Earth and Moon lie fixed on the x-axis (with the positive $\hat{x}$ vector pointed at the Moon), the y-axis lies in the Earth-Moon orbital plane (such that the frame's rotation, ω, is positive), and the z-axis is defined using the right-hand rule. The problem is normalized by using a mass ratio $\mu^* = M_M/M_E + M_M$—where M_E and M_M are the masses of the Earth and Moon, respectively—to define the vectors from the barycenter (B) to the Earth and Moon: $\vec{r}_{BE} = 1-\mu^*$ and $\vec{r}_{BM} = \mu^*$. The synodic coordinate frame for the Earth-Moon system, including the position of a third-body space object ($\vec{r}_{BS}$) and its relative positions from the Earth and Moon ($\vec{r}_{ES}$ and $\vec{r}_{MS}$, respectively) are depicted in Fig. 1.

The components of the rotating acceleration $\ddot{\vec{r}}_{BS}$ on the space object can be written [11]:

$$\ddot{x} = -\frac{(1-\mu^*)(x+r_{BE}\cos\omega t)}{r_{ES}^3} - \frac{\mu^*(x-r_{BM}\cos\omega t)}{r_{MS}^3} \tag{1}$$

$$\ddot{y} = -\frac{(1-\mu^*)(y+r_{BE}\sin\omega t)}{r_{ES}^3} - \frac{\mu^*(y-r_{BM}\sin\omega t)}{r_{MS}^3} \tag{2}$$

$$\ddot{z} = -\frac{(1-\mu^*)z}{r_{ES}^3} - \frac{\mu^* z}{r_{MS}^3} \tag{3}$$

The distances between the space object and the two principal gravitational bodies can be written:

$$r_{ES} = \sqrt{(x+r_{BE}\cos\omega t)^2 + (y+r_{BE}\sin\omega t)^2 + z^2} \tag{4}$$

$$r_{MS} = \sqrt{(x-r_{BM}\cos\omega t)^2 + (y-r_{BM}\sin\omega t)^2 + z^2} \tag{5}$$

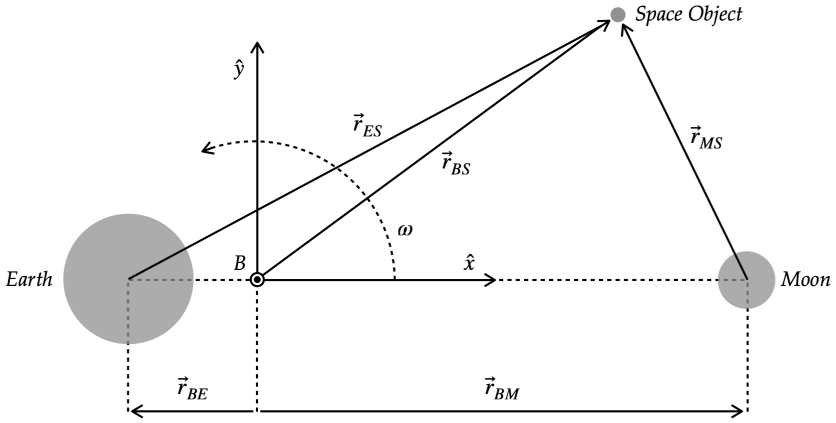


Fig. 1: Earth-Moon synodic coordinate frame. The relative size and separation of the Earth and Moon and the location of the system's barycenter have been adjusted for clarity. The true Earth-Moon barycenter is located within the Earth's terrestrial volume.

3. THREE-BODY PERIODIC ORBITS IN THE CISLUNAR SPACE

In any CR3BP system, there are five equilibrium positions in which objects' velocity and acceleration can remain at zero in all three dimensions in the synodic frame: the *Lagrange points* [11, 13]. Of the five Lagrange points in the Earth-Moon system, two are most critical in this study: L1 and L2. Both L1 and L2 lie on the x-axis in the synodic frame, at approximately 318379 km and 441076 km, respectively [14]. In the purely stable CR3BP, space objects can be placed in *halo* orbits around L1 and L2, in which they appear to periodically orbit their respective Lagrange point in the synodic frame, with periods measured in days. In the true cislunar environment, in which the CR3BP conditions are not perfectly met, space objects can maintain L1 and L2 halo orbits with minimal station-keeping. L1 and L2 halo orbits provide a range of operational benefits, including continuous line-of-sight visibility from Earth [15] and favorable conditions for human space missions [16].

Because of these operational benefits, the L1 and L2 orbital classes are particularly likely to host more space objects in the future [17] and accordingly were selected for study in this work. To diversify the objects being observed, two more orbital classes with their own unique operational benefits were also considered: distant retrograde orbits (DRO) and 3:1 resonance orbits. Distant retrograde orbits are relatively predictable, operate in a favorable deep space environment [18], and are unique in that they appear from observers on the Moon to orbit in retrograde [19]. Objects in a 3:1 resonant orbit in the Earth-Moon system orbit the Earth three times per lunar period [20]. Such an orbit is favorable for its stability and ease of observation via ground-based systems due to its relatively low perigees [21]. See Fig. 3 for visualizations of these four orbital classes in the synodic reference frame.

4. SSA ENVIRONMENT FORMULATION

A custom cislunar space situational awareness (SSA) environment is constructed using the OpenAI Gym library [22]. The cislunar SSA environment is responsible for keeping track of the RSOs' states, generating noisy measurements for RSOs within the sensor's field of view, propagating and updating the RSOs' covariances, creating the observation array for the DRL agent, and computing the instantaneous reward of the current action. Here, a distinction is made between measurement and observation array, where measurement is used to refer to the noisy angles and angle rates measured by the ground-based sensor for all RSOs within the sensor's current field of view and the observation array refers to the current state of the environment that is accessible to the DRL agent. More information on the observation array is included in section 5.3. The RSOs are propagated using the circular restricted three-body problem formulation without any external perturbations. Each rollout of an episode is limited to a two-hour finite horizon scenario. The Unscented

Kalman Filter (UKF) formulation is used to propagate and update the RSOs' covariances. The initial covariance of the RSOs at the start of each episode is uniformly initialized within the range shown in Table 1.

