

# IMPROVING CLOUD OBSERVATIONS BY AUTONOMOUSLY POINTING SATELLITES

Mary Dahl 

Massachusetts Institute of Technology  
Cambridge, MA, 02139, USA  
marydahl@mit.edu

Kerri Cahoy 

Massachusetts Institute of Technology  
Cambridge, MA, 02139, USA  
kcahoy@mit.edu

**Abstract**—One of the largest sources of uncertainty in climate research is aerosol-cloud interaction. Our capabilities to increase our understanding are limited by the quantity and quality of cloud measurements we can make. Traditional, nadir-pointing satellites waste time and power by measuring clear skies while potentially missing high-value cloud measurements just off-nadir. With the conclusion of the CloudSat and CALIPSO missions, there is a need for new satellites to monitor clouds. New satellites have an opportunity to take advantage of advanced remote sensing techniques such as autonomy to improve the yield of high-value cloud measurements. In this paper, we present a proof of concept for an autonomously pointing satellite that can dynamically target off-nadir clouds. We develop a simulation environment that evaluates the capabilities of different algorithms and satellites to complete this task while managing power and memory storage. We build this simulation with MODIS cloud masks, which contain data for cloud cover percentages, existence of cirrus clouds, and light/eclipse status. We analyze data from January 2010, when the CloudSat mission was in full operation. We compare the number of clouds measured by algorithms we developed (binary toggle, greedy, nearest greedy, and distance weighted) to CloudSat’s performance during the same period. Our best performing algorithm measures, on average, 60 percent of cloud cells with low variance, compared with CloudSat’s less than 40 percent with high variance. Dynamic targeting satellites can substantially enhance cloud monitoring and improve our ability to understand aerosol-cloud interactions for climate research.

**Index Terms**—dynamic targeting, clouds, aerosols, climate, autonomy, satellite, remote sensing

## I. INTRODUCTION **General background**

According to the 2013 report of the Intergovernmental Panel on Climate Change (IPCC), aerosol-cloud interaction is a major challenge in climate modeling [1]. In their 2021 report, they reaffirm clouds as “the largest contribution to overall uncertainty in climate feedbacks” [2].

There have been many satellites focused on cloud observation, such as NASA’s CloudSat and CALIPSO [3, 4], which ended operations in 2023, ESA-JAXA’s EarthCARE [5–7], and NASA’s PACE [8, 9]. Continued innovation in future missions will be necessary to address the needs of climate models. From 2018-2021, NASA held an Aerosol, Clouds, Convection and Precipitation Study, where they expressed a desire to improve measurements by using AI “for targeted observations: many

The authors would like to thank MathW for a fellowship. **Specific background**

of our cloud satellites spend a lot of their time observing clear air [10].”

A targeted observation system has been proven on JAXA’s GOSAT-2, a satellite that monitors greenhouse gasses. It uses a camera to find cloud-free areas in its field of view, then uses a pointing mirror to measure them [11–13]. While useful to get higher quality measurements, this system lacks the ability to plan for future observations. JPL’s upcoming SMICES small satellite seeks to use a radiometer to plan targeted radar measurements of ice storms [14, 15]. They have analyzed systems for cloud avoidance [16, 17] and storm tracking [18] with a variety of algorithms [19]. These studies did not include explicit simulation of satellite systems, instead using proxies [20]. Other work has been done looking at long-term planning for scheduling observations [21] and on analyzing the capabilities of a body-fixed “lookahead” sensor [22].

There has not yet been a general simulation that demonstrates and analyzes the activities of a targeted observation satellite. This is important for ensuring feasibility of a real life system, particularly in power usage. Having a general simulation also makes it easier to compare multiple similar mission configurations. **Knowledge gap**

In this work, we develop a simulation to explore mutually-jective prioritization for cloud measurements on a modeled satellite system. We compare the historical performance of CloudSat to a simulated satellite that plans its activities using a prioritization algorithm: binary toggle, greedy, a variant on greedy, or distance weighted. We also analyze different sizes of planning windows for these algorithms to plan on. We analyze the efficiency of each of these algorithms at measuring clouds given their action choices, power utilization, and storage management. **Here we show...**

## II. METHODOLOGY

### A. Simulation **Materials**

A high-level look at the architecture of the simulation is shown in Figure 1. It is designed similarly to a reinforcement learning framework, with an agent taking an action, receiving updated states, taking another action based on that, and so on.

