



Frontier Development Lab (FDL) Europe Technical Memorandum

SSA Live Twin for Space Weather

Authors:

Chia-Man Hung, Hala Lamduar, Martin Sanner, Emma Stevenson,
Robert Jarolim, Josh Veitch-Michaelis, Ioanna Bouri



September 2023



Frontier Development Lab (FDL) Europe Technical Memorandum

SSA LIVE TWIN FOR SPACE WEATHER (Heliophysics Onboard)

Authors:

Chia-Man Hung, University of Oxford, Oxford, UK
Hala Lamdouar, University of Oxford, Oxford, UK
Martin Sanner, University of Dundee, Dundee, UK
Emma Stevenson, Universidad Politécnica de Madrid, Madrid, Spain
Robert Jarolim, University of Graz, Graz, Austria
Josh Veitch-Michaelis, ETH Zürich, Zürich, Switzerland
Ioanna Bouri, University of Helsinki, Helsinki, Finland

Frontier Development Lab (FDL) Europe

This work has been enabled by Frontier Development Lab Europe (<https://fdleurope.org>) a public / private partnership between the European Space Agency (ESA), Trillium Technologies, the University of Oxford and leaders in commercial AI supported by Google Cloud and Nvidia Corporation.

FDL Europe and its outputs have been designed, managed and delivered by Trillium Technologies Ltd (trillium.tech). Trillium is a research and development company with a focus on intelligent systems and collaborative communities for planetary stewardship, space exploration and human health.

We express our gratitude to Google Cloud for providing extensive computational resources, as well as to SCANAI, Nvidia and D-Orbit.

The material is based upon work under a programme of, and funded by, the European Space Agency Any opinions, findings, and conclusions or recommendations expressed in this material



are those of the author(s) and do not necessarily reflect the views of the European Space Agency.

FDL Europe's public/private partnership ensures that the latest tools and techniques in Artificial Intelligence (AI) and Machine Learning (ML) are applied to basic research priorities for planetary stewardship and disaster response, for all Humankind.

1. INTRODUCTION

Early detection of solar events heading towards Earth is crucial for responding to strong solar activity potentially targeting earth. One significant event to consider is the Coronal Mass Ejection (CME). A CME involves the expulsion of large amounts of plasma into the outer heliosphere, where it interacts with the stellar wind and electromagnetic systems near Earth or other celestial bodies. A CME colliding with terrestrial infrastructure such as a power grid can cause severe and potentially permanent damage due to the massive amount of plasma involved. Detecting a CME as early as possible is essential to allow for sufficient time to develop and implement counterstrategies. [1]

Currently, data required for early CME detection is collected by a network of strategically positioned satellites. However, none of these satellites can consistently maintain a position that provides ample time between detection and potential CME impact. This situation will change with the upcoming ESA VIGIL mission, planned for launch in 2029. The VIGIL spacecraft will assume a stable orbit at L5, which requires minimal fuel to maintain. This orbit will provide an unobstructed view of the Sun and the space between the Sun and Earth, as depicted in Figure 1, employing a slew of relevant instruments. [2]

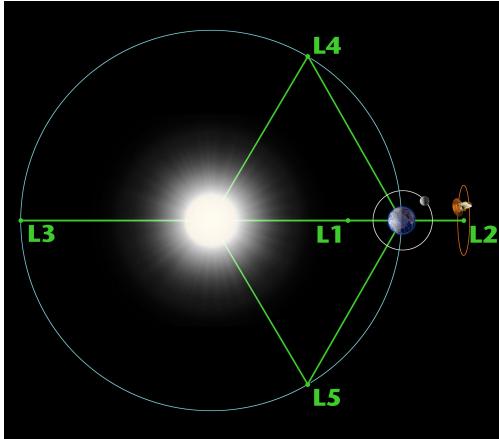


Figure 1: Sketch of the L5 configuration [3]. L1-5 are Lagrange points (points where a balance in gravitational forces of a system is achieved). Due to the solar rotation and viewpoint, estimating the direction of a CME in 3D becomes viable consistently in between the earth and the sun.

Predicting how CMEs propagate is difficult due to the interaction with the surrounding solar wind and the difficulty in predicting the 3D magnetic field of the Sun, as well as the proper 3D structure of the CME itself from a few viewpoints. The structure estimation is an inverse problem. Presented with the opportunities provided by the VIGIL mission, we seek to tackle the detection of CMEs using machine learning (ML) methods, creating a tool to estimate CME parameters (density, velocity vector, distribution) from the experiments onboard the spacecraft in conjunction with a ground-based system. In addition to estimating the 3D reconstruction of a detected CME in space for improved parameter prediction, we also seek to demonstrate the usefulness of ML methods towards improving Beacon data fidelity such as what can currently be seen with the STEREO A spacecraft. ML methods might enable a lower latency or better analytic input to space weather modelling. In summary, upcoming missions such as VIGIL provide us with an opportunity to demonstrate the use of ML onboard to reduce latency, improve our understanding of CME parameters in 3D and characterise CME propagation.

2. PIPELINE

In this section, we outline our developed machine learning pipeline and workflow for the problem of CME nowcasting. The machine learning pipeline, shown in Figure 2, is broken down into two interconnected components: onboard and ground-based processing. Onboard, we have the advantage of real-time data access from diverse instruments without latency, cadence or quality constraints, but suffer from limited computing and memory resources. Conversely, ground-based operations have access to substantial computing capabilities and extensive historical data archives (with multiple sources and therefore viewpoints available), but suffer from data latency due to downlinking.

To best make use of both resources, we distilled the challenge into two core lines of investigation:

1. Onboard, CME-aware compression of coronagraphic observations to reduce data latency, with the ultimate aim of increasing the amount of time we might have to react to an incoming CME event on Earth
2. Ground-based 3D reconstruction of CMEs using 2D observations from different viewpoints for improved CME parameter and structure estimation

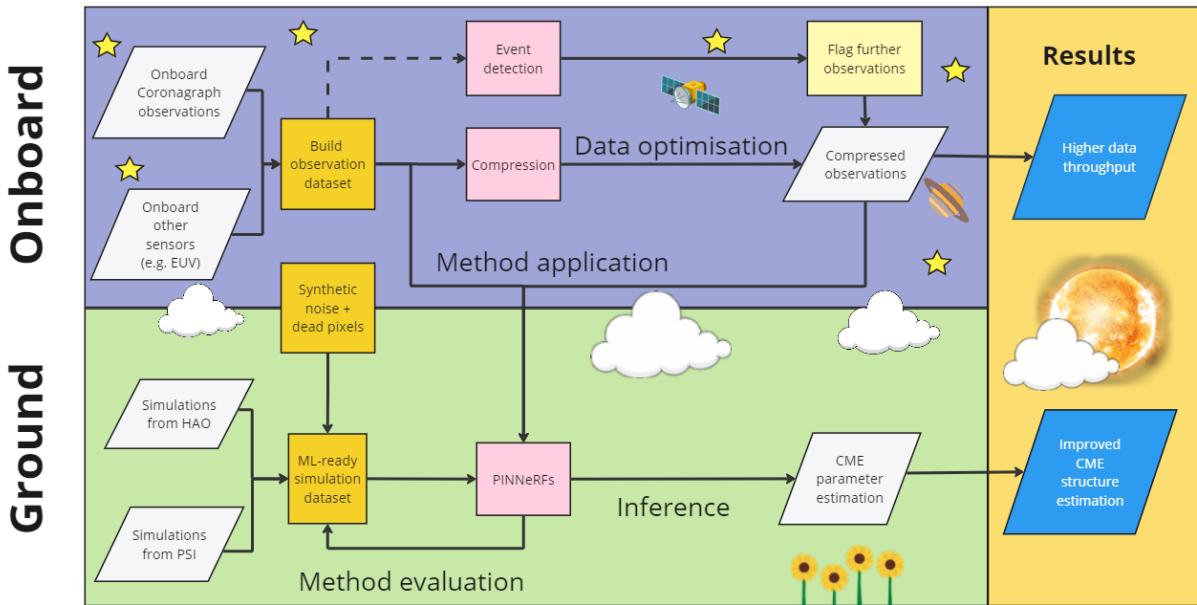


Figure 2: Machine learning pipeline.