Table 1: Initialization of RSO covariance

State	Sampling Range	
	Lower limit	Upper limit
x	100 km^2	400 km^2
y	100 km^2	400 km^2
z	100 km^2	400 km^2
V_x	0.01 km^2/s^2	0.04 km^2/s^2
V_y	0.01 km^2/s^2	0.04 km^2/s^2
V_z	0.01 km^2/s^2	0.04 km^2/s^2

4.1 Sensor parameters

The sensor parameters were largely taken from the Pan-STARRS system [23–25]. The system is located at the Haleakala Observatory, Hawaii, US, and surveys the sky for celestial objects and near-Earth objects. Though traditionally not used for the SSA mission other than enabling high-precision star catalogs [26], such sensitive telescopes will likely be needed for reliable observation of the distant cislunar RSOs. Of the several telescopes in the Pan-STARRS system, the 1.8-meter diameter Pan-STARRS1 Telescope (PS1) is used to model our sensor, which has a field-of-view diameter of 3 degrees.

Table 2: Modeled ground-based SSA sensor parameters

Parameter	Value
FOV	$3^\circ \times 3^\circ$
Slew rate	5°/s
Observations	Angles and angles rates
Lunar exclusion angle	5°
Sensor location	Haleakala Observatory, Maui, HI
Sensor coordinate	20.7082° N, 156.2561° W, 3052 km

The sensor is assumed to have a $3^\circ \times 3^\circ$ field of view (FOV) which can point anywhere in the 4π steradian sphere down to 12° elevation due to terrain and observational quality limitations. Here, a minimum elevation pointing angle of 12° is used (i.e., the sensor is not able to point below 12° from the local horizon). With 360° in azimuth and 78° in elevation to cover due to the minimum viewing horizon of 12°, this field of regard (FOR) can be fully covered by a 120×26 grid for a total of 3240 possible pointing directions. The sensor's slew rate is assumed to be 5°/s with a settle time of 10 s and an observational time of 60 s. The geometry of the sensor's FOR and slew duration is shown in Fig. 2.

4.2 Cislunar RSO initialization

For the scenarios evaluated as part of this study, objects were initialized into the following cislunar periodic orbit families: L1 halo orbits, L2 halo orbits, distant retrograde orbits, and Earth-Moon 3:1 resonant orbits. To this end, precomputed periodic solutions provided in the NASA Jet Propulsion Laboratory's three-body periodic orbit catalog [20] were used. However, most of the space objects in the periodic orbit catalog do not enter the sensor's field of regard or are obstructed by the Moon's exclusion zone during the two-hour observation window, which results in wasted computational time in propagating cislunar objects that are not of interest. Furthermore, the catalog only provides an initial state of each periodic orbit that lies on the X-Y plane of the Earth-Moon system, whereas larger variability in the initial condition of the cislunar space objects is more preferable for DRL training and evaluation. In order to make

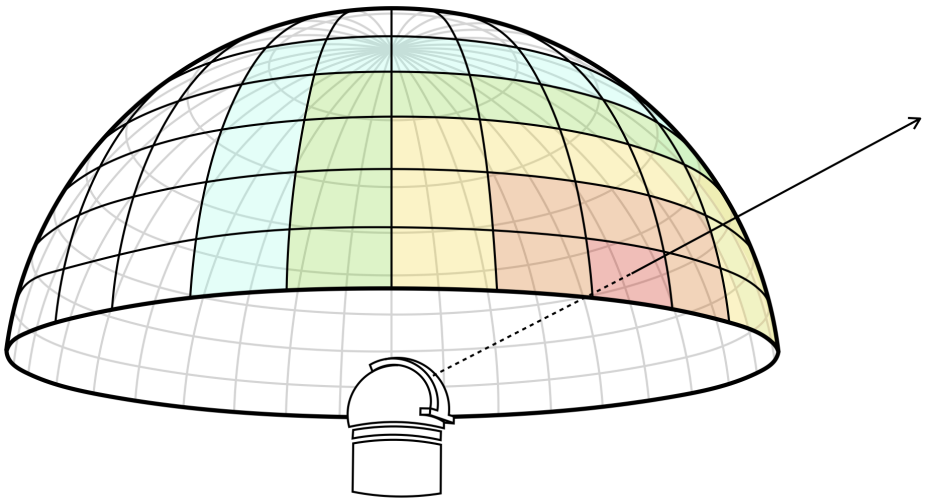


Fig. 2: Nominal field of regard for a ground-based SSA sensor.

the problem more challenging with a higher number of possible rewarding actions (i.e. having more space objects in the field of regard at any instance), a new database is constructed by resampling the periodic orbit catalog. This is achieved by propagating the initial states within the periodic orbit catalog for a full period and identifying the portion of the trajectories that fall within our field of regard. A new database is then constructed by randomly sampling the states within the observable portion of the trajectories. Fig. 3 shows the trajectories of 20 tracked cislunar space objects for a full period in the synodic frame.