The Earth module of the environment contains the state of the physical world. It is built from 1 km resolution cloud masks derived from the MODIS instrument on the Aqua satellite [23]. This works especially well for simulating CloudSat

Copyright 2024 IEEE. Published in the 2025 IEEE International Geoscience and Remote Sensing Symposium (IGARSS 2025), scheduled for 3 - 8 August 2025 in Brisbane, Australia. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works, must be obtained from the IEEE. Contact: Manager, Copyrights and Permissions / IEEE Service Center / 445 Hoes Lane / P.O. Box 1331 / Piscataway, NJ 08855-1331, USA. Telephone: + Intl. 908-562-3966.

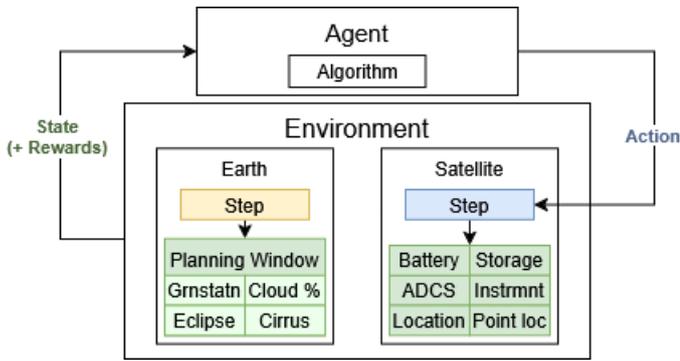


Fig. 1. The simulation software diagram. An agent, controlled by an algorithm, decides which action the satellite should take based on its and the Earth's state. The satellite performs that action, and both states are updated. This new state is sent back to the agent, and the cycle continues.

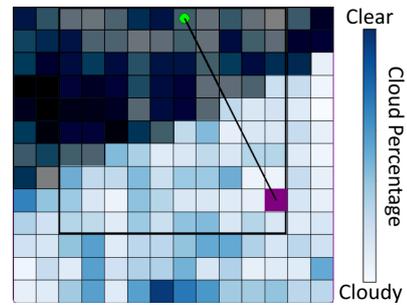


Fig. 2. A render of the simulation in Pygame. The green circle is the satellite. The semi-transparent box is the planning window. The purple square is the pointing location. The grid cells are colored by their cloud percentage: blue means empty, while white means cloudy. The darker cells indicate eclipse.

performance, as it was in the same A-Train orbit as Aqua from 2006 to 2018. By using measurements from MODIS, we have a representative depiction of what CloudSat could have measured in that time period.

To make the Earth environment, we break the cloud mask into cells the size of the footprint of CloudSat's radar. Each cell contains four pieces of information: the percentage of cloud cover, if it contains a cirrus cloud, if it contains a ground station (we use the NASA Near Earth Network), and if it is in light or eclipse. This data is communicated to the satellite via a planning window: a square of cells ahead of the satellite it has full knowledge of. At each step of the environment, the satellite is moved forward by one cell and the planning window is updated by the next row of cells.

The satellite simulation contains parameters for the satellite. It has states for its instrument, its battery, its ADCS system, its onboard storage, and its communication system. At each time step in the simulation, the agent can have the satellite perform one of four actions: toggle the instrument on or off, downlink, slew to target, or do nothing.

When the satellite is given an action, it calculates the power and memory storage loss for that action and, if both are sufficient, it performs the action and updates its state. If the instrument is on and the satellite is not actively pointing, then the cell it is pointing to is measured. Slewing time and power usage is calculated by using a bang-off-bang controller. If one time step is shorter than the time to slew to the desired target, the satellite cannot take any more actions until it is finished slewing. If the satellite ever runs out of battery or storage, it uses its next action to automatically shut the instrument down.

The simulation can render the environment to help visualize the actions of the satellite. This render is done using the Pygame engine (pygame.org), and is shown in Figure 2.