2.1 Onboard

In the onboard phase, we focus on two sub-areas, each based on using coronagraphic white-light imagery:

1. **Compression:** ML-Based algorithms for data compression to enable faster downlinking and lower latency
2. **Detection:** Deployment of an onboard classifier or detection to signal instruments to increase observational cadence upon CME detection and to detect and isolate CME-critical areas of observations to optimise compression strategies

2.2 Ground

In the ground-based phase, we adapt the neural radiance field (NeRF model) methodology, which is used for synthesising novel views of complex scenes using a sparse set of input views, to instead reconstruct the electron density of a CME in 3D given coronagraph images from viewpoints from a spacecraft such as VIGIL and from the Earth. We further expand the method in a previously unseen approach by constraining the outputs using physical equations, combining Physics Informed Neural Networks (PINNs) and NeRF in what we dub a PINeRF approach. This 3D reconstruction, for which we additionally have a time component, can then be used to estimate the parameters of a CME, improving our estimation of its structure and, crucially, its arrival time if it were to strike the Earth.

2.3 Workflow

We broke down this pipeline into a series of safe, stretch and bold goals (Figure 3) and associated work packages (WPs), divided between Onboard (OB) and Ground (G) tasks, as follows. The packages tackled are discussed further where applicable in the respective subsections for Onboard and Ground component methodology.

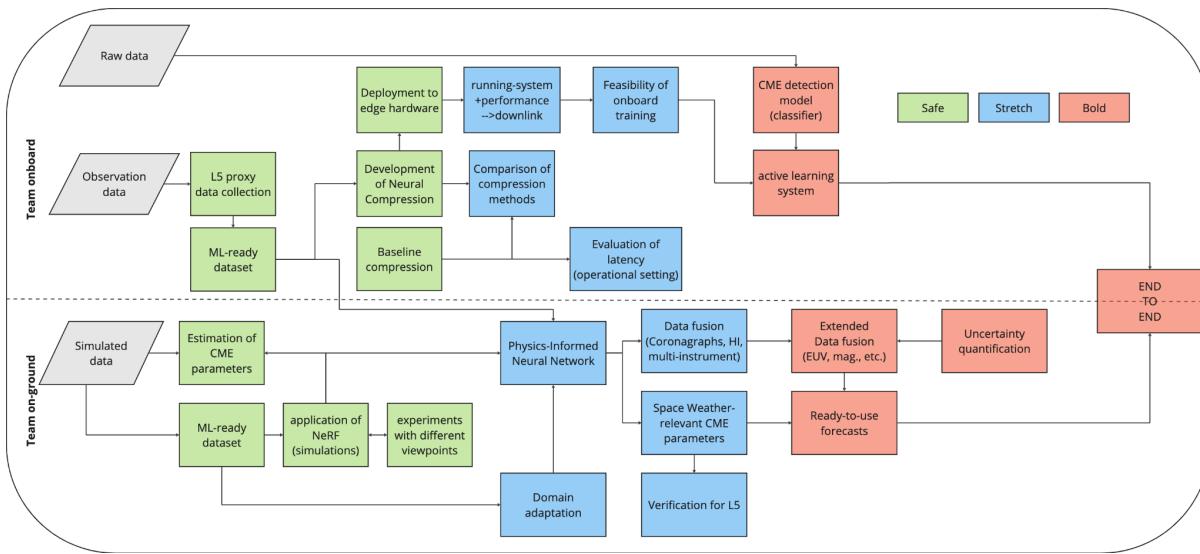


Figure 3: Project pipeline. Tasks are divided between onboard and ground components, and subdivided into safe (green), stretch (blue) and bold (red) goals.

1. WP1: Data collection and preprocessing (OB/G)

Collect and analyse data from relevant CME simulations
 Collect and analyse data from relevant proxy missions
 Generation of ML-ready datasets

2. WP2: Compression model development (OB)

Development of neural compression algorithm
 Evaluation against current compression baselines
 Testing on edge hardware
 Latency evaluation of developed method

3. WP3: NeRF model development (G)

Development of Neural Radiance Fields (NeRF) model for 4D reconstruction of electron density
 Incorporate physics constraints to the NeRF model using physics informed neural networks (PINNs)
 Extraction and estimation of relevant CME parameters
 Viewpoint sensitivity analysis



4. **WP4: Data adaptation and fusion (G)**

Develop domain adaptation approaches for fusing simulated and proxy mission data for developed NeRF model (WP3)

Incorporate data from complementary instruments (e.g. HI, EUV) in developed NeRF model (WP3) for improved CME parameter estimation

Develop uncertainty quantification techniques

5. **WP5: Onboard detector model development (OB)**

Develop onboard CME detector for flagging need for increased measurement cadence and early warning on-ground

Develop approaches for combining class activation maps from the CME detector with compression approaches (WP2) for CME-aware compression

6. **WP6: Deployment and testing (OB/G)**

Combine onboard and ground systems as well as existing solutions in an end-to-end system for generating ready to use nowcasts/forecasts of CME events

Develop approaches for outputting the probability of a CME hitting Earth including uncertainty quantification metrics

Testing of improvement in latency and predictive quality

Demonstrate feasibility with respect to the upcoming VIGIL mission

3. DATA DESCRIPTION

Our machine learning based approach is shaped by the data we have available until the launch and commissioning of the VIGIL mission or similar future missions dedicated to space weather monitoring. We consider several simulation and observational data sets, as shown in Figure 4, and described below.

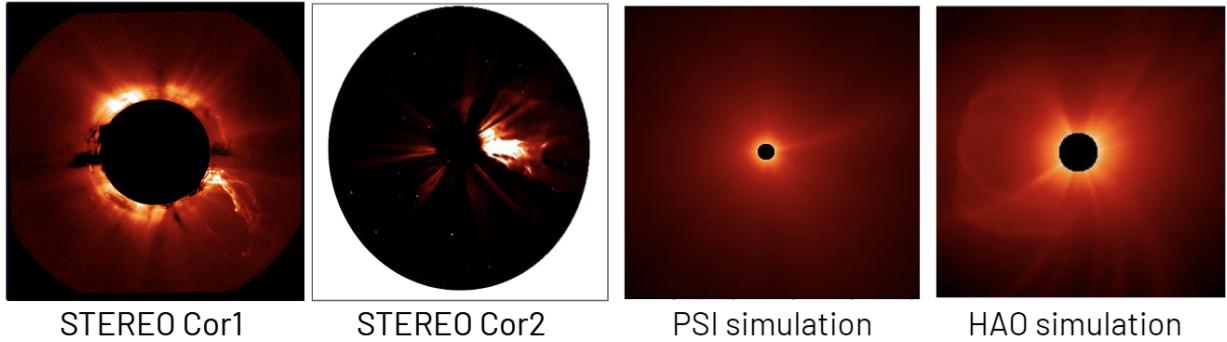


Figure 4: Examples of the four primary data sets used in this work: real observations from STEREO A and B from the Cor1 and Cor2 instruments, and simulated observations provided by HAO and PSI, being similar to Cor2 and Cor1 respectively.

3.1 Simulations

Using synthetic data as a proxy is essential to emulate the anticipated VIGIL mission. To properly benchmark the developed method, we used magnetohydrodynamic (MHD) simulations of the full Sun, where we can render viewpoints at arbitrary angles (i.e., L5 and Earth-based or L1 viewpoints). Specifically, we considered simulations of solar eruptive events developed by researchers at the *High Altitude Observatory* (HAO), and *Predictive Science Inc* (PSI), examples of which are shown in Figure 4. In each case, coronagraphic images of both polarised and total brightness are provided, which are utilised by our PINeRF reconstruction pipeline (Section 4.2), and are in line with recent observing capabilities. In addition, both simulations provide the underlying simulated density cubes, which are used to validate our approach. For these two data sets, the observer field-of-view was selected based on reference missions.

For the PSI dataset, we use a field of view that observes the inner corona in accordance with the COR1 and COR2 instruments onboard the STEREO mission. Cor2 exhibits a similar field-of-view that can be expected from the Vigil mission. The PSI data set represents a static image of the inner corona, meaning that there is no CME present. However, reconstructing the 3D density distribution from a single snapshot is essential to estimate the applicability of our method to the inner corona.

The HAO data focuses on the outer corona (ranging from 20 solar radii to about 1 astronomical unit) and is designed in preparation for the upcoming PUNCH mission. The HAO data set



provides a temporal series of CME evolution. This data set is essential to evaluate the performance of our method for the 3D reconstruction of CME density distributions, as well as their temporal evolution.

3.2 Observations

We also consider real observations from current missions, namely SoHo and STEREO A/B as a satellite-based proxy. We consider the physical setup seen in Figure 1. We note that L5 lies 60° behind the Earth as seen from the sun, with the SoHo satellite located at L1 (close to earth, pointing to the sun). Therefore, a proxy for the data from L5 may be found either by a satellite in proximity to L5 itself or from a reconstruction of a similar *relative* geometrical setup. We employ the latter option, estimating viability from the angles between STEREO A/B spacecrafts (see Figure 5), and the angle between either satellite and SoHo. From this analysis, we found two time periods spanning approximately one year each (2008, 2014) in which the STEREO A and B satellites had a separation of 60° , and before the loss of STEREO B, which would be viable for our investigations. For these periods, we could then further consider both the Cor1 and Cor2 instruments of the STEREO missions, focusing on the inner and outer corona, respectively.