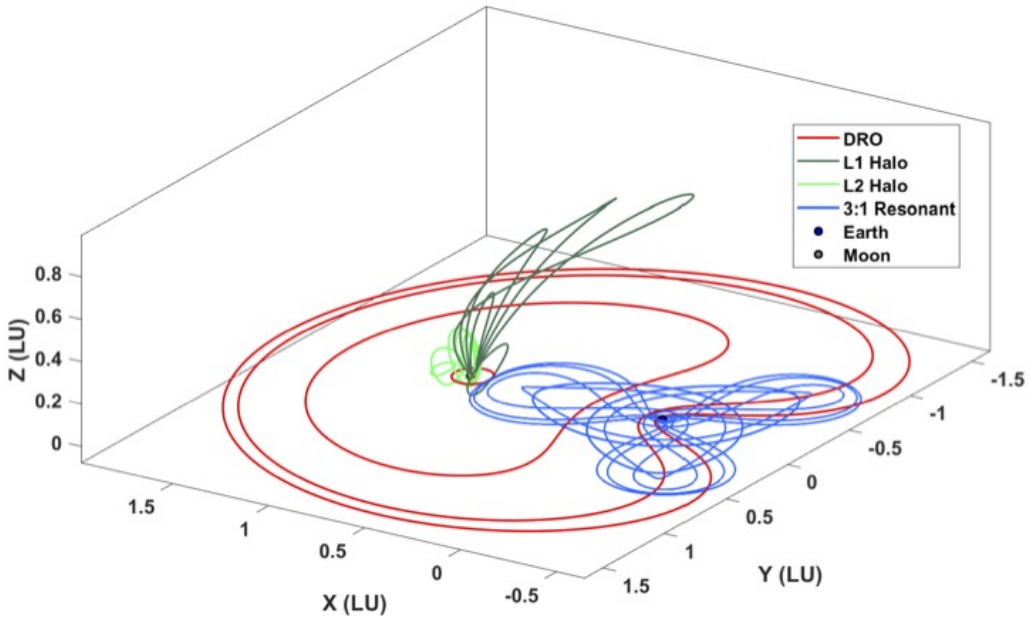


Fig. 3: Various cislunar orbits propagated for the full orbital period for the 4 types of orbits shown in this paper. 1 LU (Lunar Unit) = 389,703 km.

4.3 Sensor action

At each time step, the environment outputs an observation (a representation of the state of the environment) and queries the agent or policy for a new sensor action (pointing direction). The flowchart for this process at each time step is shown in Fig. 4. The required action time (slew, settle, and dwell time) is then computed based on the new sensor

action. The RSOs' states and covariance are then propagated forward in time based on the required action time using the CR3BP formulation. RSOs that are located in the sensor's FOV are then identified and noisy angle and angle rate measurements are generated for each of these RSOs. The sensor is assumed to be able to perfectly assign each noisy measurement to the correct RSO. A UKF update is then carried out to update the RSO's states and covariance. A new observation is then generated by the environment and the whole process is repeated.

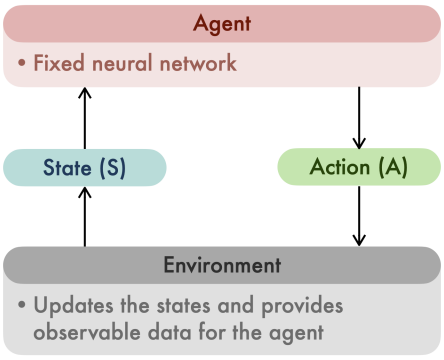


Fig. 4: Evaluation process for a single-sensor DRL agent

4.4 Reward Function

The reward function plays a significant role in the training of the DRL agents. It functions as an incentive mechanism to guide the DRL agent to a desirable environment state or action. A poorly designed reward function can lead to slow training and bad performance. The reward function needs to be tailored to the specific objective to be achieved in a concept known as reward shaping. In the cislunar sensor-tasking scenario, there can be a multitude of objectives, such as minimizing the revisit time of all RSOs within the catalog, maximizing the number of unique RSOs observed over a defined observation window, or maintaining the maximum uncertainties of any RSO to be below a certain threshold. For any of these objectives, the reward function can be modified such that the trained agent performs well with regard to that particular performance metric. It is also possible to have a reward function that is tailored for a weighted performance metric that combines multiple user objectives. In this paper, the main objective is to minimize the total uncertainty over all RSOs, i.e., minimize the mean trace covariance across all RSOs at the end of the observation window.

In the paper, the usage of two different reward functions was explored. The same reward function is used throughout the scenario without a separate final reward at the end of the observation window. The first reward function R_1 is based on information gain theory, where the reward is proportional to the reduction in the trace of RSO covariance. The second reward function R_2 is a time-discounted variant of the previous reward function R_1, where the reward is discounted by the action slew time (action cost). This will encourage the agent to minimize slewing and select nearby RSOs with high uncertainties to be observed. Both reward functions are scaled such that the reward values are in the order of 1 and the agent is penalized when an empty FOV (i.e without any RSO) is selected. The reward functions R_1 and R_2 are shown in Equations 6 and 7 respectively.

$$R_1(a_k) = \begin{cases} \arg\max_{i_k} \left[tr\left(P_{k|k-1}^{(i_k)}\right) - tr\left(P_{k|k}^{(i_k)}\right) \right] /40, & \text{if } i_k \neq \emptyset \\ -10, & \text{otherwise} \end{cases} \tag{6}$$

$$R_2(a_k) = \begin{cases} \arg\max_{i_k} \left[tr\left(P_{k|k-1}^{(i_k)}\right) - tr\left(P_{k|k}^{(i_k)}\right) \right] /0.5\delta t, & \text{if } i_k \neq \emptyset \\ -10, & \text{otherwise} \end{cases} \tag{7}$$

where i_k are RSOs within the FOV selected by action a_k, $tr\left(P_{k|k-1}^{(i)}\right)$ and $tr\left(P_{k|k}^{(i)}\right)$ are the trace of the prior and posterior covariance for RSO i, respectively, and δt is the sum of the required slew, settle, and dwell times for action a_k.

5. DEEP REINFORCEMENT LEARNING FORMULATION

The objective of the SSA system is to reduce the mean covariance of the RSO population during each episode. Two figures of merit are used to compare performance: (1) the number of RSOs observed and (2) the mean covariance of all RSOs during the episode. The PBT framework [27] was implemented using the Ray and Tune libraries [28]. Proximal Policy Optimization, a model-free DRL algorithm was also used [29]. Lastly, the hyperparameters for DRL training were taken from [3, 4].

5.1 Neural Network Formulations

The DRL agents explored in this work use an actor-critic architecture motivated by the neural network architectures used in [4, 5]. The actor-critic architecture consists of an actor network and a critic network. The actor network is responsible for generating the action policy π, whereas the critic network functions to provide an assessment of the action selected by the actor via the value function. Table 3 shows the neural network architecture used in this work. `Conv2d(32,8,4)` corresponds to using a kernel with 32 filters, an 8×8 sliding window, and a 4×4 stride. `FCL(1024)` corresponds to a fully connected layer with 1024 nodes.