### B. Limitations

#### Limitations

There are several limitations to the simulation in its current state. First, the timing and the resources necessary to acquire new planning windows must be accounted for. These windows

would be subsets of larger lookahead images. In a real system, a new lookahead image must be acquired, processed, and planned upon before the satellite moves past the end of the previous one. A lookahead image can be obtained either by periodically pointing the satellite forward or by using a second satellite to capture and send it. The image would be processed by converting it to binary cloud masks, identifying cloud types of interest, and dividing it into planning windows. On-orbit cloud segmentation is well studied [24–27], and, timewise, scales linearly with the number of pixels. To make power and time estimates for onboard performance, these algorithms can be broken into their basic computational structures [28] and the performance of those functions can be benchmarked on space processors [29]. This same process can be used to assess the power requirements of the planning algorithms. With current estimations, we have determined a lookahead image with a 100 km length can meet movement requirements.

Performance of instruments when used off-nadir is also not currently captured. Active instruments that rely on distance to targets – lidars, radars – can have increased error the further off-nadir they point. Errors in beam location [30] and Doppler shift [31] from pointing errors are exacerbated at large angles. However, these errors are seen in applications that use concentrated beam returns, like altimetry off of the ocean. Because clouds give diffuse lidar and radar responses, we expect these errors to be limited. Nevertheless, we plan to integrate pointing limitations for active instruments.

In the future, we plan to simulate optical instruments. Many optical instruments – radiometers, polarimeters – only work in the light, and also require measurements to be taken at certain solar zenith angles. This is both to ensure compatibility with look up tables and to avoid unfavorable solar zenith angles, as high angles affect retrieval of many important cloud properties [32]. Limitations on pointing angles will be included.

### C. Base Case

#### Experimental set up

Because our goal is to improve the performance on cloud satellites of the past, we use CloudSat as a reference case. We analyze data from January 2010, before CloudSat moved out of the A-Train orbit. CloudSat is represented with a baseline algorithm that toggles its instrument on a fixed duty cycle.

## D. Algorithms

## Standard methods + deviations

We analyze four different algorithms for this work: binary toggle, greedy, nearest greedy, and a distance weighted method, inspired by A\*. All algorithms that can slew to a target follow a common structure, only varying the method by which the target is chosen. They prioritize, in this order:

- 1) Turning off the instrument if the battery is  $\leq 30\%$
- 2) Turning on the instrument if the battery is  $> 30\%$
- 3) Slewing to a ground station and downlinking if the buffer is  $80\%$
- 4) Picking a cloud cell to point to

Each algorithm has the same scoring system to rank cells:

$$CloudScore = w_1 \frac{ClearPixels}{TotalPixels} + (1 - w_1) * Cirrus \quad (1)$$

Where *Cirrus* is a binary value of the presence of a cirrus cloud (0 meaning present, 1 meaning not present) and  $w_1$  is adjusted to change the priority of choosing a cell with high cloud cover and a cell with a cirrus cloud. The best *CloudScore* has a value of 0, meaning complete cover and a cirrus, and the worst has a value of 1. All algorithms are also constrained to not measure the same cell multiple times.

1) *Binary Toggle*: The satellite checks the next cell it would measure. If the cell has a *CloudScore* below 0.5, it turns the instrument on. If not, it turns it off. After the satellite downlinks, it returns to point nadir. This algorithm prioritizes battery life above all.

2) *Greedy*: The satellite tries to point at the cell with the best *CloudScore*, regardless of how long it will take to slew. While likely to score the highest amount of cloud measurements, it is likely to waste power and time resources by slewing frequently.

3) *Nearest Greedy*: This variant looks to restrict the amount of slewing. The satellite looks at the cells with the  $n$  most valuable *CloudScores*. It then targets the one that is closest to the current pointing position. Tuning the value for  $n$  is problematic: if  $n$  is too small, the algorithm misses nearby cells that score slightly worse, and if  $n$  is too big, it acts too similarly to greedy. We set  $n$  to the number of cells divided by 100, but restrict it to at least 4 and at most 50.

4) *Distance Weighted*: Each cell is scored with both its *CloudScore* and by its distance,  $D$ , to the current pointing location by the equation:

$$DistanceScore = w_2 CloudScore + (1 - w_2)D \quad (2)$$

Where  $w_2$  can be adjusted to favor nearby cells or *CloudScore*. The satellite then points to the cell with the best *DistanceScore*. We used a  $w_2$  of 0.4. This algorithm is the most computationally expensive of the four options.