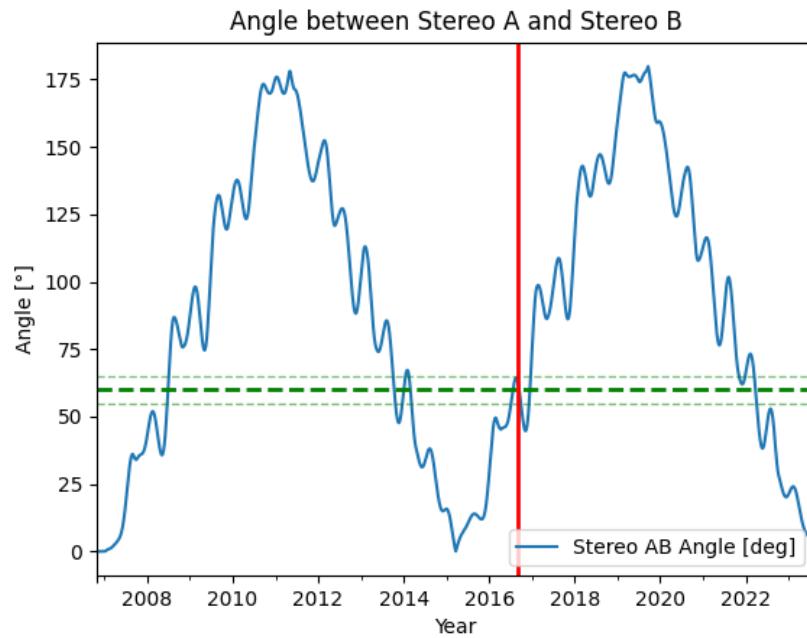


Figure 5: The angle (in deg.) between the STEREO A and B spacecrafts. Lagrange point 5 takes up a 60° angle with Earth, therefore proxy data of what a satellite on Earth may see based on the view from L5 might be gathered by calculating when STEREO A/B take on a similar constellation (apart from also being close to L5). Angles were calculated based on the NASA Horizon tool for calculating Ephemerides [4]. The red line signifies the approximate last contact with STEREO B. Potential angular consideration after this time is not viable due to the inability to recover STEREO B views. The green band signifies the angular range between 55° to 65° - the band around which we seek to emulate L5.



4. METHODOLOGY

4.1 Onboard

Our methodology for the onboard branch is illustrated in Figure 6, beginning with an onboard classifier that is designed to activate a higher observational cadence when a strong CME is detected. To train a model for this task, we first needed to construct a labelled data set, comprising a set of coronagraphic images and an accurate set of associated labels identifying which images contain a CME and should be flagged as such by the predictive model. For this, we first investigated the viability of generating labels using automated methods such as CACTus (Computer Aided CME Tracking)[5] (under [CACTus CME catalog](#)) for the creation of this benchmark data set. However, on finding that this method introduced a high number of incorrect labels for the STEREO images in our treatment, we elected to focus on curating a data set using expert human labels web scraped from available catalogues such as a Cor1 Catalog found under [Observer's Log & CME Catalog](#). Given the timestamp of an image, the catalogue was used to provide a binary label on the presence of a CME, but also had the advantage of being able to provide the direction of the CME for label validation and comment keywords such as “massive” and “faint” to enable better model performance evaluation.

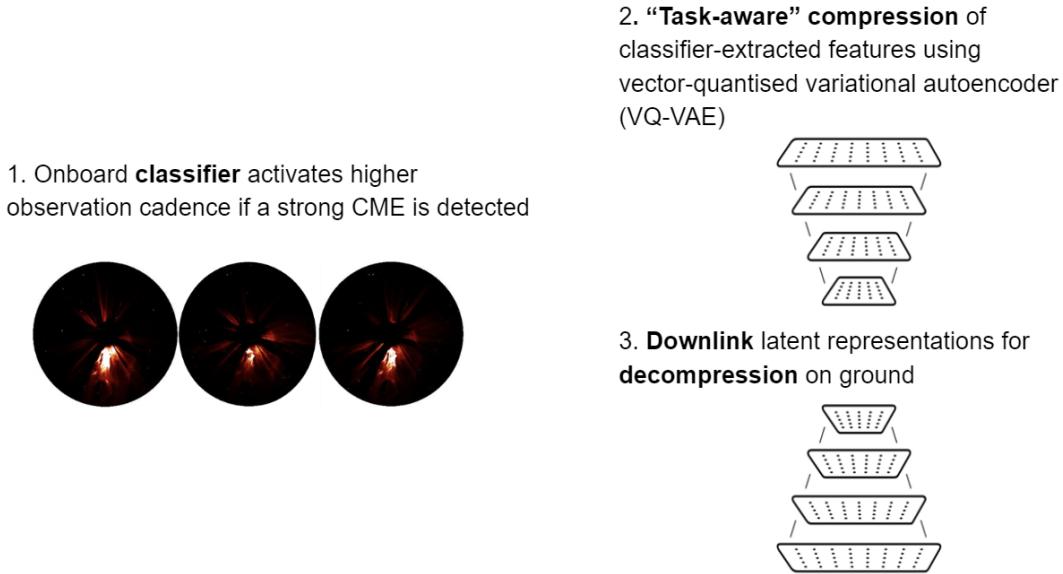


Figure 6: Onboard approach — CME detection + task-aware compression + downlinking.

While this classifier is the first step of the machine learning pipeline in our long-term vision, it originally formed part of the “bold and crazy” goals of our project (Figure 3), and thus work on the curation of this data set, and preliminary feasibility studies are still ongoing. During our initial experiments, we focussed on STEREO A and B images in their three available polarisation angles (0° , 120° , 240°) for February 2014, which we found to contain strongly visible CMEs from the perspective of either or both satellites. After labelling using online catalogue information as described above, we pre-processed the images by first subtracting a background image (generated from all non-CME images from the previous day for each satellite and polarisation angle), before differencing each image with respect to the previous image, and applying normalisation.

For the classifier itself, we selected the ResNet-18 architecture [6], a pre-trained convolutional neural network (CNN) which we anticipate will be compatible with VIGIL’s future onboard computing resources. The model is intended for training on ground using the aforementioned data set before deployment onboard the satellite, although we foresee active learning of this model onboard to maintain or improve performance as a future avenue of research.



One advantage of using a machine learning based model for this classification task (rather than existing automated methods such as CACTus), is the ability of the network to provide not just a predicted label for a given observation, but also a Class Activation Map (CAM). CAMs indicate the regions in an input image that contribute the most to a particular class prediction, and are generated by projecting the output feature maps back to the input space, effectively highlighting relevant areas. This is of particular interest in the context of this work, as it would allow the satellite to downlink only the important sections of an image, and more specifically, only those that pertain to a CME. This would decrease the amount of data that needs to be downlinked.

In this way, the classifier could be applied onboard as a detector, reducing the time needed to flag the satellite to take a higher cadence of observations during a CME event, and also be used for “task-aware” compression. This would feed directly into a second line of investigation on the use of neural-network-based compression algorithms. In this approach, the model (also trained on ground and deployed onboard) compresses the data into a latent vector representation suitable for downlinking. Once on the ground, this data is then decompressed and integrated into the ground-based segment of our approach (Figure 6). As discussed, the classifier and task-aware compression experiments are ongoing and thus the associated results omitted from this report.

In parallel, we investigated the use of neural network based compression algorithms using coronagraphic images available from STEREO A/B Cor2 observations (at this stage without the application of CAMs). We chose a vector-quantised variational encoder (VQ-VAE) [7] for the compression model. It differs from variational autoencoder (VAE) in two ways. The encoder network outputs discrete, rather than continuous, codes; and the prior is learnt, rather than static and chosen in advance. The output of the encoder $z(x)$ is mapped to the nearest point in the discrete embedding space. The gradient of the loss with respect to z will push the encoder to change its output, which could in turn change the configuration in the decoder's forward passes. In this work, we tried VQ-VAE and several of its variants from an open-source codebase called CompressAI [8].