Table 3: Neural network architecture

Architecture	Input Size	Output Size	Actor Model	Critic Model
`CNN_v1`	$120 \times 26 \times 11$	3120	`Conv2d(32,8,4)`, `Conv2d(64,4,2)`, `Conv2d(64,3,1)`, `FCL(1024)`, `FCL(3120)`	`Conv2d(32,8,4)`, `Conv2d(64,4,2)`, `Conv2d(64,3,1)`, `FCL(512)`, `FCL(1)`

Fig. 5 shows the TensorFlow dataflow graph of the `CNN_v1` neural network architecture. The actor network takes in the $120 \times 26 \times 11$ observation array, described in section 5.3, and outputs a probability distribution over the 3120 possible actions. The actor model consists of two parallel dataflow paths. The main dataflow path on the left has the structure shown in Table 3 and is responsible for calculating the action probability for all possible actions. On the other hand, the secondary dataflow path on the right corresponds to the action-masking function. The action-masking function is responsible for penalizing actions that result in an empty FOV. Thus, it encourages the action policy to focus on actions that can lead to an occupied FOV. Similarly, the critic model takes in the $120 \times 26 \times 11$ observation array and outputs the value function for the current state and action. The value function can be utilized to improve the action policy.

5.2 Action Space

The action space corresponds to the set of all possible actions that the agent can select. In the cislunar SSA scenario described in the previous section, the action space consists of all possible pointing directions of the ground-based sensor. In order to simplify the problem, the continuous action space is first discretized into a discrete action space based on the FOV of the sensor. Based on the sensor parameters in section 4.1, the $360^\circ \times 78^\circ$ field of regard is discretized into $3^\circ \times 3^\circ$ grids. Note, a minimum elevation pointing angle of 12° is used (i.e., the sensor is not able to point below 12° from the local horizon). This yields an action space that is represented by a 120×26 grid for elevation pointing directions and azimuth pointing directions, respectively. This results in 3120 possible distinct actions.

5.3 Observation Space

Similarly, the observation space is also discretized into $3^\circ \times 3^\circ$ grids. The discretization of the observation space allows for the same formulation to be used regardless of the actual number of RSOs within the field of regard. The observation array consists of a three-dimensional array with dimensions $120 \times 26 \times 11$. The first two dimensions represent the possible pointing directions and the third dimension corresponds to the data layer index. When multiple RSOs are located within the same grid, only the values corresponding to the RSO with the largest uncertainties are used. Each row in the observation array represents a pointing azimuth, while each column represents a pointing elevation. The observation array is populated such that the current pointing direction is always in the 60^{th} row (i.e, the

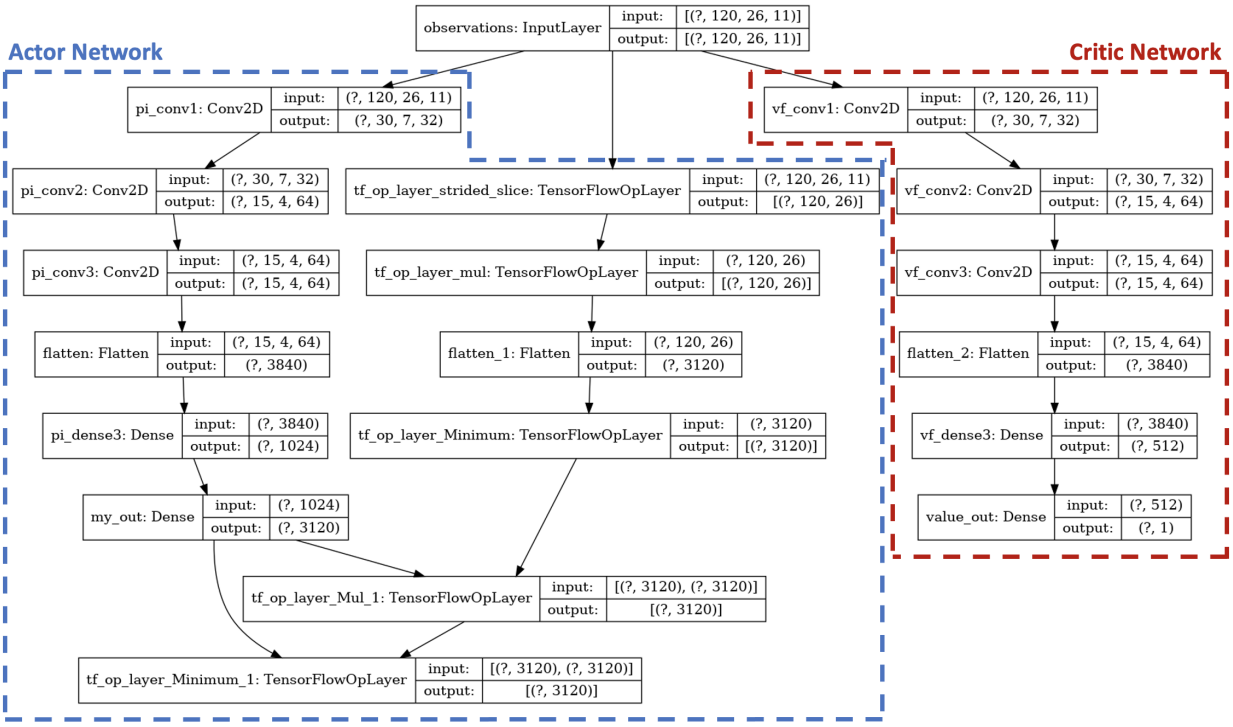


Fig. 5: Neural network architecture for `CNN_v1`

center row of the observation array). The 11 observation data for each grid is shown in Table 4. The current pointing direction is a Boolean value, where it is 1 if the current grid is the current pointing direction and 0 otherwise.