We considered other path planning algorithms, such as RRT or reinforcement learning, but determined they added more complexity than was necessary to solve the planning problem.

## E. Assessment Criteria

The algorithm variations are summarized in Table I. We analyze different sizes of planning windows to evaluate how

more options affects the decisions the algorithms make. We also vary weighing cloud percentage and cirrus presence equally ( $w_1 = 0.5$ ) and favoring cirrus clouds ( $w_1 = 0.05$ ).

TABLE I  
THE ALGORITHM VARIATIONS THAT WERE TESTED FOR THIS STUDY.

Algorithm	Planning Window (km)	Priority
Baseline (CloudSat rep)	x	Equal, cirrus
Binary Toggle	x	Equal, cirrus
Greedy	10, 14, 16, 20, 100	Equal, cirrus
Nearest Greedy	10, 14, 16, 20, 100	Equal, cirrus
Distance Weighted	10, 14, 16, 20, 100	Equal, cirrus

Each algorithm variation is simulated over a week at a time. We assess the performance on the efficiency and the quality of its measurements. For *efficiency*, we look at what the algorithm chose to do. We assess this on both the frequency of action choices and on the total power expenditure. For *quality*, we look at the number of cells measured in three categories: any amount of cloud cover, over 50% cloud cover, and if they contain a cirrus cloud. These are compared to a maximum, calculated by choosing the best cell at each time step, ignoring pointing, power, and planning window size limitations.

We considered using the amount of clouds downlinked as a metric, but we found that each algorithm was able to downlink all cells that were measured, so it is not presented here.

## III. RESULTS

## A. Actions

Figure 3 shows how often a subset of the algorithms chose each action from January 22-28 (other weeks were similar). These match our instincts – the greedy and nearest greedy algorithms point the most, while the distance weighted algorithms are more conservative. Binary toggle downlinks the least as it fills its buffer slowest. When the algorithms have larger planning windows, they spend slightly more time pointing and much less time toggling their instrument, as they less frequently are in situations with no good measurements.

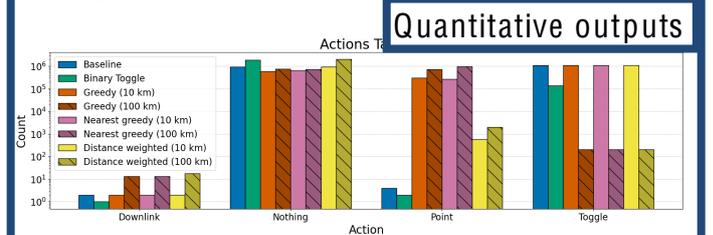


Fig. 3. The frequency of actions taken by each algorithm for the smallest and largest planning windows. The majority of time is spent doing nothing (typically making measurements), pointing (if capable), or toggling. Note the log scale on the y axis.

## B. Battery

Most algorithms performed similarly, utilizing approximately 300 Wh over one eclipse period (about 45 minutes) and never fully depleting. The exception was binary toggle,

## Quantitative outputs

which used only 200 Wh. Greedy with the largest planning window was the second most efficient (270 Wh), followed by the nearest greedy with the largest planning window (278 Wh). This is due to how slewing to point works. While the satellite is pointing, it automatically turns the instrument off. Because greedy algorithms take the “point” action the most, they ultimately save battery by having their instruments off most frequently. Toggling frequently may have adverse affects on the long-term health of a real inst

**Factual conclusion**

method versus going into a low power state is future work.

### C. Cloud Measurements

## Quantitative outputs

Figure 4 shows the percentage of the maximum cloud measurement each algorithm achieved. This plot shows the average performance of one week over the month. The baseline measures, on average, 40% of cloudy cells, with large variance for every metric. Its performance is solely decided by how cloudy the center of the swath is. It has similar performance to all 10 km slewing algorithms. Binary toggle appears to be the worst performing algorithm. However, it has the highest proportion of cells with 50% of clouds and cirrus cells of any algorithm. It also has very low variance, as, regardless of how cloudy or not in the center of the swath, it will always act the same way.