The Vector-Quantizer module of a VQ-VAE essentially learns a codebook of vectors from the training samples. Then each compressed vector that comprises the output of the encoder is replaced by its nearest respective vector in the learned discrete codebook. The discretization of the latent space in VQ-VAEs makes the optimization process simpler. In addition, unlike many VAE architectures that feature a more complex decoder module, VQ-VAEs are less prone to posterior collapse. In VQ-VAEs there is no need for a KL-divergence loss, since this is handled

through the process of matching input vectors to the nearest embedding in the VQ module. Therefore, unlike VAEs, in VQ-VAEs there is no Kullback-Leibler term that can become zero and cause posterior collapse. As an example, consider Figure 7, showcasing an example of a pre-processed observational input from Cor2 with r-squared masking and clipping applied, run through the VQ-VAE, as well as the difference. We can see that the relevant outflows are kept in the example.

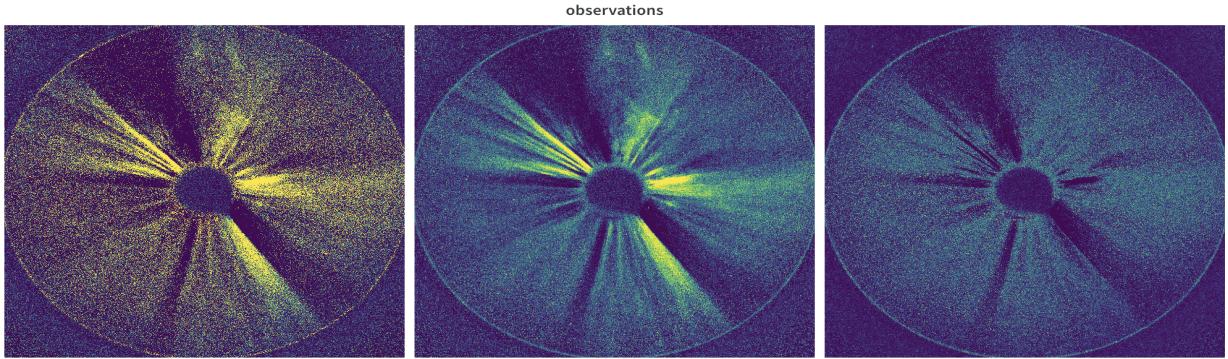


Figure 7: Example of the compression through VQ-VAEs on observational data. Left: pre-processed original image; centre: compressed and reconstructed image; right: difference.

Any innovation in the field of onboard compression needs to be not only carefully motivated and developed, but also thoroughly benchmarked against existing and currently used methods (e.g. the CCSDS compression standards). In the short time available during the research sprint, it can be particularly challenging to complete such a thorough investigation, especially when the onboard compression component is only part of a larger pipeline. However, we believe that further work in this direction can be promising — more specifically, we would like to link the output of the onboard CME detector (that is also in development) to the onboard compressor. As discussed above, the onboard CME detector can produce an attention map — a mask where each pixel of the input image is assigned a probability of its likelihood to contain information of a CME event. This attention map can then be used to bolster the onboard compression, such that areas where a CME event is taking place are compressed so that higher resolution is maintained, unlike other areas of the image that might be of lesser interest. Intuitively this can draw a parallel to the concept of adaptive mesh refinement (AMR) that is often used in simulations, and which has also been used recently for the development of adaptive image compression methods [9].

Coronagraph observations from the STEREO Cor2 Instrument are taken at three polarisation angles in order to estimate the total and polarised brightness of the incoming radiation. For our

compression, we treat each polarisation angle separately. After the decompression, the resulting three observations are combined into the input of the ground-based pipeline. For this we use the IDL *SECCHI_PREP* routine that processes the three polarisation angles into images of total and polarised brightness. An overview is shown in Figure 8.

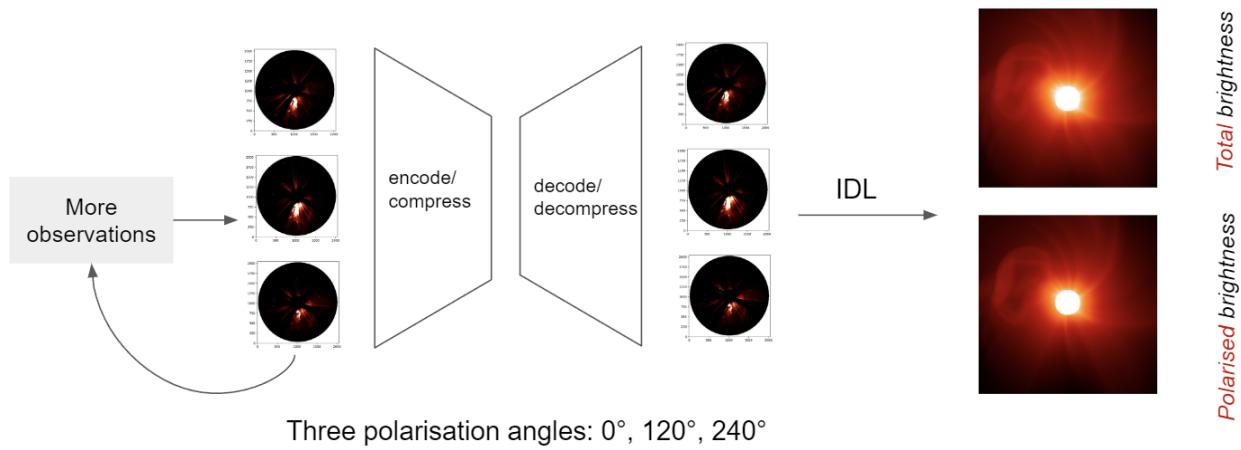
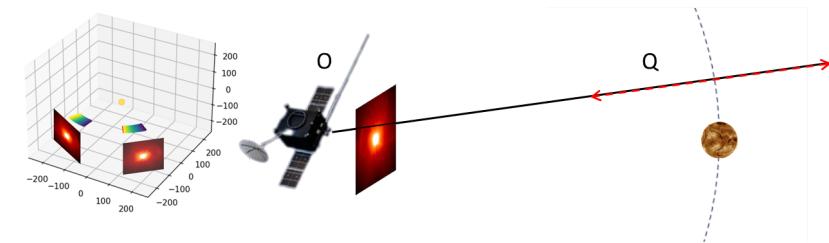


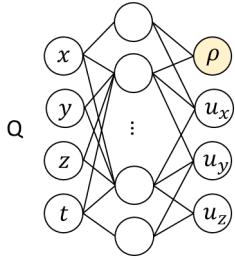
Figure 8: The onboard observations are taken in 3 different polarisation angles: 0°, 120°, 240°. An IDL script is used to convert them into total brightness and polarised brightness.

4.2 Ground

1. For each viewpoint and each pixel in the image, sample points along the line-of-sight.

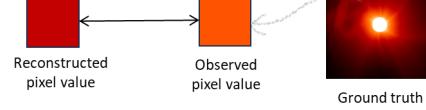


2. Forward pass for each sampled point.



3. For each pixel, integrate the intensity from Thomson scattering to reconstruct the pixel value (prediction).

$$\tilde{I}_{\text{pixel}} = \int I_{\text{Thomson}}(O, Q) \rho(Q) dQ$$



4. Establish physics constraints.

$$\text{Continuity loss: } \mathcal{L}_c = \left| \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right|$$

$$\text{Radial loss: } \mathcal{L}_r = \left\| \frac{\mathbf{u}}{\|\mathbf{u}\|_2} - \hat{\mathbf{e}}_r \right\|_2$$

$$\text{Velocity loss: } \mathcal{L}_v = \left\| \frac{\mathbf{u}}{\|\mathbf{u}\|_2} - \mathbf{u}_{\text{target}} \right\|$$

5. Optimise to match observed pixel value and physics constraints.

$$\mathcal{L}_{\text{total}} = \left| \tilde{I}_{\text{pixel}} - I_{\text{pixel}} \right| + \lambda_c \mathcal{L}_c + \lambda_r \mathcal{L}_r + \lambda_v \mathcal{L}_v$$

Figure 9: Ground approach — PINeRF = Physics-Informed Neural Network (PINN) + Neural Radiance Field (NeRF).

The downlinked images are enhanced by the application of clipping (e.g. similar to the reasoning behind the solar mask, solar brightness can be overwhelming, thus hiding reflections due to e.g. CMEs). Background subtraction on the images from the first image in the current data run also filters pre-existing components, highlighting the new information in incoming images, therefore easing the task at hand. Finally, an r-squared masking process can be applied; that is, pixels further out from the centre of the image are highlighted, placing higher importance on regions where a CME is expected to appear - particularly useful for visualisations.

To span a volume of interest from the initial total and polarised brightness measurements, we have to turn them into sequences of rays from the viewpoint O to the deprojected pixel coordinate Q, as seen in Figure 9. Points in the ray are sampled in two approaches, using stratified and hierarchical sampling along the line of sight.