Table 4: Observation information for each grid

Layer	Data per observation grid
1	Number of RSOs
2	Elevation fraction location of RSO
3	Azimuth fraction location of RSO
4	Range of RSO
5	Elevation velocity of RSO
6	Azimuth velocity of RSO
7	Range velocity of RSO
8	Max trace covariance of RSOs
9	Sum of RSOs trace covariance
10	Mean of RSOs trace covariance
11	Current pointing direction

Due to the large range of action time (action slew, settle, and dwell time), the observation grid is partitioned into 6 regions as shown in Fig. 6. The data for each region is populated with a different propagation time based on each region's approximate action slew duration. Different propagation times are used to better reflect the true expected relative position of the RSOs and more accurately reflect what the agent is expected to observe.

5.4 Training

The DRL agents are implemented using Tensorflow [30] and the Ray Tune toolbox [28]. The DRL agents are trained using proximal policy optimization with PBT. The PBT framework allows for the joint optimization of the training hyperparameters and the neural network. The training of DRL agents is sensitive to training hyperparameters, such as

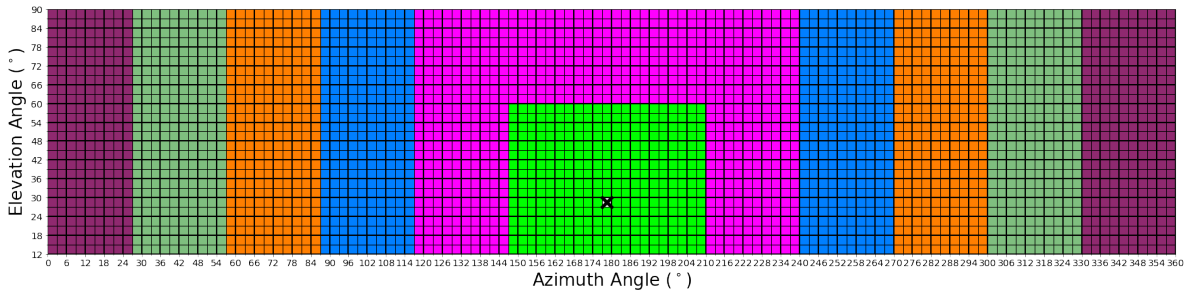


Fig. 6: Field of regard and propagation gradient.

learning rate, training batch size, and mini-batch size, where non-optimal training hyperparameters can lead to slow learning and sometimes even catastrophic failure and divergence of the DRL agent. The PBT framework overcomes this issue by training a population of DRL agents at the same time, each having a set of slightly different training hyperparameters. Poor-performing DRL agents are then replaced with a duplicate of the top-performing agent with slight perturbations in their training hyperparameters. The PBT framework has previously been shown to improve convergence and achieve a higher final reward for a suite of challenging problems [27].

Both the training and evaluation of the DRL agent was completed using a single compute node with 48 CPU cores at the MIT Lincoln Laboratory Supercomputing Center (LLSC) [31]. The initialization and mutation sampling range of the training hyperparameters used are shown in Table 5, where *Randint* and *Uniform* indicate random integer sampling and random uniform sampling within the provided range, respectively.

Table 5: Initialization and mutation range of hyperparameters for hyperparameter optimization with PBT for the DRL agents

Hyperparameter	Initialization	Mutation Sampling Range
Learning rate	[3.75e-4, 5e-3]	*Uniform*(1e-6, 1e-2)
Minibatch size	*Randint*(128, 768)	*Randint*(128, 768)
Batch size	*Randint*(1024, 3072)	*Randint*(1024, 3072)
Entropy coefficient	[1e-1, 1e-2]	*Uniform*(1e-5, 1e-1)

Under the PBT framework in this study, seven DRL agents are trained concurrently using the PBT framework. After every 100 training iterations, the performances of the trained agents are ranked based on their mean episodic reward. Poor-performing DRL agents (those with a mean episodic reward that ranks in the bottom 30%) are replaced by the top-performing DRL agents. The training hyperparameters of these duplicated top-performing agents are then varied with a 25% probability to be re-sampled from the mutation range provided in Table 5 or to be perturbed from their current values.

Fig. 7 shows the mean episodic reward for the `CNN_v1` DRL agent trained using the different reward functions outlined in section 4.4. The `CNN_v1` DRL agent was trained on reward function R_1 and reward function R_2 for 700 training iterations. At zero training iterations, both DRL agents frequently selected empty FOV to observe, hence resulting in a large negative mean episodic reward as both reward functions penalize the DRL agent when an empty FOV is selected. The mean episodic reward plots for both DRL agents show logarithmic growth where the mean episodic reward increases quickly in the beginning, but the gains decrease and become more difficult as time progresses. During the initial stages, the DRL agents were able to significantly improve their performance by learning to avoid empty FOV to observe. The improvement in performance gradually decreases once the DRL agents learned to filter out empty FOV. At this point in time, the DRL agents require more training iterations to obtain noticeable performance gain due to the complex nonlinearity nature of the cislunar SSA problem. It can also be observed in Fig. 7 that the DRL agents have not fully converged yet during this training period; it can be seen that the mean episodic reward is still increasing with the number of training iterations. For the agent trained using reward function R_1, the training hyperparameters of this DRL agent were mutated once at 600 training iterations by the PBT framework. Here we can

notice the mean episodic reward plot jumping back from the 600^{th} training iterations to the 500^{th} training iterations. Prior to the mutation, the PBT framework noticed that the current agent was performing subpar compared to the rest of the population at the 500^{th} training iterations. The PBT framework will then duplicate one of the better-performing agents to replace the current lower-performing agent. When the least performing agent is replaced, the training metadata (i.e training iterations) of the better performing agent is copied over together with the training hyperparameters and network weights. The training hyperparameter of the duplicated agent is then perturbed to explore the training hyperparameter space.