On average, the best performing slewing algorithm at all planning window sizes is nearest greedy. The 100 km window measures over 60% of the maximum number of cloudy cells with low variance. While performance improves for all slewing algorithms as the planning window size increases, both in average number of measurements and in variance, there are diminishing returns. This corresponds to action choice. Larger windows mean there are more valuable cells available at each time step, but to reach them, more time is spent pointing instead of measuring.

## Factual conclusion

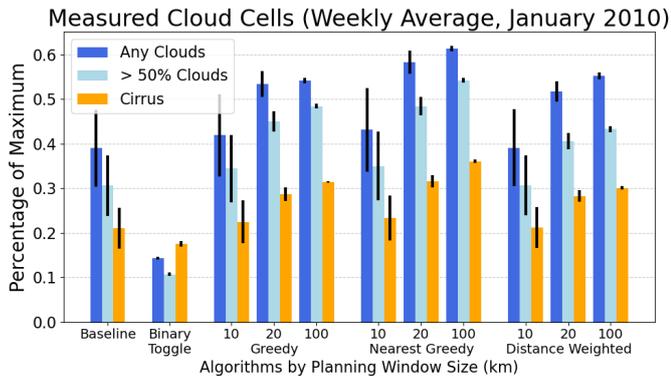


Fig. 4. The results of each algorithm variant. Larger planning windows increase the quantity of measurements considerably, but there are diminishing returns for windows greater than 20 km. 14 and 16 km omitted for simplicity.

We also look specifically at cirrus collection by adjusting the weight  $w_1$  from Equation 1 in Figure 5. Binary toggle sees significant improvement in the cirrus collection. There is a small improvement for distance weighted, as, because it

spends less time pointing than the other slewing algorithms, it has more time to measure more cells overall.

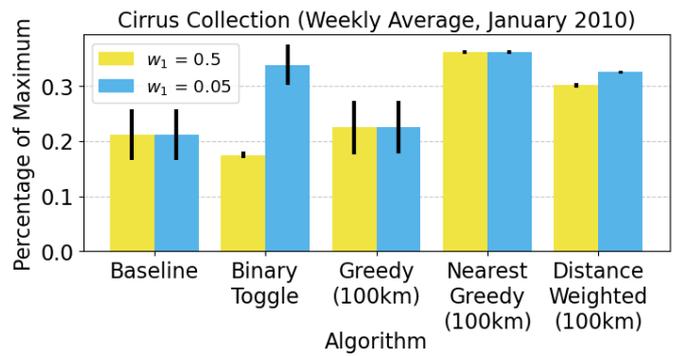


Fig. 5. The difference in cirrus collection for each algorithm under different values for  $w_1$  in Equation 1 (lower weights mean cirrus priority). Binary toggle and distance weighted both show improvement.

## IV. DISCUSSION

## Interpretation

There are several insights from the different algorithm variants. First, the runs with only a 10 km planning window barely outperformed the baseline. As the planning window increases in size, performance also increases. This rate slows at around 20 km, indicating anything larger than that is unnecessary. Nearest greedy was the best performing algorithm. Its limitation on pointing caused it to spend less time than greedy pointing at slightly more optimal cells that were far away. While binary toggle measures the least clouds overall, its power savings and high proportion of good measurements to total measurements makes it an excellent option for low budget missions that cannot afford slewing or a complex lookahead system where quality is more important than quantity. The distance weighted algorithm was successful at limiting point actions, but it came at the cost of performance.

## Future Work

In future work, we will make the simulation more accurate by improving on the previously mentioned limitations. For lookahead image capturing, we will simulate different architectures. For instance, if there is only one satellite, the distortions in the image from body pointing may affect masking and therefore cell selection. If there are two satellites, computation responsibilities could be distributed between them, but crosslinks become important to analyze. We will integrate instrumentation slewing requirements. We plan to simulate radars, lidars, polarimeters, and radiometers, and will analyze their relative viability in a dynamic environment.

This work provides strong baseline for future mission design using an autonomous targeting and planning approach. Our methods consistently provide a higher quantity of high quality measurements than CloudSat. With the development of this simulation, we can continue to assess a multitude of instruments and mission architectures to best quantify options for real missions.