Our model is adapted from the 2022 FDL project SuNeRF, modified to reconstruct a 4D electron density and velocity between the surface of the Sun and the Earth.[10]

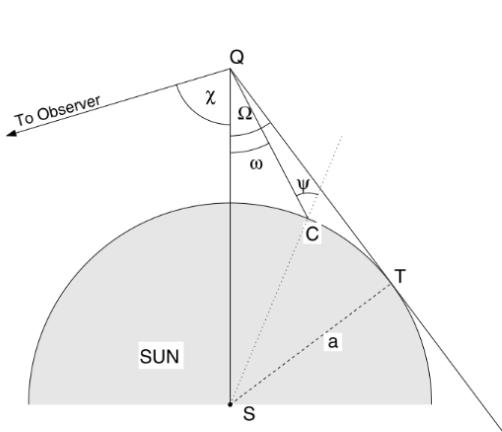


As illustrated in Figure 9, our model combines a physics-informed neural network (PINN) together with Neural Radiance Field (NeRF) and is called PINeRF. The model takes as input a position x , y , z and a timestep t , predicting the density (ρ) and velocity (u_x , u_y , u_z) at that position and time.

As the training data consists of the total and polarised brightness from different viewpoints, we need to note the options available to us.

For viewpoints, we are considering the scenario of only having access to one viewpoint (from the perspective of L5), two viewpoints (with data from both L5 and Earth being considered (or similarly to it from observations), two viewpoints with “background” (which is here referring to the situation where we have a full time sequence of images from two viewpoints alongside some initial views from views scattered throughout the solar system at radius of the earth - this is for instance the case of some additional satellite information being available), the case of 5 satellites (such as L5/Vigil, SoHo (L1/Earth), Stereo A (while it is still available), potentially inputs from satellites such as the Parker Solar Probe, or the Solar Orbiter) (though this is more of a theoretical exercise to test the system outputs when provided with more data), as well as testing the system in a theoretical full view configuration (where the entire solar sphere is visible nonstop, from simulation).

The realistic scenarios are the 2 viewpoints case as a baseline (assuming no satellite is online besides the ability to watch the sun from L5 and earth), though a higher view count is strictly speaking more relevant, assuming the ability to access multiple satellite data sources in white light coronagraphs.



$$A = \cos \Omega \sin^2 \Omega,$$

$$B = -\frac{1}{8} \left[1 - 3 \sin^2 \Omega - \frac{\cos^2 \Omega}{\sin \Omega} (1 + 3 \sin^2 \Omega) \ln \left(\frac{1 + \sin \Omega}{\cos \Omega} \right) \right],$$

$$C = \frac{4}{3} - \cos \Omega - \frac{\cos^3 \Omega}{3},$$

$$D = \frac{1}{8} \left[5 + \sin^2 \Omega - \frac{\cos^2 \Omega}{\sin \Omega} (5 - \sin^2 \Omega) \ln \left(\frac{1 + \sin \Omega}{\cos \Omega} \right) \right].$$

$$I_T = I_0 \frac{\pi \sigma_e}{2z^2} [(1-u)C + uD]$$

$$I_P = I_0 \frac{\pi \sigma_e}{2z^2} \sin^2 \chi [(1-u)A + uB]$$

$$I_{tot} = 2I_T - I_P$$

Figure 10: Thomson scattering. Left: Diagram for visualisation of the definitions of different terms. Right: Equations used in our code to compute the polarised and total intensities.

The NeRF model, as set up for the 2022 project, was designed to estimate EUV Intensity data from any perspective. To recalibrate it from EUV data to densities and velocities, we assume an underlying electron density (to be described by our networks alongside the electron velocity), which we can use with the equations in Figure 10 to describe the total and polarised intensities accounting for all intensities along a line of sight.

To ensure sensible incorporation of all points along the line of sight, two models are applied, a fine and a coarse model, each with their own point sample setup - the “coarse” model employs stratified sampling techniques on the available line of sight distances, whereas the “fine” model employs hierarchical sampling. Points are generated in a coordinate system working in Solar Radii ($1 R_\odot = 6.957e8 m$), with time being defined in days.

The emerging total and polarised brightness can then be checked against the two initial images, setting up the training loop. Both the fine and the coarse model follow the same network structure, employing sine functions as activations (effectively turning the input space from coordinates into k-space) along 8 linear units, yielding 4 outputs - the electron density ρ_e and velocity vector \underline{u} . [10]

With the output of the ground model being the electron density and velocity at each sampled point, becomes necessary to enhance the previous system with additional physics to finetune



the viability of the result - otherwise, the velocity of particles in the field would remain unconstrained, allowing for e.g. the influx of particles back into the sun.

Turning the NeRF into a PINeRF was achieved by the application of common assumptions about CME propagation. These assumptions were equipped with Lagrange multipliers and appended to the loss function slated to match the predicted emerging polarities to the original ones.

The modifications incorporate the full range of network outputs into the training.

For the initial studies, we have identified three changes to the loss, noted in Figure 9:

1. The Mass Continuity Equation $\frac{\partial \rho_e}{\partial t} + \rho_e (\nabla \cdot \underline{u}) + \underline{u} \cdot \nabla \rho_e = 0$ for electron density ρ_e and velocity vector \underline{u}
2. A directional regularisation, $(\hat{\underline{u}} - \hat{\underline{e}_r})^2 = 0$ for velocity unit vector $\hat{\underline{u}}$ and radial unit vector $\hat{\underline{e}_r}$
3. A velocity regularisation incentivises the network to meet a required velocity for CMEs to break free, defined as $||\underline{u}| - |\underline{u}|_{target}| = 0$ given a target speed for the CME derived from the underlying data if available or otherwise from solar wind consideration.

In a perfect system, all of these will equal zero, allowing the network to optimise matching the input images. Otherwise, they act to improve the network output for physical characteristics.

The logic behind the selected modifications is as follows.

For one, mass must be constrained in the system. This constraint signifies that coronal mass ejections travel out from the solar volume, not in, with mass not being added at random points in time and space. It is important to note here that the solar wind can deposit mass in interaction with e.g. CMEs [11]. In turn, the use of the requirement here implies that we assume all mass to be present in the system up-front.

Second, we note that coronal mass ejections follow the geometry of the initial flux ropes from the solar surface that led to the eruption.

As a result, they tend to travel radially out from the sun, making the use of the directional requirement sensible. [12]

Finally, if we assume some initial amount of purely static matter, it becomes possible for the system to maintain this static solar environment. Without the third requirement, this implies $\underline{u} = 0$.



This behaviour is of no interest to us, signifying the state without CME or surrounding solar wind (which is providing environmental values for the area surrounding the sun, noting a counteracting pressure for instance against which the CMEs have to move).

Therefore, setting an expected velocity is recommended. Based on the HAO Simulations, we expect a velocity of $|\underline{u}|_{target} = 75 \frac{R_\odot}{3 Days} \approx 201 \frac{km}{s}$.

The timescale of the system is as such set at 3 days. We would expect viable CMEs (that is, those that can reach earth) to be able to surpass pressure due to surrounding solar wind.

Based on summaries in [12], one should expect a continuous wind between

$$200 - 900 \frac{km}{s} = 74.51 - 335.32 \frac{R_\odot}{3 Days} \text{ and supersonic wind around an average of}$$

$$400 - 500 \frac{km}{s} = 149 - 186 \frac{R_\odot}{3 Days} \text{ (with highs reaching}$$

$$700 - 750 \frac{km}{s} = 260.8 - 279.43 \frac{R_\odot}{3 Days} \text{ and slow solar wind reaching}$$

$$300 - 400 \frac{km}{s} = 111.77 - 149 \frac{R_\odot}{3 Days}.$$

Similarly, we can find expected CME speeds around $300 - 500 \frac{km}{s} = 111.77 - 186 \frac{R_\odot}{3 Days}$.

This yields the parameter space for the target velocity, as well as an expectation for what winds might be surpassed - expecting a CME to be mostly unaffected by continuous wind.

In total, the CME can be considered as a local surplus of density and velocity. Therefore, with a system successfully capable of predicting the coordinate space of densities and velocities, we can track statistical outliers - the top percentiles - in both density and speed simultaneously to restrict the prediction to the CME. Therefore, it is possible to predict the CME parameters from these groups of points. Similarly, the statistics of the system can be covered by repeated experiments of the same setup - that is, the same applied viewpoints (input image orientation) and Lagrange multipliers alongside training parameters. Calculating the standard deviations from this setup then allows one to restrict the errors in the system and any reported quantities. Additionally, the error in known quantities - specifically the density for the simulation - can also be considered in the same way. This yields a measure of the systematic error associated with the system, opening up the opportunity for uncertainty quantification if factoring in the errors in the original system. Without those errors however, quantifying the uncertainty in the output is problematic - it constitutes a further wanted enhancement to the system.