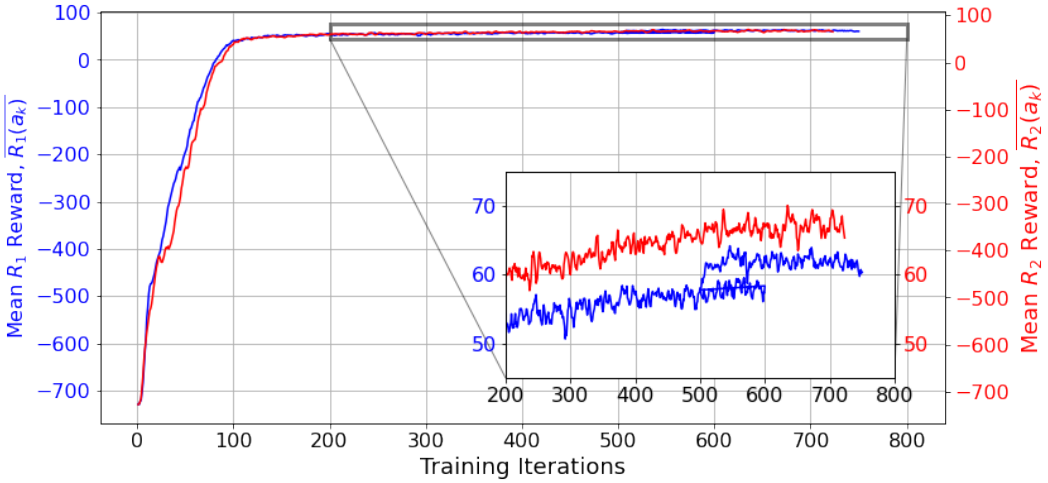


Fig. 7: Training statistics for the two different reward functions

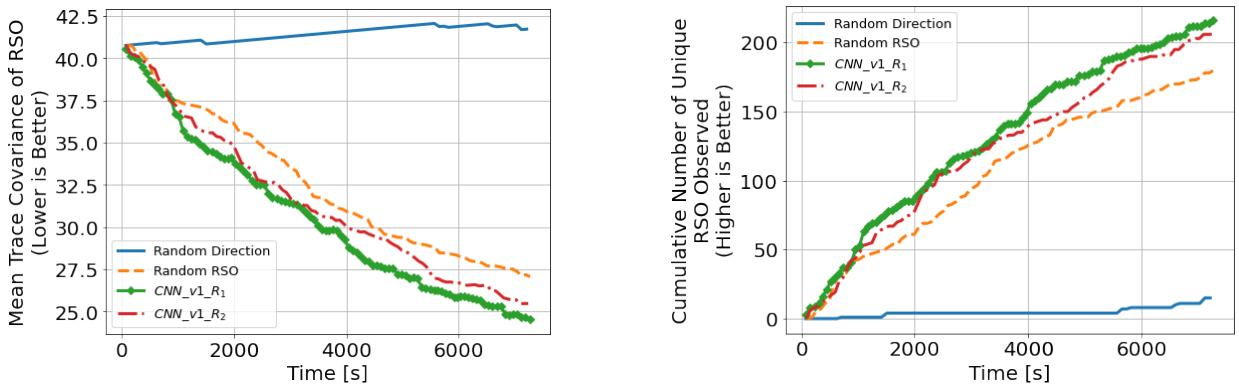
6. RESULTS AND DISCUSSION

Two different random agents are created as a baseline to evaluate the performance of the trained DRL agent. The first random agent (*Random Direction*) randomly selects a pointing direction to observe, whereas the second random agent (*Random RSO*) randomly selects an RSO within the current field of regard to observe. The previously described trained agents and the random agents were evaluated in a single-seeded environment with 400 randomly generated cislunar RSOs. Figs. 8a and 8b show the evolution of the mean trace covariance and the cumulative number of unique RSOs observed over a two-hour observation window, respectively. The trained DRL agents outperformed the *Random Direction* agent on both performance metrics—final mean RSOs uncertainties (mean trace covariance) and the cumulative number of unique RSOs observed over the two-hour observation window.

Due to the large observation and action space and relatively small number of RSOs, the *Random Direction* agent ended up picking an empty field of view to observe most of the time. The RSOs are initialized with large initial uncertainties and the mean uncertainties of all RSOs gradually drop as the RSOs are observed by the *Random RSO* agent. However, the *Random RSO* agent frequently makes subpar decisions, such as re-observing previously observed RSOs or making large slewing motions to observe RSOs that are located far away from its current pointing direction as indicated by the lower cumulative number of unique RSO observed over the observation window. The DRL agents were able to learn action policies that filter out unoccupied viewing directions and select "beneficial" RSOs to be observed in order to maintain low overall mean RSO uncertainties. The DRL agents that were trained using different reward functions have a slight difference in performance. The DRL agent trained on reward function 1 (R_1) has a slightly higher performance. It is hypothesized that the simpler reward function 1 allows the DRL agent to learn faster; do note, however, that the DRL agents have not fully converged as shown in Fig. 7. With more training iterations, the performance of the DRL agents is expected to further improve.

Fig. 9 shows the pointing direction selected by the $CNN\_v1\_R_1$ DRL agent. The DRL agent minimizes action slew time by making minimal changes in the pointing directions between subsequent time steps. Most of the azimuth pointing directions are centered around 35° which corresponds to the azimuth location of the Moon.