## Final remarks and implications

## REFERENCES

- [1] T. Stocker, D. Qin, G.-K. Plattner, M. Tignor, S. Allen, J. Boschung, A. Nauels, Y. Xia, V. Bex, and P. e. Midgley, Eds., *Climate Change 2013: The Physical Science Basis. Contribution of Working Group I to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2013.
- [2] V. Masson-Delmotte, P. Zhai, A. Pirani, S. Connors, C. Péan, S. Berger, N. Caud, Y. Chen, L. Goldfarb, M. Gomis, M. Huang, K. Leitzell, E. Lonnoy, J. Matthews, T. Maycock, T. Waterfield, O. Yelekçi, R. Yu, and B. e. Zhou, Eds., *Climate Change 2021: The Physical Science Basis. Contribution of Working Group I to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge, United Kingdom and New York, NY, USA: Cambridge University Press, 2021.
- [3] G. L. Stephens, D. G. Vane, S. Tanelli, E. Im, S. Durden, M. Rokey, D. Reinke, P. Partain, G. G. Mace, R. Austin, T. L'Ecuyer, J. Haynes, M. Lebsock, K. Suzuki, D. Waliser, D. Wu, J. Kay, A. Gettelman, Z. Wang, and R. Marchand, "Cloudsat mission: Performance and early science after the first year of operation," *Journal of Geophysical Research: Atmospheres*, vol. 113, no. D8, 2008. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2008JD009982>
- [4] S. Tanelli, S. L. Durden, E. Im, K. S. Pak, D. G. Reinke, P. Partain, J. M. Haynes, and R. T. Marchand, "Cloudsat's cloud profiling radar after two years in orbit: Performance, calibration, and processing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3560–3573, 2008.
- [5] H. W. Barker, M. P. Jerg, T. Wehr, S. Kato, D. P. Donovan, and R. J. Hogan, "A 3d cloud-construction algorithm for the earthcare satellite mission," *Quarterly Journal of the Royal Meteorological Society*, vol. 137, no. 657, pp. 1042–1058, 2011. [Online]. Available: <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.824>
- [6] A. J. Illingworth, H. W. Barker, A. Beljaars, M. Ceccaldi, H. Chepfer, N. Clerbaux, J. Cole, J. Delanoë, C. Domenech, D. P. Donovan, S. Fukuda, M. Hiraoka, R. J. Hogan, A. Huenerbein, P. Kollias, T. Kubota, T. Nakajima, T. Y. Nakajima, T. Nishizawa, Y. Ohno, H. Okamoto, R. Oki, K. Sato, M. Satoh, M. W. Shephard, A. Velázquez-Blázquez, U. Wandinger, T. Wehr, and G.-J. van Zadelhoff, "The earthcare satellite: The next step forward in global measurements of clouds, aerosols, precipitation, and radiation," *Bulletin of the American Meteorological Society*, vol. 96, no. 8, pp. 1311 – 1332, 2015. [Online]. Available: <https://journals.ametsoc.org/view/journals/bams/96/8/bams-d-12-00227.1.xml>
- [7] T. Wehr, T. Kubota, G. Tzeremes, K. Wallace, H. Nakatsuka, Y. Ohno, R. Koopman, S. Rusli, M. Kikuchi, M. Eisinger, T. Tanaka, M. Taga, P. Deghaye, E. Tomita, and D. Bernaerts, "The earthcare mission – science and system overview," *Atmospheric Measurement Techniques*, vol. 16, no. 15, pp. 3581–3608, 2023. [Online]. Available: <https://amt.copernicus.org/articles/16/3581/2023/>
- [8] E. T. Gorman, D. A. Kubalak, D. Patel, A. Dress, D. B. Mott, G. Meister, and P. J. Werdell, "The NASA Plankton, Aerosol, Cloud, ocean Ecosystem (PACE) mission: an emerging era of global, hyperspectral Earth system remote sensing," in *Sensors, Systems, and Next-Generation Satellites XXIII*, S. P. Neek, P. Martimort, and T. Kimura, Eds., vol. 11151, International Society for Optics and Photonics. SPIE, 2019, p. 111510G. [Online]. Available: <https://doi.org/10.1117/12.2537146>
- [9] I. Cetinic, C. R. McClain, P. J. Werdell, S. Platnick, O. Coddington, S. A. Ackerman, R. Frey, A. Heidinger, A. Walter, K. G. Meyer *et al.*, "Pace technical report series, volume 4: Cloud retrievals in the pace mission: Pace science team consensus document," Goddard Space Flight Center, Tech. Rep., 2018.