5. TOOLS, COMPUTE, AND SOFTWARE ENVIRONMENT

Our project was developed primarily using cloud infrastructure provided by FDL partners. This included several systems on Google Cloud Platform (GCP) and GPU/compute machines provided by Scan AI and NVIDIA.

Since the project was developed in a remote fashion, the team used shared development environments accessed through Visual Studio (VS) Code. This proved to be an effective tool for both solo and pair programming workflows. All code was checked into Github using distinct branches for the onboard and ground components during the sprint; these branches are planned to be merged in the future. For deployment and testing, we provide a Docker container that can be used for NeRF generation with a single command.

5.1 Software Framework

All our models were developed in Python using the Pytorch and Pytorch Lightning frameworks. This allowed us to leverage existing ML models, including some previously developed at FDL ([SuNeRF](#)). For tracking model training and other hyperparameter optimisation, we used Weights and Biases ([W&B](#)) via an educational licence. An up-to-date list of libraries used in the project can be found in the requirements specification¹ in our repository, but we used standard/common libraries as much as possible - for example, Numpy and Matplotlib.

Our data preparation leveraged a few domain-specific libraries (SunPy², Astropy³) to pull data from NASA's archives. This included ephemerides and instrument data products from SoHo and STEREO.

5.2 Hardware

We used a mixture of GPU models for development - the NVIDIA V100 and A100 for NeRF development and the T4 for onboard models. The compute/RAM requirement for these compression and classification models were significantly less than the NeRFs. These GPUs were allocated through GCP's Compute Engine service and we also used FDL-provided instances from SCAN AI and NVIDIA. D-Orbit provided a "flatsat" computer that we used for

¹ <https://github.com/FrontierDevelopmentLab/2023-europe-space-weather/blob/main/pyproject.toml>

² <https://sunpy.org/>

³ <https://www.astropy.org/>



testing onboard applications using a containerised OpenVino environment (SpaceCloud, see figure below).

For data storage we used a mixture of local persistent storage on GCP and buckets for larger archival datasets (e.g. time-series data products from satellites).

```
[root@spacecloud-ode-00-f0 ~]#
[root@spacecloud-ode-00-f0 ~]#
This is a SpaceCloud(R) compatible machine.
SpaceCloud ODE (Optimized Development Environment)
For support: spacecloud.support@unibap.com or https://unibap.com/en/support
SpaceCloud Operating System (SCOS) based on Ubuntu 20.04 (FOCAL)
SCOS version : 2.0

root@spacecloud-ode-00-f0
OS: Ubuntu 20.04 focal
Kernel: x86_64 Linux 5.15.15-051515-generic
Uptime: 1d 2h 43m
Shell: sh
Disk: 21G / 81G (28%)
CPU: AMD GX-412HC SOC with Radeon R3E Graphics @ 4x 1.2GHz
RAM: 516MiB / 1707MiB
```

Figure 11: Hardware specifications of Unibap SpaceCloud Framework, which acts as an approximation of the onboard environment.

6. RESULTS

6.1 Onboard

In this section, we present the preliminary results of our proposed methodology. Starting onboard, we first need to understand whether the hardware onboard the satellite would be capable of running our compression algorithm. To verify this, we measured the time it takes to compress an image in a Unibap SpaceCloud Framework (Figure 11), which acts as an approximation of the onboard environment.

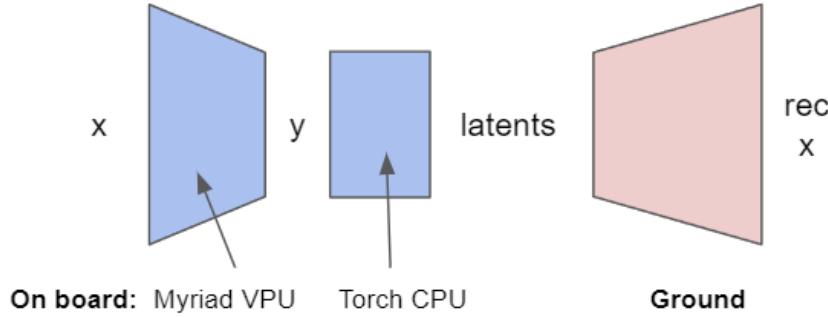


Figure 12: Encoder and decoder of our compression model. The encoder is separated into two parts: $x \rightarrow y$ and $y \rightarrow \text{latents}$. The encoder is run onboard whereas the decoder is on ground.

As shown in Figure 12, in our implementation, the onboard encoder is separated into two parts. First, we encode x to y . Then, we pass y through a few more layers to get the latents. Decoding the latents is done on the ground, so we do not need to measure its time. We compare using the Myriad neural computing stick for the encoding, against only using CPU. The results of this comparison are displayed in Figure 13.

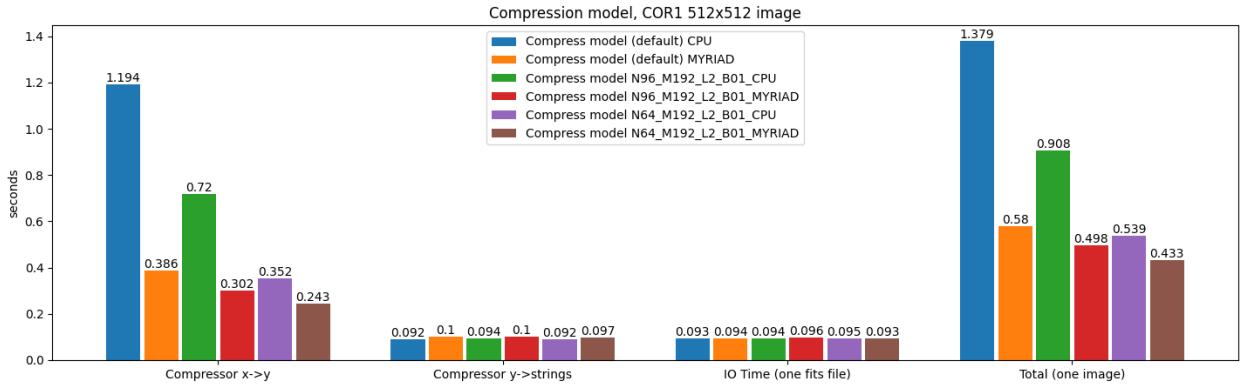


Figure 13: Time measurement of different components of different compression models on Flatsats. M,N,L are hyperparameters that determine the model architecture and model size.

For the default model (blue vs orange), if we include the IO file loading time, we achieve approximately 58% time reduction using Myriad (blue vs orange). If we change the M,N,L hyperparameters of the model, we change the model architecture and also the model size. The green and red are both the model with the same set of hyperparameters, but one is measured

on Myriad while the other on CPU. Similarly, purple and brown models are another pair of measurements. From the results, we observe that encoding x to y is where the time measurement differs. In all pairs, the compression is faster on Myriad than on CPU.