100 Monte Carlo runs were then carried out to better compare the aggregate performance of the DRL agents and



(a) Evolution of mean trace covariance

(b) Total number of Unique RSOs visited

Fig. 8: Comparison of results for agents trained with various reward functions and the random agents. Note: $CNN\_v1\_R_1$ represents a `CNN_v1` DRL agent trained using reward function R_1

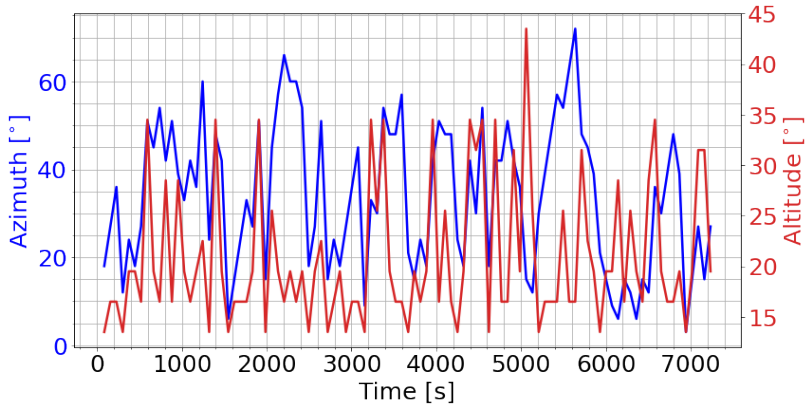
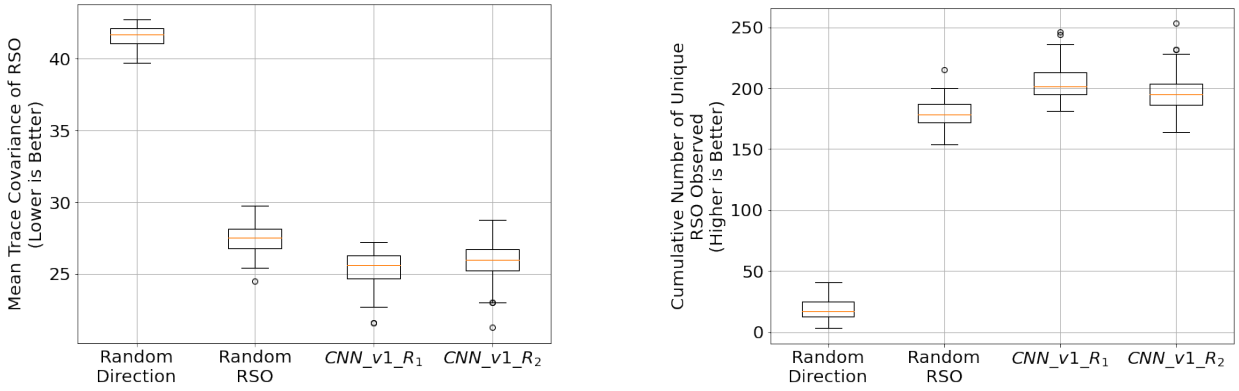


Fig. 9: Pointing directions chosen by the DRL agent $CNN\_v1\_R_1$ in the seeded environment

the random policies. The final mean trace covariance and the cumulative number of unique RSOs observed over the two-hour observation window are shown in Figs. 10a and 10b, respectively.



(a) Evolution of mean trace covariance

(b) Total number of Unique RSOs visited

Fig. 10: Comparison of results for agents trained with various reward functions and the random agents. Note: $CNN\_v1\_R_1$ represents a `CNN_v1` DRL agent trained using reward function R_1

The Monte Carlo results agree well with our observation from the single evaluation run, where the DRL agents outperformed the random policies on both performance criteria. The $CNN\_v1\_R_1$ DRL agent performed the best followed by the $CNN\_v1\_R_2$ DRL agent.

7. CONCLUSION

Space situational awareness (SSA) for the cislunar regime is becoming increasingly important with the increasing space activities in the cislunar regime. Currently, there is a lack of SSA information for the cislunar regime and this poses collision risks to our space assets that are operating in this regime. In this paper, we show that the sensor tasking problem for cislunar SSA can be solved using a deep reinforcement learning (DRL) approach. 100 Monte Carlo runs were carried out the assess the performance of the trained DRL agent and the DRL approach is able to outperform the random policies (random pointing direction and random RSO) in terms of both performance criteria—final mean trace covariance and the cumulative number of unique RSOs observed over the two-hour observation window. The DRL agents have not fully converged after 700 training iterations and their performance is expected to further improve with additional training.

In future works, the author team will be exploring the usage of recurrent neural networks (RNN) such as long short-term memory (LSTM) for this problem, as it is highly time dependent. We also plan to extend the current cislunar SSA environment to support multiple ground-based and space-based sensors working cooperatively to achieve SSA for cislunar space over multiple nights.The author team also plans to explore higher-fidelity sensor models that can be used to differentiate between current optical and radar sensors for cislunar SSA.

ACKNOWLEDGEMENT

This research was sponsored by the United States Air Force Research Laboratory and the United States Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the United States Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein. The authors would also like to acknowledge the support of this work by the Air Force's Office of Scientific Research under Contract FA9550-18-1-0115 and thank the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC, database, and consultation resources that have contributed to the research results reported in this paper. Daniel Jang is supported by the MIT Lincoln Laboratory Fellowship and Thomas G. Roberts is supported by the National Science Foundation Graduate Research Fellowship under Grant No. 1745302.

REFERENCES

[1] R. Linares and R. Furfaro. Dynamic Sensor Tasking for Space Situational Awareness via Reinforcement Learning. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, HI, 2016.

[2] R. Linares and R. Furfaro. An Autonomous Sensor Tasking Approach for Large Scale Space Object Cataloging. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, HI, 2017.

[3] D. Jang, P.M. Siew, D. Gondelach, and R. Linares. Space Situational Awareness Tasking For Narrow Field Of View Sensors: A Deep Reinforcement Learning Approach. In *71st International Astronautical Congress*. International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law, 2020.

[4] P.M. Siew, D. Jang, and R. Linares. Sensor Tasking for Space Situational Awareness Using Deep Reinforcement Learning. In *AIAA/AAS Astrodynamics Specialist Conference*, Big Sky, MT, 2021.

[5] T.G. Roberts, P.M. Siew, D. Jang, and R. Linares. A Deep Reinforcement Learning Application to Space-Based Sensor Tasking for Space Situational Awareness. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, 2021.

[6] S. Fedeler, M. Holzinger, and W. Whitacre. Sensor Tasking in the Cislunar Regime using Monte Carlo Tree Search. *Advances in Space Research*, 70(3):792–811, 2022.