- [10] A. Gettelman, G. R. Carmichael, G. Feingold, A. M. Da Silva, and S. C. Van den Heever, "Confronting future models with future satellite observations of clouds and aerosols," *Bulletin of the American Meteorological Society*, vol. 102, pp. E1557–E1562, 2021.
- [11] H. Suto, F. Kataoka, N. Kikuchi, R. O. Knuteson, A. Butz, M. Haun, H. Buijs, K. Shiomi, H. Imai, and A. Kuze, "Thermal and near-infrared sensor for carbon observation Fourier transform spectrometer-2 (TANSO-FTS-2) on the Greenhouse gases Observing SATellite-2 (GOSAT-2) during its first year in orbit," *Atmospheric Measurement Techniques*, vol. 14, no. 3, pp. 2013–2039, 2021. [Online]. Available: <https://amt.copernicus.org/articles/14/2013/2021/>
- [12] R. Glumb, G. Davis, and C. Lietzke, "The tanso-fts-2 instrument for the gosat-2 greenhouse gas monitoring mission," in *2014 IEEE Geoscience and Remote Sensing Symposium*, 2014, pp. 1238–1240.
- [13] R. Imasu, T. Matsunaga, M. Nakajima, Y. Yoshida, K. Shiomi, I. Morino, N. Saitoh, Y. Niwa, Y. Someya, Y. Oishi, M. Hashimoto, H. Noda, K. Hikosaka, O. Uchino, S. Maksyutov, H. Takagi, H. Ishida, T. Y. Nakajima, T. Nakajima, and C. Shi, "Greenhouse gases observing satellite 2 (gosat-2): mission overview," *Progress in Earth and Planetary Science*, vol. 10, no. 1, p. 33, 2023. [Online]. Available: <https://doi.org/10.1186/s40645-023-00562-2>
- [14] J. Swope, S. Chien, X. Bosch-Lluis, Q. Yue, P. Tavallali, M. Ogut, I. Ramos, P. Kangaslahti, W. Deal, and C. Cooke, "Using intelligent targeting to increase the science return of a smart ice storm hunting radar," in *International Workshop on Planning & Scheduling for Space (IWSPSS)*, July 2021. [Online]. Available: <https://ai.jpl.nasa.gov/public/papers/Swope-SMICES-targeting-IWSPSS-2021.pdf>
- [15] M. Ogut, X. Bosch-Lluis, P. Kangaslahti, I. Ramos-Perez, J. F. Munoz-Martín, J. Cooperrider, Q. Yue, J. Swope, P. Tavallali, S. Chien, O. Pradhan, W. Deal, and C. Cooke, "Autonomous capabilities and command and data handling design for the smart remote sensing of cloud ice," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 7119–7122.
- [16] Z. Hasnain, J. Mason, J. Swope, J. V. Hook, and S. A. Chien, "Agile Spacecraft Imaging Algorithm Comparison for Earth Science," in *JPL Artificial Intelligence Group*, 2021. [Online]. Available: <https://api.semanticscholar.org/CorpusID:249618618>
- [17] A. Candela, J. Swope, and S. Chien, "Dynamic targeting for cloud avoidance to improve science of space missions," in *16th Symposium on Advanced Space Technologies in Robotics and Automation*, June 2022. [Online]. Available: <https://ai.jpl.nasa.gov/public/documents/papers/Candela-DT-ASTRA-2022.pdf>
- [18] A. Candela, J. Swope, S. Chien, H. Su, and P. Tavallali, "Dynamic targeting for improved tracking of storm features," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, 2022, pp. 5313–5316.
- [19] A. Kangaslahti, I. Zilberstein, A. Candela, and S. Chien, "Search applications for integrated planning and execution of satellite observations using dynamic targeting," in *International Conference on Automated Planning and Scheduling (ICAPS) Workshop on Planning and Robotics (PlanRob)*, Banff, Canada, June 2024. [Online]. Available: <https://ai.jpl.nasa.gov/public/documents/papers/dt-planrob-2024.pdf>
- [20] A. Kangaslahti, A. Candela, J. Swope, Q. Yue, and S. Chien, "Dynamic targeting of satellite observations incorporating slewing costs and complex observation utility," in *IEEE International Conference on Robotics and Automation (ICRA 2024)*, Yokohama, Japan, May 2024. [Online]. Available: <https://ai.jpl.nasa.gov/public/documents/>