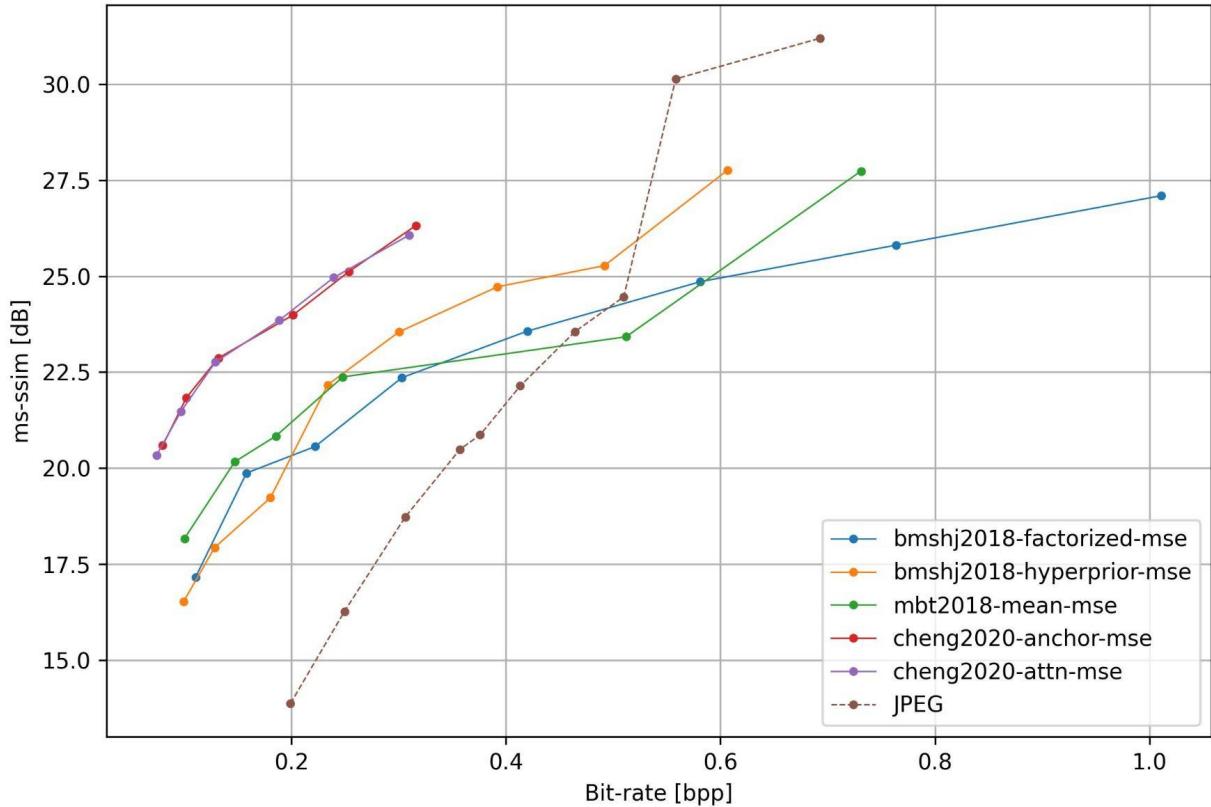


Figure 14: Preliminary results for onboard compression: ms-ssim vs bit-rate among VQ-VAE variants (bmshj2018-factorized-mse [13]; bmshj2018-hyperprior-mse [14]; mbt2018-mean-mse [15], cheng2020-anchor [16], cheng2020-attn [17]) and JPEG (baseline).

In terms of the quality of compression performance, in Figure 14, we plot multi-scale structural similarity (ms-ssim) versus bit rate (bpp=bit-per-pixel). Ms-ssim is a compression quality measurement capturing the structural information. Bpp is an absolute measure and represents the average number of bits needed to encode each image pixel information. In general, a higher bit rate means more bits are used to encode the image, thus corresponds to a higher compression quality. For the model evaluation, we run the models on a fixed set of

pre-processed Cor2 images. Our preliminary results show that VQ-VAE variants are comparable to JPEG, and in the range of low bit-rate (<0.5 bpp), they exhibit higher compression quality.

6.2 Ground

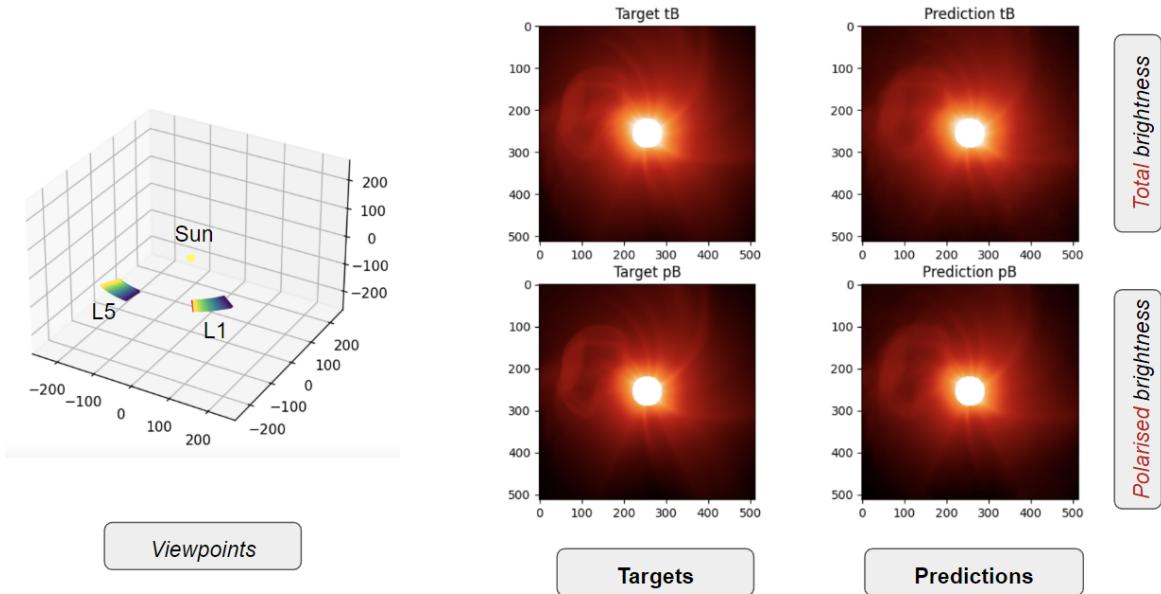


Figure 15: PINeRF qualitative results. Left: Two viewpoints L1 and L5. Right: Prediction of total and polarised brightness vs ground truth.

In Figure 15, we show some qualitative results on reconstructed pixel values alongside the ground truth, using a model trained on two viewpoints. They are nearly identical from human eyes.

It is possible to train a model from only two viewpoints, even though this may seem like an ill-posed problem. The key is to exploit the rotation of the Sun. Even though we are observing from fixed viewpoints in the Earth-Sun system, the Sun's rotation causes us to see different parts of its surface over time. This provides us with additional information that can be used to train a model.



It is worth noting that even if we made a mistake in our equations in Thomson scattering, we could still get good reconstructions because the model is trained to match the reconstructed pixel values to the ground truth. It is therefore essential to compare what we are most interested in, which is the density output.

	2 viewpoints (L5 + L1)
No physics constraints	1.66
Continuity	1.40
Continuity + radial	1.30
Continuity + radial + velocity regularisation	1.43

Table 1: PINeRF quantitative results — Mean absolute error of predicted density ρ and ground truth over timesteps (using simulated data).

For quantitative results (see Table 1), we conduct an ablation study on physics constraints and train four different models:

- Trained without any physics constraints; only matching the pixel values.
- Adding the continuity loss.
- Adding the continuity loss and the radial loss.
- Trained with all three physics equations.

We compare the mean absolute error of predicted density ρ and ground truth over timesteps using simulated data. In simulated data, we have the ground truth of the density, which is not available in the real data. In the real world, the density is not something we can observe directly and is something we are predicting.

The preliminary results show that adding physics equations helps the model learn the density output. The best result is obtained by adding the continuity and the radial losses.



To estimate the error in our prediction, we can attempt an uncertainty quantification using repeated experiments with the same settings. The setup provides two direct options here. Option 1 entails the calculation of the deviation in individual point predictions, working through the standard deviation at every point. Calculating confidence in the point prediction then estimates the differences throughout the volume, including background, yielding an overall analysis of features the model is good at estimating.

While of course this option is viable for the overall system output, for the express goal of the program, the ability to predict the 3D structure of a detected CME from input images and estimate its' movement on that level implies that a better predictor of variance within the system would be to evaluate the deviations on only the targeted positions.

Performing such a volume restriction to only relevant substructures in the generated space means that individual voxels can have further variance in them without disturbing the measurement (for instance, having slightly different configurations of the background density, or the density close to the sun would then not disturb the measurement of the CME parameter errors).

As such, we have set up the systems to include an error estimation mode, calculating which positions belong to a CME in a given realisation, then calculating the standard deviation of those points specifically over a sequence of different experiments with the same parameter sets. We perform this analysis on both system outputs - the density and the velocity.

Analysing the output on HAO simulator data this way for instance, we find that the different models trained with solely the continuity constraint currently have a standard deviation in

density on average of $2.214e25 \frac{\text{electrons}}{R_\odot^3}$, expecting an average CME density of

$$1.180e26 \frac{\text{electrons}}{R_\odot^3}.$$

We perform a similar analysis of the velocities found by restriction of particle speeds and densities, finding that the velocity vector has standard mean standard deviations over the different timesteps of $(0.558, 0.533, 0.440) \frac{R_\odot}{3 \text{ days}}$ (for the x,y,z component specifically).

However, the reported mean velocity over the different timesteps is

$$-(1.263, 2.44, 0.191) \frac{R_\odot}{3 \text{ days}} \rightarrow |v| = 2.754 \frac{R_\odot}{3 \text{ days}}.$$

This is too low for the expected CME as pointed out earlier, signifying that there is a mismatch between the generated velocities and expectations that needs to be tracked further in training. In essence, one way to look at this would be to consider the effect of restricting the regularisation of resulting features to only the components in question. One option here would be to rework the loss modification to be based on only the individual voxels making up the CME. Based on the previous analysis of differences to the simulation, however, we find that the overall progression is still viable - yet has space for improvement.