[7] E.E. Fowler, S.B. Hurtt, and D.A. Paley. Observability Metrics for Space-Based Cislunar Domain Awareness, 2020.
[8] A. Wilmer, R.A. Bettinger, and B. Little. Cislunar Periodic Orbit Constellation Assessment for Space Domain Awareness of L1 and L2 Halo Orbits. *ASCEND 2021*, 2021.
[9] M. Holzinger, C.C. Chow, and P. Garretson. A Primer on Cislunar Space, 2021.
[10] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods and Applications*. Springer Berlin, 2013.
[11] D.A. Vallado. *Fund. of Astrodynamics and Applications*. Microcosm, Hawthrone, California, 4th edition, 2013.
[12] V. Szebehely. *Theory of Orbits: The Restricted Problem of Three Bodies*. New York, 1967.
[13] J.L. Lagrange. Essai sur le probleme des trois corps. *Œuvres*, 6:229–324, 1772.
[14] C. Maccone. Lunar Farside Radio Lab : A "Cosmic Study" by the International Academy of Astronautics. *54th International Astronautical Congress of the International Astronautical Federation, the International Academy of Astronautics, and the International Institute of Space Law*, 2003.
[15] K. Bocam, C. Walz, D. Bodkin, C. Pappageorge, V. Hutchinson, M. Dennis, and W. Cugn. A Blueprint for Cislunar Exploration: A Cost-Effective Building Block Approach for Human Lunar Return. In *AIAA Space 2012*, Pasadena, CA, 2012.
[16] W. Pratt, C. Buxton, S. Hall, J. Hopkins, and A. Scott. Trajectory Design Considerations for Human Missions to Explore the Lunar Farside from the Earth-Moon Lagrange Point EM-L2. In *AIAA Space 2013*, San Diego, CA, 2013.
[17] K. Johnson. Fly Me to the Moon: Worldwide Cislunar and Lunar Missions. *Center for Strategic and International Studies*, 2022.
[18] R. Whitley and R. Martinez. Options for Staging Orbits in Cislunar Space. *2016 IEEE Aerospace Conference*, 2016.
[19] C. Ocampo and G. Rosborough. Transfer Trajectories for Distant Retrograde Orbiters of the Earth. *Advances in the Astronautical Sciences*, 82(2):1177–1200, 1993.
[20] NASA JPL. JPL Three-Body Periodic Orbit Catalog, 2022.
[21] M.R. Thompson. Cislunar Orbit Determination and Tracking via Simulated Space-Based Measurements. In *Advanced Maui Optical and Space Surveillance Technologies Conference*, Maui, HI, 2021.
[22] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zarema. OpenAI Gym, 2016.
[23] Nicholas Kaiser. Pan-STARRS: a Wide-Field Optical Survey Telescope Array. In Jacobus M. Oschmann Jr., editor, *Ground-based Telescopes*, volume 5489, pages 11 – 22. International Society for Optics and Photonics, SPIE, 2004.
[24] K. W. Hodapp, N. Kaiser, H. Aussel, W. Burgett, K. C. Chambers, M. Chun, T. Dombeck, A. Douglas, D. Hafner, J. Heasley, J. Hoblitt, C. Hude, S. Isani, R. Jedicke, D. Jewitt, U. Laux, G. A. Luppino, R. Lupton, M. Maberry, E. Magnier, E. Mannery, D. Monet, J. Morgan, P. Onaka, P. Price, A. Ryan, W. Siegmund, I. Szapudi, J. Tonry, R. Wainscoat, and M. Waterson. Design of the Pan-STARRS Telescopes. *Astronomische Nachrichten*, 325(6-8):636–642, 2004.
[25] L. Denneau, R. Jedicke, T. Grav, M. Granvik, J. Kubica, A. Milani, P. Vereš, R. Wainscoat, D. Chang, F. Pierfederici, N. Kaiser, K.C. Chambers, J.N. Heasley, E.A. Magnier, P.A. Price, J. Myers, J. Kleyna, H. Hsieh, D. Farnocchia, C. Waters, W.H. Sweeney, D. Green, B. Bolin, W.S. Burgett, J.S. Morgan, J.L. Tonry, K.W. Hodapp, S. Chastel, S. Chesley, A. Fitzsimmons, M. Holman, T. Spahr, D. Tholen, G.V. Williams, S. Abe, J.D. Armstrong, T.H. Bressi, R. Holmes, T. Lister, R.S. McMillan, M. Micheli, E.V. Ryan, W.H. Ryan, and J.V. Scotti. The Pan-STARRS Moving Object Processing System. *Publications of the Astronomical Society of the Pacific*, 125(926):357–395, apr 2013.
[26] D. Monet. Space Situational Awareness Applications of the PS1 AP Catalog. In *The Advanced Maui Optical and Space Surveillance Technologies Conference*, page E40, January 2006.
[27] M. Jaderberg, V. Dalibard, S. Osindero, W.M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, C. Fernando, and K. Kavukcuoglu. Population Based Training of Neural Networks, 2017.
[28] P. Moritz, R. Nishihara, S. Wang, A. Tumanov, R. Liaw, E. Liang, M. Elibol, Z. Yang, W. Paul, M.I. Jordan, and I. Stoica. Ray: A Distributed Framework for Emerging AI Applications. In *Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI '18)*, Carlsbad, CA, 10 2018. USENIX.
[29] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal Policy Optimization Algorithms, 2017.
[30] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Lev-

enberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, M. Zheng, X.Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, M. Zheng, X.Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015.

[31] A. Reuther, J. Kepner, C. Byun, S. Samsi, W. Arcand, D. Bestor, B. Bergeron, V. Gadepally, M. Houle, M. Hubbell, M. Jones, A. Klein, L. Milechin, J. Mullen, A. Prout, A. Rosa, C. Yee, and P. Michaleas. Interactive Supercomputing on 40,000 Cores for Machine Learning and Data Analysis. In *Proceedings of the IEEE High Performance Extreme Computing Conference (HPEC)*, Waltham, MA, 7 2018. Institute of Electrical and Electronics Engineers.