papers/Kangaslahti\_DT\_ICRA\_2024.pdf

- [21] A. Herrmann and H. Schaub, "Reinforcement learning for the agile earth-observing satellite scheduling problem," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 5, pp. 5235–5247, 2023.
- [22] S. Kacker and K. Cahoy, "Reinforcement-Learned Lookahead Heuristics for Earth-Observing Satellites," *Small Satellite Conference*, Aug. 2024. [Online]. Available: <https://digitalcommons.usu.edu/smallsat/2024/all2024/60/>
- [23] S. Platnick, K. G. Meyer, M. D. King, G. Wind, N. Amarasinghe, B. Marchant, G. T. Arnold, Z. Zhang, P. A. Hubanks, R. E. Holz, P. Yang, W. L. Ridgway, and J. Riedi, "The modis cloud optical and microphysical products: Collection 6 updates and examples from terra and aqua," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 502–525, Jan 2017, epub 2016 Oct 26.
- [24] S. Kacker, A. Meredith, K. Cahoy, and G. Labreche, "Machine learning image processing algorithms onboard OPS-SAT," *Small Satellite Conference*, Aug. 2022. [Online]. Available: <https://digitalcommons.usu.edu/smallsat/2022/all2022/65/>
- [25] S. Kacker, A. Meredith, J. Kusters, H. Tomio, V. Felt, and K. Cahoy, "On-orbit rule-based and deep learning image segmentation strategies," in *AIAA SCITECH 2022 Forum*, 2022, p. 0646.
- [26] J. H. Jeppesen, R. H. Jacobsen, F. Inceoglu, and T. S. Toftegaard, "A cloud detection algorithm for satellite imagery based on deep learning," *Remote sensing of environment*, vol. 229, pp. 247–259, 2019.
- [27] M. Segal-Rozenhaimer, A. Li, K. Das, and V. Chirayath, "Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (cnn)," *Remote Sensing of Environment*, vol. 237, p. 111446, 2020.
- [28] K. Asanović, R. Bodik, B. C. Catanzaro, J. J. Gebis, P. Husbands, K. Keutzer, D. A. Patterson, W. L. Plishker, J. Shalf, S. W. Williams, and K. A. Yelick, "The landscape of parallel computing research: A view from berkeley," Tech. Rep. UCB/EECS-2006-183, Dec 2006. [Online]. Available: <http://www2.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.html>
- [29] T. M. Lovelley and A. D. George, "Comparative analysis of present and future space-grade processors with device metrics," *Journal of Aerospace Information Systems*, vol. 14, no. 3, pp. 184–197, 2017. [Online]. Available: <https://doi.org/10.2514/1.I010472>
- [30] C. Parsons and E. Walsh, "Off-nadir radar altimetry," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 27, no. 2, pp. 215–224, 1989.
- [31] H. Kumagai, H. Kuroiwa, S. Kobayashi, and T. Orikasa, "Cloud profiling radar for EarthCARE mission," in *Microwave Remote Sensing of the Atmosphere and Environment III*, C. D. Kummerow, J. Jiang, and S. Uratuka, Eds., vol. 4894, International Society for Optics and Photonics. SPIE, 2003, pp. 118 – 125. [Online]. Available: <https://doi.org/10.1117/12.469124>
- [32] D. P. Grosvenor and R. Wood, "The effect of solar zenith angle on modis cloud optical and microphysical retrievals within marine liquid water clouds," *Atmospheric Chemistry and Physics*, vol. 14, no. 14, pp. 7291–7321, 2014. [Online]. Available: <https://acp.copernicus.org/articles/14/7291/2014/>