As an example, consider Figure 16, showcasing the results of the ICARUS system for a single time step in a top-down view. The run shown here has been trained on all available viewpoints.

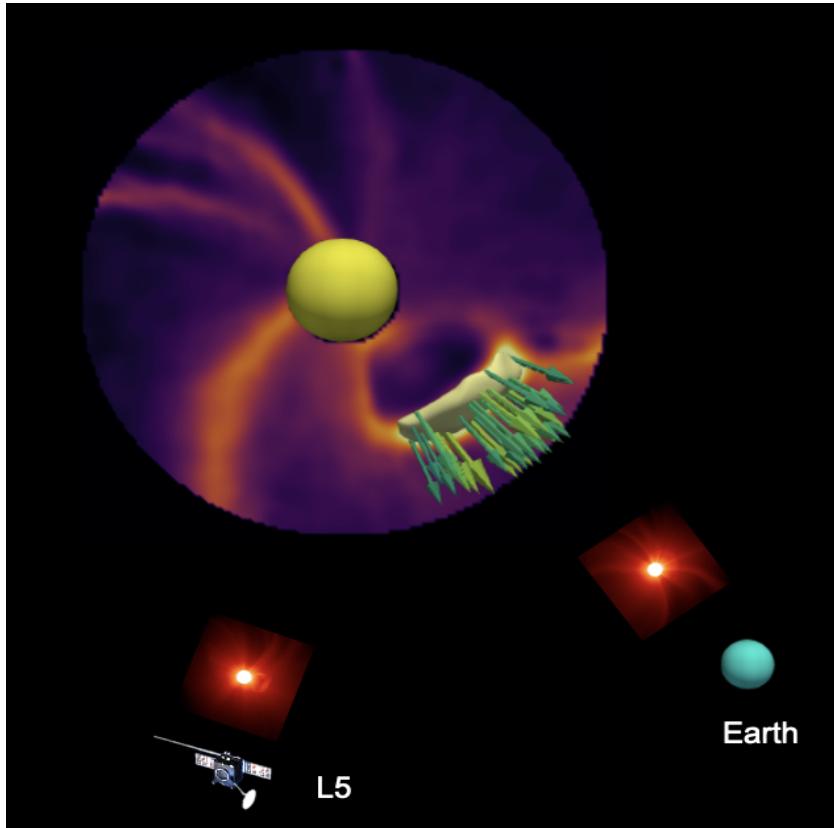


Figure 16: Top-down view of the results of ICARUS on simulation data. Shown are the position of Earth and L5, with VIGIL in place, alongside examples of what the respective views look like.



Restricting the viewpoint count causes the CME topology to change. This is to be expected, as the inverse problem posed by the CME structure (with different electron configurations being in principle able to generate the same views depending on viewpoint) might not be uniquely solved. However, with the systematic ability to bind in new data sources of the same type currently, with this portion of the ICARUS pipeline being designed for options with multiple simultaneous inputs, the multi-viewpoint scenario is more realistic than the single or dual viewpoint one. Expanding this pipeline for the fusion of different datatypes is a possible further expansion of this, though it would require a rework in outputs of the system, as there still need to be links between what the system produces and what inputs are available.

7. CONCLUSION

With this FDL project we present an end-to-end pipeline that combines ground-based and onboard compute environments to advance our space weather situational awareness.

7.1 Onboard Results

For future space weather monitoring missions, we propose an onboard CME-aware compression system capable of optimising and reducing the downlinking time to Earth. To achieve this, we introduced two ML-based modules: a classification module for CME detection in coronagraphic images, and a compression module for downlinking. This system has the potential to improve on classical methods by autonomously boosting observation cadence during a CME event and only downlinking CME-critical data. Leveraging ResNet and VQ-VAE models for onboard classification and compression, we explore our novel approach on proxy observation data from the STEREO mission. For the compression module, we investigate performance on a “flatsat”, a platform approximating the onboard hardware environment, and demonstrate a significant reduction in compression time on Myriad VPU compared to CPU. In the future, we aim to adapt our onboard solution as more hardware specifications are disclosed regarding the VIGIL mission, and ultimately finetune our models on real data collected upon its launch.

7.2 Ground Results

We propose a novel approach to reconstruct the solar corona from white light images taken from two viewpoints in the constellation of the anticipated vigil mission (Earth + L5). Our PINNeRF approach utilises physical equations for continuity and radial velocity profiles in the 3D reconstruction of the NeRF approach. Qualitatively, our reconstruction on simulated data



from only two viewpoints (L1 and L5) show reconstruction that are close to the ground-truth density distribution of the CME simulation. Quantitatively, our ablation study on physics constraints indicates that incorporating continuity and radial constraints improves density reconstruction performance, with further performances expected on additional experimentation. With this we demonstrate that AI methods can vastly improve 3D reconstructions of the CME profiles, which provides critical information for the estimation of space weather hazards and enables novel insights into the topology and propagation of CMEs. Our results were obtained from simulated data, which enables a thorough evaluation of our approach.

As the next step, we plan to apply this novel method for 3D CME reconstructions to observational data, using the STEREO twin-satellites and the results from our onboard pipeline. In the future, as more space weather satellites are launched beyond VIGIL, we expect to benefit from additional viewpoint observations that can be further incorporated into our PINNeRF architecture.



REFERENCES

- [1] D. S. Kimball. A study of the aurora of 1859, 1960.
- [2] OSCAR, Esa. OSCAR page for Vigil Satellite Payload - URL:https://space.oscar.wmo.int/satellites/view/vigil_I5
- [3] ESA, Lagrange point sketch, https://www.esa.int/var/esa/storage/images/esa_multimedia/images/2017/11/lagrange_points/17_234002-1-eng-GB/Lagrange_points_article.jpg, as seen 12.7.2023
- [4] Nasa, Horizon Ephemeris Calculator,<https://ssd.jpl.nasa.gov/horizons/app.html#/>, as used 12.7.2023.
- [5] E. Robbrecht and D. Berghmans. Automated recognition of coronal mass ejections (CMEs) in near-real-time data. , 425:1097–1106, October 2004.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. CoRR, abs/1512.03385, 2015
- [7] Van Den Oord, A., & Vinyals, O. (2017). Neural discrete representation learning. Advances in neural information processing systems, 30.
- [8] Bégaint, J., Racapé, F., Feltman, S., & Pushparaja, A. (2020). Compressai: a pytorch library and evaluation platform for end-to-end compression research. arXiv preprint arXiv:2011.03029.
- [9] Feischl, M., & Hackl, H. (2023). Adaptive Image Compression via Optimal Mesh Refinement. arXiv preprint arXiv:2304.01640.
- [10] Kyriaki-Margarita Bintsi, Robert Jarolim, Benoit Tremblay, Miraflor Santos, Anna Jungbluth, James Paul Mason, Sairam Sundaresan, Angelos Vourlidas, Cooper Downs, Ronald M. Caplan, and Andrés Mñoz Jaramillo. Sunerf: Validation of a 3d global reconstruction of the solar corona using simulated euv images, 2022.
- [11] T A. Howard, C. D. Fry, J. C. Johnston, and D. F. Webb. On the evolution of coronal mass ejections in the interplanetary medium. The Astrophysical Journal, 667(1):610, sep 2007.



- [12] Eric Priest. Magnetohydrodynamics of the Sun. Cambridge University Press, 2014
- [13] Factorized Prior model from J. Balle, D. Minnen, S. Singh, S.J. Hwang, N. Johnston: "Variational Image Compression with a Scale Hyperprior", Int Conf. on Learning Representations (ICLR), 2018.
- [14] Scale Hyperprior model from J. Balle, D. Minnen, S. Singh, S.J. Hwang, N. Johnston: "Variational Image Compression with a Scale Hyperprior" Int. Conf. on Learning Representations (ICLR), 2018.
- [15] Scale Hyperprior with non zero-mean Gaussian conditionals from D. Minnen, J. Balle, G.D. Toderici: "Joint Autoregressive and Hierarchical Priors for Learned Image Compression", Adv. in Neural Information Processing Systems 31 (NeurIPS 2018).
- [16] Anchor model variant from "Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules", by Zhengxue Cheng, Heming Sun, Masaru Takeuchi, Jiro Katto.
- [17] Self-attention model variant from "Learned Image Compression with Discretized Gaussian Mixture Likelihoods and Attention Modules", by Zhengxue Cheng, Heming Sun, Masaru Takeuchi, Jiro Katto.